Massachusetts Institute of Technology

Ph.D. Thesis Proposal

Department of Aeronautics and Astronautics

A Bayesian Approach to Optimal Sequential Experimental Design using Approximate Dynamic Programming

Ph.D. Candidate:
Xun Huan

Committee:
Prof. Youssef Marzouk (Chair)
Prof. John Tsitsiklis
Prof. Karen Willcox

External Evaluator:
Prof. Mort Webster

January 30, 2013

Abstract

Experimental data play an essential role in developing and refining models of physical systems. Not all experimental results are equally useful, and some experiments can be much more valuable than others. Well-chosen experiments can thus lead to substantial resource savings. Optimal experimental design seeks to quantify and maximize the value of experimental data.

Common current practice for designing multiple experiments consists of suboptimal approaches: open-loop design that chooses all experiments simultaneously, and greedy design that optimally selects the next experiment without accounting for the future. In this thesis, we develop a rigorous formulation in a closed-loop dynamic programming (DP) framework that yields the true optimal sequence of experiments under the uncertainty of their results. The framework is suitable for nonlinear models and for the experimental purpose of Bayesian parameter inference. Furthermore, we develop a set of numerical tools that solve the DP design problem with particular attention to approximation methods for computationally intensive models. These tools include various methods of approximate dynamic programming, as well as numerical techniques such as polynomial surrogates and stochastic optimization that accelerate the computations.

In this thesis, the superiority of the DP design is demonstrated in a simple linear-Gaussian model problem. Future work will apply the DP design framework to a realistic application problem involving optimal sensor placement in a convection-diffusion field.

Contents

1 Introduction 4

2 Thesis Objectives 5

3 Literature Review 6
- 3.1 Optimal Experimental Design 6
  - 3.1.1 Open-Loop Design 6
  - 3.1.2 Closed-Loop Design 7
- 3.2 Stochastic Optimization 8
- 3.3 Approximating Computationally Intensive Models 8
- 3.4 Dynamic Programming and Approximate Dynamic Programming 9

4 Optimal Design Formulations 10
- 4.1 Open-Loop 11
- 4.2 Closed-Loop 12
  - 4.2.1 Greedy 13
  - 4.2.2 Dynamic Programming 14

5 Numerical Methods 14
- 5.1 Open-Loop 14
  - 5.1.1 Expected Utility Estimator 15
  - 5.1.2 Stochastic Optimization 15
  - 5.1.3 Polynomial Chaos Approximation 18
- 5.2 Closed-Loop 20
  - 5.2.1 Greedy 20
  - 5.2.2 (Approximate) Dynamic Programming 21

6 Example: Linear-Gaussian Model 27
- 6.1 Constant Noise Variance 28
  - 6.1.1 Analytical Results 28
  - 6.1.2 Numerical Results 29
- 6.2 State-Dependent Noise Variance 30
  - 6.2.1 Analytical Results 31
  - 6.2.2 Numerical Results 33
- 6.3 State- and Design-Dependent Noise Variance 34

7 Future Work 35
- 7.1 First Objective: Closed-Loop Design Formulation 35
  - 7.1.1 Intended Stopping Rule 36
  - 7.1.2 Unexpected Changes to the Number of Experiments 36
  - 7.1.3 Formulations Robust to Horizon Change 37
- 7.2 Second Objective: ADP Solution Methods 39
  - 7.2.1 New ADP Algorithm 39
  - 7.2.2 State Representation 40
- 7.3 Model Problems and Applications 41
  - 7.3.1 Simple Model Problems 42

7.3.2 Experiments with Fixed Form and Stationary Parameters 42
7.3.3 Experiments with Varying Form and Stationary Parameters 42
7.3.4 Experiments with Varying Form and Non-Stationary Parameters 42
7.3.5 Above and Beyond 43

8 Milestones and Proposed Schedule 43
8.1 Milestones 43
8.2 Proposed Schedule 44

1 Introduction

Experimental data play an essential role in developing and refining models of physical systems. For example, data may be used to update knowledge of parameters in a model or to discriminate among competing models. Whether obtained through field observations or laboratory experiments, however, data may be difficult and expensive to acquire. Even controlled experiments can be time-consuming or delicate to perform. In this context, maximizing the value of experimental data—designing experiments to be "optimal" by some appropriate measure—can dramatically accelerate the modeling process. Experimental design thus encompasses questions of where and when to measure, which variables to interrogate, and what experimental conditions to employ.

The concept of experimental design has existed for a long time. Traditional experimental design methods, such as factorial and composite designs, are largely used as heuristics for exploring the relationship between input factors and response variables. *Optimal* experimental design, on the other hand, uses a model to guide the choice of experiments for a particular purpose, such as parameter inference, prediction, or model discrimination. Here are a few practical examples.

Combustion Kinetics Combustion phenomena, such as the ignition of a hydrogen-air mixture, are often modeled by a chemical mechanism composed of many elementary reactions. Parameters governing the rates of these reactions, know as kinetic rate parameters, are usually inferred from experimental measurements such as ignition delay times [1, 2]. These kinetic parameters have large uncertainties even today [3, 4, 5, 6], and more data are needed to reduce them. The experiments are often prepared and conducted using, for example, shock tubes. Targeting a particular kinetic parameter, one may repeat the same experiment multiple times under different temperatures and pressures, with different initial concentrations of reactants, and selecting different output quantities to observe (e.g., selected species concentrations) at different times. Optimal experimental design provides guidance to these choices. Such a design involves nonlinear models and targets inference of stationary (time-independent) parameters [7].

Optimal Sensor Placement The United States government has initiated a number of terrorism prevention measures since the events of 9/11. The BioWatch program [8] focuses on the prevention and response to scenarios where a biological pathogen is released in a city. One of the main goals is to find and intercept the (possibly mobile) contaminant source and eliminate it as soon as possible. Often too dangerous to send personnel into the contamination zone, a limited number of measurements may be available from remote-controlled robotic vehicles. It is crucial for the sequence of measurements to yield the most information on the location of the source. The author has investigated a stationary version (static source location) of the problem in [9], and will explore more elaborate scenarios, including mobile source location, in this thesis project (see Section 7.3.4).

Weather Balloons High-altitude weather balloons are deployed by the National Weather Service (NWS) on a daily basis from about 100 locations across the U.S. to obtain temperature, pressure, and other atmospheric measurements for inputs to numerical weather prediction models [10]. The measuring device, known as a radiosonde, typically costs hundreds of dollars, and is usually lost after deployment. Adding labor, tracking, transportation, and storage costs, NWS spends tens of million of dollars to "buy" these data every year. However, if the location and timing of deployments are chosen such that the data collected are most informative for making weather predictions, deployment frequency can potentially be reduced drastically, leading to substantial savings. This is an example of optimal experimental design for the purpose of reducing uncertainty in model-based predictions.

We are interested in the design of multiple experiments. In practice, these experiments can be pursued via two approaches:

- *Open-loop*: All the experiments must be selected concurrently, and we either do not have or cannot use the results from any of the experiments we are conducting (i.e., no feedback). This approach may be necessary due to practical constraints such as a schedule that require all experiments to be performed at the same time. This approach is also known as *batch design*.
- *Closed-loop*: The experiments may be conducted in sequence, allowing the new data to be used to plan the upcoming experiments (i.e., with feedback). This approach is more general than the open-loop approach, and is thus guaranteed to yield better or at least equally good designs. This approach is also known as *sequential design*.

In this thesis, we focus on the most general design formulation—closed-loop design—using a dynamic programming (DP) formulation. Our interest lies in physically realistic and typically nonlinear models. These models are almost always computationally intensive, and due to nonlinearity, Gaussian assumptions are not appropriate. This proposal is structured as follows. Section 2 states the thesis objectives, and Section 3 provides an overview of previous work done in various fields relevant to this project. Rigorous *formulation* of selected design approaches—namely open-loop, greedy, and DP designs—are presented in Section 4, with the key contribution being optimal experimental design under the DP framework. The *numerical methods* for solving these problems are described in Section 5, along with results from the author's past work. Section 6 highlights a detailed investigation of the different experimental design approaches applied to a linear-Gaussian model problem. Also included are preliminary results from the DP design, which are both motivating and insightful. Finally, the proposal ends with a discussion of future work and thesis timeline in Sections 7 and 8, respectively.

2 Thesis Objectives

This thesis project has two primary objectives:

1. to rigorously formulate the optimal closed-loop experimental design problem, for the purpose of Bayesian parameter inference, using a dynamic programming approach; and
2. to develop the numerical tools that find the optimal experimental conditions in a computationally feasible manner.

These objectives, along with proposed solutions, are described in the upcoming sections.

3 Literature Review

3.1 Optimal Experimental Design

3.1.1 Open-Loop Design

The problem of optimally designing experiments has received much attention in the statistics community and in many science and engineering applications. When observables depend linearly on parameters of interest, common solution criteria for the optimal experimental design problem are written as functionals of the information matrix [11]. These criteria include the well-known 'alphabetic optimality' conditions, e.g., A-optimality to minimize the average variance of parameter estimates, or G-optimality to minimize the maximum variance of model predictions. Bayesian analogues of alphabetic optimality, reflecting prior and posterior uncertainty in the model parameters, can be derived from a decision-theoretic point of view [12]. For instance, Bayesian D-optimality can be obtained from a utility function containing Shannon information while Bayesian A-optimality may be derived from a squared error loss. In the case of linear-Gaussian models, the criteria of Bayesian alphabetic optimality reduce to mathematical forms that parallel their non-Bayesian counterparts [12].

For nonlinear models, however, exact evaluation of optimal design criteria is much more challenging. More tractable design criteria can be obtained by imposing additional assumptions, effectively changing the form of the objective; these assumptions include linearizations of the forward model, Gaussian approximations of the posterior distribution, and additional assumptions on the marginal distribution of the data [13, 12]. In the Bayesian setting, such assumptions lead to design criteria that may be understood as *approximations of an expected utility.* Most of these involve prior expectations of the Fisher information matrix [14]. Cruder "locally optimal" approximations require selecting a "best guess" value of the unknown model parameters and maximizing some functional of the Fisher information evaluated at this point [15]. None of these approximations, though, is suitable when the parameter distribution is broad or when it departs significantly from normality [16]. A more general design framework, free of these limiting assumptions, is preferred [17, 18].

More rigorous information theoretic criteria have been proposed throughout the literature. The seminal paper of Lindley [19] suggests using expected gain in Shannon information, from prior to posterior, as a measure of the information provided by an experiment; the same objective can be justified from a decision theoretic perspective [20, 21]. Sebastiani and Wynn [22] propose selecting experiments for which the marginal distribution of the data has maximum Shannon entropy; this may be understood as a special case of Lindley's criterion. Maximum entropy sampling (MES) has seen use in applications ranging from astronomy [23] to geophysics [24], and is well suited to nonlinear models. Reverting to Lindley's criterion, Ryan [25] introduces a Monte Carlo estimator of expected information gain to design experiments for a model of material fatigue. Terejanu *et al.* [26] use a kernel estimator of mutual information (equivalent to expected information gain) to identify parameters in chemical kinetic model. The latter two studies evaluate their criteria on every element of a finite set of possible designs (on the order of ten designs in these examples), and thus sidestep the challenge of *optimizing* the design criterion over general design spaces. And both report significant limitations due to computation expense; [25] concludes that "full blown search" over the design space is infeasible, and that two order-of-magnitude gains in computational efficiency would be required even to discriminate among the enumerated designs.

The application of optimization methods to experimental design has thus favored simpler design objectives. The chemical engineering community, for example, has tended to use linearized and locally optimal [27] design criteria or other objectives [28] for which deterministic optimization strategies are suitable. But in the broader context of decision theoretic design formulations, sampling is required. [29] proposes a curve fitting scheme wherein the expected utility was fit with a regression model, using Monte Carlo samples over the design space. This scheme relies on problem-specific intuition about the character of the expected utility surface. Clyde *et al.* [30] explore the joint design, parameter, and data space with a Markov chain Monte Carlo (MCMC) sampler; this strategy combines integration with optimization, such that the marginal distribution of sampled designs is proportional to the expected utility. This idea is extended with simulated annealing in [31] to achieve more efficient maximization of the expected utility. [30, 31] use expected utilities as design criteria but do not pursue information theoretic design metrics. Indeed, direct optimization of information theoretic metrics has seen much less development. Building on the enumeration approaches of [24, 25, 26] and the one-dimensional design space considered in [23], [18] iteratively finds MES designs in multi-dimensional spaces by greedily choosing one component of the design vector at a time. Hamada *et al.* [32] also find "near-optimal" designs for linear and nonlinear regression problems by maximizing expected information gain via genetic algorithms. Guestrin, Krause and others [33, 34] find near-optimal placement of sensors in a discretized domain by iteratively solving greedy subproblems, taking advantage of the submodularity of mutual information. More recently, the author has made several contributions addressing the coupling of rigorous information theoretic design criteria, complex physics-based models, and efficient optimization strategies on nonlinear experimental design from a Bayesian perspective [35, 7].

3.1.2 Closed-Loop Design

In comparison to open-loop, closed-loop or sequential experimental design has seen much less development. One of the difficulties is a practical one, as addressing such a problem would require substantial computational resources. Only recently with advances in computing power and numerical techniques have we begun to see attempts in tackling sequential experimental design. For example, Cavagnaro *et al.* [36] address sequential design for the purpose of model discrimination based on mutual information and using sequential Monte Carlo techniques, while Solonen *et al.* [37] propose a framework using a variance-based utility and importance sampling for inference. The need for sequential experiments has been acknowledged by many authors. However, majority of the proposed solutions revolve around the suboptimal greedy design.

The truly optimal design method lies within the DP framework [38]. While the method is not yet popular in the experimental design community compared to the more intuitive and simple greedy approach, it is starting to gain traction with many authors beginning to acknowledge its optimality even when suboptimal methods are used [39, 40]. Due to its notorious computational requirements, few attempts have been made to implement a DP framework for design. Most of these cases require simplifications from knowledge of the problem (e.g., sufficient statistics) combined with numerical approximation techniques (e.g., grid search and Monte Carlo simulations), and even then have only attempts been made to solve simple stopping problems with binary decisions [41, 42, 43].

DP is a well studied subject in the controls, operations research, and artificial intelligence communities. Many opportunities exist, and should be capitalized, by borrowing and integrating the techniques from these fields into the sequential experimental design context [44].

3.2 Stochastic Optimization

The optimal experimental design framework involves the maximization of a noisy objective function. This is addressed by stochastic optimization techniques.

There are many approaches for solving continuous optimization problems with stochastic objectives. While some do not require the direct evaluation of gradients (e.g., Nelder-Mead [45], Kiefer-Wolfowitz [46], and simultaneous perturbation stochastic approximation [47]), other algorithms can use gradient evaluations to great advantage. Broadly, these algorithms involve either stochastic approximation (SA) [48] or sample average approximation (SAA) [49], where the latter approach must also invoke a gradient-based deterministic optimization algorithm. Hybrids of the two approaches are possible as well.

The Robbins-Monro algorithm [50] is one of the earliest and most widely used SA methods, and has become a prototype for many subsequent algorithms. It involves an iterative update that resembles steepest descent, except that it uses stochastic gradient information. SAA (a.k.a. the retrospective method [51] and the sample-path method [52]) is a more recent approach, with theoretical analysis initially appearing in the 1990s [49, 52, 53]. Convergence rates and stochastic bounds, although useful, do not necessarily reflect empirical performance under finite computational resources and imperfect numerical optimization schemes. To the best of our knowledge, extensive numerical testing of SAA has focused on stochastic programming problems with special structure (e.g., linear programs with discrete design variables) [54, 55, 56, 57, 58]. While numerical improvements to SAA have seen continual development (e.g., estimators of optimality gap [59, 60] and sample size adaptation [61, 62]), the practical behavior of SAA in more general optimization settings is largely unexplored. A detailed numerical assessment of SAA in a nonlinear continuous-variable design setting has been performed by the author [9].

3.3 Approximating Computationally Intensive Models

The experimental design problem often involves an underlying physical process governed by a set of partial differential equations (PDEs), known as the “forward model”. The numerical methods involve solving the forward model repeatedly, making the overall process computationally expensive. To make these calculations tractable, one would like to replace the forward model with a cheaper “surrogate” model that is accurate over the entire regions of the model input parameters.

Surrogates can be generally categorized into three classes [63, 64]: data-fit models, reduced-order models, and hierarchical models. Data-fit models capture the input-output relationship of a model from available data points, and assume regularity by imposing interpolation or regression. Given the data points, it matters not how the original model functions, and it may be treated as a black box. One classical example is the Gaussian process [65, 66], while polynomial chaos (PC) expansion is another example that is also a spectral method [67, 68, 69, 70, 71, 72, 73, 74]. The latter builds a subspace from a set of orthogonal polynomial basis functions, and exploits the regularity in the dependence of model outputs on uncertain input parameters. Reduced-order models are based on a projection of the output space onto a smaller, lower dimensional subspace. One example is the proper orthogonal decomposition (POD), where a set of “snapshots” of the model outputs are used to construct a basis for the subspace [75, 76, 77]. Finally, hierarchical models are those where simplifications are performed based on the underlying physics. Techniques based on grid-coarsening, simplification of mechanics, addition of assumptions, are of this type, and are often the basis of multi-fidelity analysis [78].

3.4 Dynamic Programming and Approximate Dynamic Programming

Dynamic programming (DP) is an algorithm that provides a solution to the general problem of optimal decision making under uncertainty. Central to DP is the famous Bellman's equation [79], describing the relationship between cost or reward incurred immediately with the expected cost or reward of the uncertain future. DP has been used extensively in many major fields, including controls, operations research, and artificial intelligence, each with its own set of vocabulary and notation; we adopt mostly the conventions from the controls field in our work.

While the DP formulation is able to describe a very general set of problems, it is difficult to solve due to the "curse of dimensionality". Typically, only special classes of problems with specific structures, such as those with linear system and quadratic cost [80], have analytical solutions. Despite the extensive theoretical development of DP [81], its use in real-life applications is still at a conceptual level [82] as solutions are only possible through numerical approximations, and even then is a daunting task. As a result, substantial research has been devoted in making these numerical methods both practical and accurate, and this field is known as *approximate dynamic programming* (ADP) (a.k.a. neuro-dynamic programming and reinforcement learning) [83, 84, 85].

One source of "curse of dimensionality" is due to an often very large number of discrete states and controls, or if these variables are continuous. One approach to mitigate this overwhelming number of possible scenarios is to discretize [86] and aggregate [87] the states and controls. These techniques, however, are often applied to Markov decision processes where a system model is simply a transition matrix or function [88].

When a system *model* is available, the focus of ADP is usually on approximating the value functions (a.k.a. cost-to-go functions) in order to mitigate the exponential growth of scenarios. Value functions are often projected to a lower-dimensional subspace spanned by a set of "features" via a linear architecture. The construction and selection of features often rely on experience and insights into the problem. Once the features are fixed, one can then compute the coefficients of the approximation functions via algorithms such as backward induction or sample paths. In the former, projections are made through, for example, quadrature or Monte Carlo, based on the approximation function of the previous stage. Due to the recursive reliance on approximations, backward induction can lead to an exponential growth in the approximation error [89]. Sample paths (scenarios) are simulated in the latter approach, and they exploit the current approximation to explore the state space [83]. The approximation functions are then updated by algorithms such as recursive least squares [90], Kalman filter [91], and temporal differencing (TD) [92]. TD schemes in particular have received considerable attention and is supported by deep theoretical background [93, 84] and has seen substantial success in applications [94]. Exploitation alone in the sample path method can be dangerous as it may cycle only visited states; balance exploitation with exploration remains an important issue in ADP algorithms [95].

Another route of simplification is to form the problem using post-decision states [96, 97], which can be advantageous as it allows a model-free (i.e., without the need of the original system equation) formulation. Q-factors are a special type of post-decision state, and constructing approximation functions on Q-factors is known as Q-learning, perhaps the most widely used technique in reinforcement learning [98, 99].

4 Optimal Design Formulations

We are interested in designing multiple (N) experiments. The goal of these experiments is assumed to infer a finite number of model parameters of interest. Parameter inference is of course an integral part of calibrating models from experimental data [65]. The expected utility framework developed here can be generalized to other experimental goals as well.

We will formulate our experimental design criterion in a Bayesian setting. Bayesian statistics offers a foundation for inference from noisy, indirect, and incomplete data; a mechanism for incorporating physical constraints and heterogeneous sources of information; and a complete assessment of uncertainty in parameters, models, and predictions. The Bayesian approach also provides natural links to decision theory [100], a framework we will exploit in this work.

The Bayesian paradigm [101] treats unknown parameters as random variables; we will focus on the case of continuous parameters and variables. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, where Ω is a sample space, $\mathcal{F}$ is a σ-field, and $\mathbb{P}$ is a probability measure on $(\Omega, \mathcal{F})$. Let the vector of real-valued random variables[1] $\theta : \Omega \rightarrow \mathbb{R}^{n_\theta}$ denote the uncertain parameters of interest, i.e., the parameters to be conditioned on experimental data. θ is associated with a measure μ on $\mathbb{R}^{n_\theta}$, such that $\mu(A) = \mathbb{P}\left(\theta^{-1}\left(A\right)\right)$ for $A \in \mathbb{R}^{n_\theta}$. We then define $f(\theta) = d\mu/d\theta$ to be the density of θ with respect to Lebesgue measure. For the present purposes, we will assume that such a density always exists. Similarly we treat the data from the kth experiment ($k = 0, \ldots, N-1$): y_k, as an $\mathbb{R}^{n_y}$-valued random variable endowed with an appropriate density. $d_k \in \mathbb{R}^{n_d}$ denotes the design variables or experimental conditions for the kth experiment. Hence n_θ is the number of uncertain parameters, and n_y and n_d are the number of observations and design variables for each experiment, respectively. For simplicity, we assume n_y and n_d to be independent of k, but this certainly needs not be the case in this formulation. If one performs an experiment under conditions d and observes a realization of the data y, then the change in one's state of knowledge about the model parameters is given by Bayes' rule:

$$f(\theta|y, d, I) = \frac{f(y|\theta, d, I) f(\theta|d, I)}{f(y|d, I)} = \frac{f(y|\theta, d) f(\theta|I)}{f(y|d, I)}. \qquad (1)$$

For simplicity of notation, we shall use $f(\cdot)$ to represent all density functions; which specific distribution it corresponds to is reflected by its arguments (e.g., formally, $f(\theta|y, d, I)$ is $f_{\theta|y,d,I}(\theta|y, d, I)$). Here, I is the information vector representing all background information, including previous experimental design choices and measurements, if any. $f(\theta|d, I)$ is the prior density, $f(y|\theta, d, I)$ is the likelihood function, $f(\theta|y, d, I)$ is the posterior density, and $f(y|d, I)$ is the evidence. The second equality is due to the following reasonable assumptions:

- data from the current experiment are independent from the background information when conditioned on the uncertain parameters and the current experimental conditions—and so $f(y|\theta, d, I) = f(y|\theta, d)$; and
- knowing the design of an experiment without knowing its data does not affect our belief (i.e., the prior should not change based on what experiment we choose to do)—therefore $f(\theta|d, I) = f(\theta|I)$.

[1] For simplicity, we will use lower case to represent both the random variable and its realizations in this work.

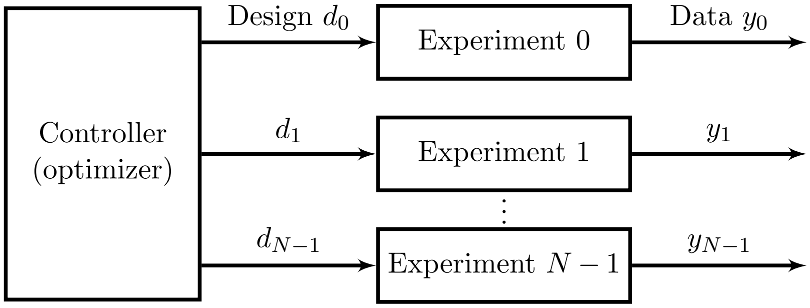


Figure 1: Open-loop design.

4.1 Open-Loop

The open-loop approach involves designing all N experiments concurrently. In other words, no results (data) from any of the experiments would be available in helping designing any other experiments—hence, there is no feedback of information (see Figure 1). This formulation then requires a single objective function to be optimized.

In forming the objective function, we take a decision theoretic approach [100]. We start with a *utility function* $u(d, y, \theta)$ that reflects the collective usefulness of the N experiments under conditions $d = \{d_0, \ldots, d_{N-1}\}$, given a particular set of the corresponding experimental outcomes $y = \{y_0, \ldots, y_{N-1}\}$ and particular value of the parameters θ. As we have only a distribution for the data y (parameterized by d) that may be observed, and a distribution of the parameters θ (the prior), we are interested in maximizing information gain *on average*, by taking the expectation. We then arrive at the *expected utility* $U(d)$:

$$U(d) \quad = \quad \int_{\mathcal{Y}} \int_{\mathcal{H}} u(d, y, \theta)\, f(\theta, y|d)\, d\theta\, dy = \int_{\mathcal{Y}} \int_{\mathcal{H}} u(d, y, \theta)\, f(\theta|y, d) f(y|d)\, d\theta\, dy, \tag{2}$$

where $\mathcal{H}$ is the support of $f(\theta)$, and $\mathcal{Y}$ is the support of $f(y|d)$. Note that we have omitted the conditioning on I for each density term for simplicity.

Our choice of utility function u is rooted in information theory. Following [19, 20], we set $u(d, y, \theta)$ to be

$$u(d, y, \theta) \equiv \int_{\mathcal{H}} f(\tilde{\theta}|y, d) \ln \left[\frac{f(\tilde{\theta}|y, d)}{f(\tilde{\theta})} \right] d\tilde{\theta} + g\left(f(\cdot), d, y\right). \tag{3}$$

The first term reflects the information gain in the experiment. This term involves an "internal" integration over the parameter space ($\tilde{\theta}$ is a dummy variable representing the parameters), therefore it is not a function of the parameters θ. It is equal to the relative entropy, or Kullback-Leibler (KL) divergence, from the posterior after all N experiments have been performed, to the prior before any experiment is performed. We choose to use the KL divergence for several reasons. First, the KL divergence is a special case of a wide range of divergence measures that satisfy the minimal set of requirements to be a valid measure of information on a set of experiments [102]. These requirements are based on the sufficient ordering (or "always at least as informative" ordering) of experiments, and are developed rigorously from likelihood ratio statistics, in a general setting without specifically targeting decision theoretic or Bayesian perspectives. Thus, the various measures satisfying these requirements, although different, do the job of reflecting information content of experiments. Second, the KL gives an intuitive indicator of the information gain in the sense of Shannon information [103]. Since the KL reflects the difference between two distributions, a large KL divergence from posterior to prior implies that the data y decrease entropy in θ by a large amount, and hence those data are more informative for parameter inference. Indeed, the KL divergence reflects the difference in information carried by two distributions in units of nats [103, 104], and also the expected information gain is equivalent to the *mutual information* between the parameters θ and the data y, given the experimental designs d. Third, such a formulation for general nonlinear models (nonlinear in the uncertain parameters) is consistent with linear experimental design theory based on the Fisher information matrix [11] in the sense that when a linear model is used, it simplifies to the linear D-optimality design. Finally, the use of information measure contrasts with a loss function in that, while the former does not target a particular task (such as estimation) in the context of a decision problem, it provides a general guidance of information gain that performs well for a wide range of tasks albeit not best for any particular task.

The second term is a reward functional, such as one that reflects the financial cost of the experiments, which can depend on data and designs, and our current state of belief about the parameters (i.e., the prior density function). The expected utility is thus

$$\begin{aligned} U(d) &= \int_{\mathcal{Y}}\int_{\mathcal{H}} u(d,y,\theta) f(\theta|y,d)\,d\theta\, f(y|d)\,dy \\ &= \int_{\mathcal{Y}}\int_{\mathcal{H}} \left\{ \int_{\mathcal{H}'} f(\tilde{\theta}|y,d) \ln\left[\frac{f(\tilde{\theta}|y,d)}{f(\tilde{\theta})}\right] d\tilde{\theta} + g(f(\cdot),d,y) \right\} f(\theta|y,d)\,d\theta\, f(y|d)\,dy \\ &= \int_{\mathcal{Y}} \left\{ \int_{\mathcal{H}'} f(\tilde{\theta}|y,d) \ln\left[\frac{f(\tilde{\theta}|y,d)}{f(\tilde{\theta})}\right] d\tilde{\theta} + g(f(\cdot),d,y) \right\} f(y|d)\,dy. \end{aligned} \tag{4}$$

To simplify notation, $\tilde{\theta}$ in Equation (4) is replaced by θ, yielding

$$\begin{aligned} U(d) &= \int_{\mathcal{Y}} \left\{ \int_{\mathcal{H}} \ln\left[\frac{f(\theta|y,d)}{f(\theta)}\right] f(\theta|y,d)\,d\theta + g(f(\cdot),d,y) \right\} f(y|d)\,dy. \qquad (5) \\ &= \mathbb{E}\left[D_{\mathrm{KL}}\left(f(\cdot|y,d)||f(\cdot)\right) + g(f(\cdot),d,y)\right], \end{aligned}$$

where again for simplicity, we use $\mathbb{E}$ to represent all expectation operators; which specific distribution it corresponds to is reflected by its arguments (e.g., formally, the $\mathbb{E}$ in the above equation is $\mathbb{E}_{y|d}$). The expected utility U is therefore a combination of the *expected information gain* in θ (first term) plus expected experimental rewards (second term).

Finally, the expected utility must be maximized over the design space $\mathcal{D}$ to find the optimal experimental design

$$d^* = \arg\max_{d\in\mathcal{D}} U(d). \tag{6}$$

From hereon, we will simply write $\max_d$ instead of $\max_{d\in\mathcal{D}}$ for all maximization operators, with the understanding that the design variable is from its corresponding design space.

4.2 Closed-Loop

The closed-loop approach involves designing and performing the experiments in sequence, such that one postpones the design of each experiment as late as possible, taking advantage of the feedback of

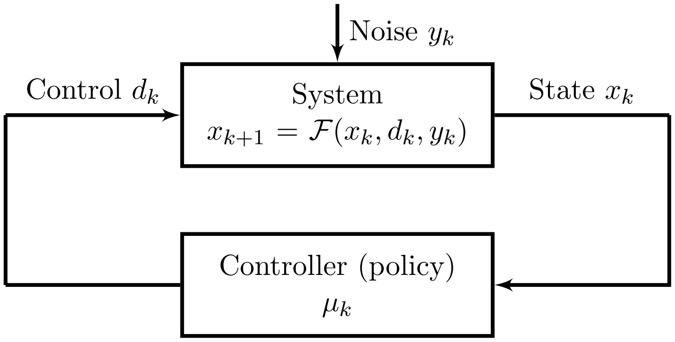


Figure 2: Closed-loop design.

information from previous experiments. Mathematically, the designs can now depend on available observed values of random variables. More generally, it is convenient to define a *state* variable x_k that captures all the information of the system necessary in making future decisions. In this case, it is natural to let the state be the PDF conditioned on all background information:

$$x_k = f(\theta|I_k). \tag{7}$$

In particular, the information vector I_k also contains all previous experiments and results $\{d_0, y_0, \ldots, d_{k-1}, y_{k-1}\}$. This conditional PDF fully describes our most current state of knowledge about the uncertain parameters. The state evolves according to the *system equation*, which is simply Bayes' rule from Equation (1):

$$x_{k+1} = \mathcal{F}(x_k, d_k, y_k) = \frac{f(y_k|\theta, d_k)x_k}{f(y_k|d_k, I_k)}. \tag{8}$$

We are then interested in finding the optimal decision *rule* (or *policy*) $\pi \equiv \{\mu_0(x_0), \mu_1(x_1), \ldots, \mu_{N-1}(x_{N-1})\}$ consisting a set of functions $\mu_k(x_k) = d_k$ telling us what design conditions to use depending on what the current system state is. The concept of the closed-loop design is illustrated in Figure 2.

The policy is optimal with respect to some underlying objective. Taking the same information-theoretic approach, we would like to ultimately maximize

$$J_\pi(x_0) = \int_{\mathcal{Y}} \left\{ \int_{\mathcal{H}} \ln\left[\frac{x_N}{x_0}\right] x_N \, d\theta + \sum_{k=0}^{N-1} g_k\left(x_k, \mu_k(x_k), y_k\right) \right\} f(y|x_0, \pi) \, dy. \tag{9}$$

where $x_k, k = 1, \ldots, N$ subject to the system equation (Equation (8)), and $g_k(\cdot)$ is the rewards function for the kth experiment. This objective is the same as the open-loop expected utility from Equation (5), except policy functions μ now replace design variables d.

4.2.1 Greedy

The greedy approach addresses the closed-loop formulation by solving a sequence of open-loop subproblems. Intuitively, it optimizes for the next experiment only, without taking into account the expected future gains (or even knowing how many future experiments are there) as a consequence of the decision at hand. Yet, it is still a closed-loop formulation since data from all previous experiments are taken into account.

The greedy policy is obtained from

$$\mu_k(x_k) = \arg\max_{d_k} \mathbb{E}\left[\int_{\mathcal{H}} \ln\left[\frac{x_{k+1}}{x_k}\right] x_{k+1}\, d\theta + g_k(x_k, d_k, y_k)\right], \tag{10}$$

where $x_k, k = 1, \ldots, N$ subject to the system equation (Equation (8)). There are a couple of advantages of the greedy approach. First, as we shall see after we introduce the Bellman equation in dynamic programming approach, the greedy formulation contains no recursion and thus one fewer source to the "curse of dimensionality". Second, the "ignorance" of future can sometimes be a blessing, for example, if the total number of experiments is unknown. However, the greedy method is, afterall, a simplification to the full objective function (Equation (9)), and is therefore a suboptimal policy.

4.2.2 Dynamic Programming

The dynamic programming (DP) formulation of the closed-loop approach is based on the idea of breaking the quantity in Equation (9) into a sequence of "tail" sub-problems. The centerpiece of this formulation is the Bellman equation that describe the maximization of the expected total future rewards in each of the sub-problems:

$$J_N(x_N) = \int_{\mathcal{H}} \ln\left[\frac{x_N}{x_0}\right] x_N\, d\theta \tag{11}$$

$$J_k(x_k) = \max_{d_k} \mathbb{E}\left[g_k(x_k, d_k, y_k) + J_{k+1}\left(\mathcal{F}(x_k, d_k, y_k)\right)\right] \tag{12}$$

for $k = 0, \ldots, N-1$. The J_k functions are thus known as the "reward-to-go" or "value" functions. The optimal policy is simply the arg-max: $\mu_k(x_k) = d_k^*$.

The Bellman's equation (11) and (12) optimize Equation (9) exactly and provides the optimal policy [81]. However, they have a recursive structure that involves nested maximization and expectation. This leads to an exponential growth in computation with the number of stages N (the "curse of dimensionality"). Analytic solution is almost always impossible to obtain except for a few simple class of problems. Numerical methods will be discussed in the next section.

5 Numerical Methods

We now describe the numerical methods used to solve the problems formulated in Section 4. Majority of the tools described in this section have been implemented by the author as past work, with the exception of the state representation in greedy and DP designs, and all but the "one-step lookahead backward induction" part of Section 5.2.2. Specific results from author's previous work will be labeled under "Work-To-Date" blocks. The methods not yet investigated are part of future work of this thesis project.

5.1 Open-Loop

Several pieces of numerical tools will be use for the open-loop approach. First, a Monte Carlo estimator is formed to approximate the expected utility. Second, stochastic optimization algorithms are used to efficiently finding the optimal designs. Third, polynomial chaos surrogates replace the forward model to help accelerate the computation and also make available gradients for the optimization process.

5.1.1 Expected Utility Estimator

Typically, the expected utility in Equation (5) has no closed form, even if the underlying forward model is replaced by a simpler surrogate (such as polynomials); it can only be approximated numerically. The posterior may be approximated using Laplace approximation [105], but the error can be high when the true distribution departs substantially from Gaussianity. Methods based on discretization in the parameter space, such as binning of samples [18] and tensor-product quadrature [106, 32] become impractical for no more than 3 or 4 dimensions. Kernel density estimation (KDE) [107] and k-nearest neighbor [26] methods build global functions from localized distributions (usually Gaussian) around samples. These methods can potentially achieve good performance, but are very sensitive to parameter tuning (such as the Gaussian bandwidth) and are not practical for high dimensions. Nonparametric methods such as particle representations used in [39] also suffers from dimensionality difficulties. Monte Carlo [21, 25] estimates, on the other hand, are simple and can handle much higher dimensions than other methods, and is the selected method in this thesis.

Following Ryan [25], $U(d)$ is first re-expressed as

$$\begin{aligned} U(d) &= \int_{\mathcal{Y}} \int_{\mathcal{H}} \ln \left[\frac{f(\theta|y,d)}{f(\theta)} \right] f(\theta|y,d)\, d\theta\, f(y|d)\, dy \\ &= \int_{\mathcal{Y}} \int_{\mathcal{H}} \ln \left[\frac{f(y|\theta,d)}{f(y|d)} \right] f(y|\theta,d)\, f(\theta)\, d\theta\, dy \\ &= \int_{\mathcal{Y}} \int_{\mathcal{H}} \left\{ \ln \left[f(y|\theta,d) \right] - \ln \left[f(y|d) \right] \right\} f(y|\theta,d)\, f(\theta)\, d\theta\, dy, \end{aligned} \tag{13}$$

where the second equality is due to the application of Bayes' theorem to the quantities both inside and outside the logarithm. In the special case where the Shannon entropy of $f(y,\theta|d)$ is independent of the design variables d, the first term in Equation (13) becomes constant for all designs [108] and can be dropped from the objective function. Maximizing the remaining term—which is the entropy of $f(y|d)$—is then equivalent to the maximum entropy sampling approach of Sebastiani and Wynn [22]. Here we retain the more general formulation of Equation (13) in order to accommodate, for example, likelihood functions containing a measurement error whose magnitude depends on y or d. Using Monte Carlo sampling to estimate the integrals, we obtain a doubly-nested Monte Carlo estimator [25]

$$U(d) \approx \hat{U}_{N,M}(d,\theta_s,y_s) \equiv \frac{1}{N} \sum_{i=1}^{N} \left\{ \ln \left[f(y^{(i)}|\theta^{(i)},d) \right] - \ln \left[\frac{1}{M} \sum_{j=1}^{M} f(y^{(i)}|\tilde{\theta}^{(i,j)},d) \right] \right\}, \tag{14}$$

where $\theta_s \equiv \left\{\theta^{(i)}\right\} \cup \left\{\tilde{\theta}^{(i,j)}\right\}$, $i = 1 \ldots N$, $j = 1 \ldots M$, are i.i.d. samples from the prior f; and $y_s \equiv \left\{y^{(i)}\right\}$, $i = 1 \ldots N$, are independent samples from the likelihoods $f(\cdot|\theta^{(i)},d)$. The variance of this estimator is approximately $A(d)/N + B(d)/NM$ and its bias is (to leading order) $C(d)/M$ [25], where A, B, and C are terms that depend only on the distributions at hand. While the estimator $\hat{U}_{N,M}$ is biased for finite M, it is asymptotically unbiased.

5.1.2 Stochastic Optimization

Maximizing U via a grid search over $\mathcal{D}$ is clearly impractical, since the number of grid points grows exponentially with dimension. Since only a Monte Carlo estimate $\hat{U}_{N,M}$ of the objective function is available, another naïve approach would be to use a large sample size (N, M) at each d and then apply a deterministic optimization algorithm, but this is still too expensive. (And even with large sample sizes, $\hat{U}_{N,M}$ is effectively non-smooth.) Instead, we would like to use only a *few* Monte Carlo samples to evaluate the objective at any given d, and thus we need algorithms suited to noisy objective functions. In particular, we focus on algorithms for continuous design spaces.

Stochastic Approximation

(a) *Robbins-Monro*: The main variation of stochastic approximation is the Robbins-Monro (RM) algorithm [50], which has the iterative update:

$$x_{j+1} = x_j - a_j \hat{g}(x_j). \tag{15}$$

Here $\hat{g}(x_j)$ is an unbiased gradient estimator. The gain sequence a_j should satisfy the following properties

$$\sum_{j=0}^{\infty} a_j = \infty \quad \text{and} \quad \sum_{j=0}^{\infty} a_j^2 < \infty. \tag{16}$$

One natural choice is the harmonic step size sequence $a_j = \beta/j$, where β is some appropriate scaling constant. With various technical assumptions on $\hat{g}$ and g, it can be shown that RM converges to the exact solution almost surely [48].

Choosing the sequence a_j is often viewed as the Achilles' heel of RM, as the algorithm's performance can be very sensitive to step size. We acknowledge this fact and do not downplay the difficulty of choosing an appropriate gain sequence, but there exist logical approaches to selecting a_j that yield reasonable performance. More sophisticated strategies, such as search-then-converge learning rate schedules [109], adaptive stochastic step size rules [110], and iterate averaging methods [111, 48], have been developed and successfully demonstrated in applications.

(b) *Kiefer-Wolfowitz*: Kiefer-Wolfowitz (KW) stochastic approximation [46] is a special case of RM where the estimator $\hat{g}(x_j)$ is approximated using finite difference. Under certain regularity conditions, it is guaranteed to converge. The main drawback of KW is the potentially estimator high variance as well as high computational cost due to the need of 2^{n_d} objective evaluations.

(c) *Simultaneous Perturbation Stochastic Approximation*: Motivated to mitigate the 2^{n_d} objective evaluations from KW, Spall proposed the simultaneous perturbation stochastic approximation (SPSA) [47, 112]. The key of this method is that it approximates gradient with only two *random* perturbations instead of 2^{n_d}, regardless of the problem's dimension. An intuitive justification for SPSA is that error in the gradient "averages out" over a large number of iterations [47]. Convergence proofs with varying conditions and assumptions can be found in [113, 114, 115]. Randomness introduced through the noisy objective $\hat{U}_{N,M}$ and the finite-difference-like perturbations allows for a global convergence property [116].

Sample Average Approximation

The central idea of SAA is to reduce the stochastic optimization problem to a deterministic problem, by fixing the noise throughout the entire optimization process. If the noise is design-dependent, it is first transformed to a design-independent random variable, which can always possible in practice since the random numbers in any computation are fundamentally generated from uniform random (or really pseudo-random) numbers. The noise variables at different d then share a common distribution, and a common set of realizations is employed at all values of the design variable.

SAA approximates the true optimization problem under a particular set of realizations of the noise random variable. The *same* set of realizations is used for different values of d during the optimization process, thus making the minimization problem deterministic. (One can view this approach as an application of common random numbers.) A deterministic optimization algorithm can then be chosen to find the optimum of the deterministic problem as an approximation to the true optimum. Under certain assumptions on the objective function and the design space, the optimal design and objective estimates in SAA generally converge to their respective true values *in distribution* at a rate of $1/\sqrt{N}$ [49, 53]. Stochastic bounds are available to construct an estimate to the optimality gap (difference between the true optimal value and the optimal from the SAA instance) [59, 60]. One could use the optimality gap estimator and its variance to decide whether more runs are required, or which approximate optimal designs are most trustworthy.

At this point, we have reduced the stochastic optimization problem to a series of deterministic optimization problems; a suitable deterministic optimization algorithm is still needed to solve them. One popular candidate is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [117]. BFGS is a gradient-based method for solving deterministic nonlinear optimization problems, widely used for its robustness, ease of implementation, and efficiency. It is a quasi-Newton method, iteratively updating an approximation to the (inverse) Hessian matrix from objective and gradient evaluations at each stage. BFGS is shown to converge super-linearly to a local minimum if a quadratic Taylor expansion exists near that minimum [117]. The limited memory BFGS (L-BFGS) [117] method can also be used when the design dimension becomes very large (e.g., more than 10^4), such that the dense inverse Hessian cannot be stored explicitly.

Other Methods

(a) *Nelder-Mead*: The Nelder-Mead (NM) nonlinear simplex algorithm [45] is a gradient-free method that has been well studied and is widely used for deterministic optimization. The details of the algorithm are thus omitted from this discussion but can be found, e.g., in [45, 118, 119]. This algorithm has a natural advantage in dealing with noisy objective functions because it requires only a *relative ordering* of function values, rather than the magnitudes of differences (as in estimating gradients). Minor modifications to the algorithm parameters can improve optimization performance for noisy functions [118]. Constraints in NM are handled simply by projecting from the infeasible point to the nearest feasible point.

■ **Work-To-Date**: The author has made extensive empirical study on several stochastic optimization algorithms: SPSA and NM in a hydrogen-oxygen autoignition kinetic parameter inference problem [7], and RM and SAA-BFGS in an optimal sensor placement challenge for diffusion source inversion [9]. Figure 3 presents results from the second application, which indicates better performance from algorithms that take advantage of available gradient information.

We note that the gradient information used in RM and SAA-BFGS are not the exact gradient of $U(d)$. Such computation is generally not possible given that we only have a Monte Carlo estimator of $U(d)$. Instead, we make use of infinitesimal perturbation analysis [120, 121, 122] to interchange the integration and differentiation operators in order to take advantage of the gradients of the

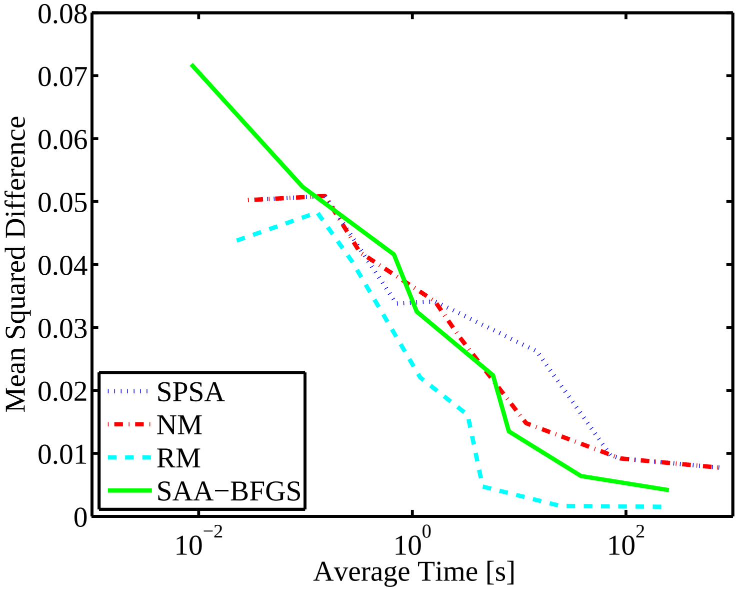


Figure 3: Mean squared error versus computational time of SPSA, NM, RM, and SAA-BFGS algorithms applied to an optimal sensor placement problem in a 2D scalar diffusion field for the purpose of inferring the source location. Methods that make use of gradient information, namely, RM and SAA-BFGS, achieve lower error levels. More details can be found in [9].

Monte Carlo realizations. However, even then the differentiation of the Monte Carlo estimator can be expensive as it would involve either finite difference of the forward model (typically simulation of a physical process governed by PDEs) or its adjoint problem. We thus build a surrogate for the forward model, using polynomial chaos (described below). The analytical gradient of the estimator based on the surrogate can then be derived and evaluated extremely cheaply. This is what is used in RM and SAA-BFGS. More details can be found in [9]. ■

5.1.3 Polynomial Chaos Approximation

Even with an effective Monte Carlo estimator of the expected utility and efficient algorithms for stochastic optimization, the complex physical model (often systems of PDEs) embedded in Equation (14) still must be evaluated repeatedly, over many values of the model parameters and design variables. It is then useful to replace these forward models with surrogates that are computationally cheaper. Mathematical models of the experiment $G(\theta, d)$ enter the inference and design formulation through the likelihood function $f(y|\theta, d)$, for example, in an additive fashion

$$y = G(\theta, d) + \epsilon, \tag{17}$$

where ϵ is the noise random variable with some known distribution. To make these calculations tractable, one would like to replace the forward model with a cheaper "surrogate" model that is accurate over the entire prior support $\mathcal{H}$ and the entire design space $\mathcal{D}$.

While many model surrogacy options exist (Section 3.3), we decide to use polynomial chaos approximation. Polynomial chaos has seen extensive use in a range of engineering applications (e.g., [123, 5, 124, 125]) including parameter estimation and inverse problems (e.g., [126, 127, 128]).

More recently, it has also been used in open-loop optimal Bayesian experimental design [35, 7], with excellent accuracy and multiple order-of-magnitude speedups over direct evaluations of forward model.

Any random variable v with finite variance can be represented by an infinite series

$$v = \sum_{|\mathbf{i}|=0}^{\infty} a_{\mathbf{i}} \Psi_{\mathbf{i}}(\xi_1, \xi_2, \ldots), \tag{18}$$

where $\mathbf{i} = (i_1, i_2, \ldots)$, $i_j \in \mathbb{N}_0$, is an infinite-dimensional multi-index; $|\mathbf{i}| = i_1 + i_2 + \ldots$ is the l_1 norm; $a_{\mathbf{i}} \in \mathbb{R}$ are the expansion coefficients; ξ_i are independent random variables; and

$$\Psi_{\mathbf{i}}(\xi_1, \xi_2, \ldots) = \prod_{j=1}^{\infty} \psi_{i_j}(\xi_j) \tag{19}$$

are multivariate polynomial basis functions [69]. Here ψ_{i_j} is an orthogonal polynomial of order i_j in the variable ξ_j, where orthogonality is with respect to the density of ξ_j,

$$\mathbb{E}\left[\psi_m(\xi)\psi_n(\xi)\right] = \int_{\Xi} \psi_m(\xi)\psi_n(\xi) f(\xi)\, d\xi = \delta_{m,n} \mathbb{E}\left[\psi_m^2(\xi)\right], \tag{20}$$

and Ξ is the support of $f(\xi)$. The expansion (18) is convergent in the mean-square sense [129]. For computational purposes, the infinite sum in (18) must be truncated to some finite stochastic dimension n_s and a finite number of polynomial terms. A common choice is the "total-order" truncation $|\mathbf{i}| \leq p$, but other truncations that retain fewer cross terms, a larger number of cross terms, or anisotropy among the dimensions are certainly possible [74].

In the optimal Bayesian experimental design context, the model outputs depend on both the parameters and the design variables. Constructing a new polynomial expansion at each value of d encountered during optimization is generally impractical. Instead, we can construct a *single* PC expansion for each component of G, depending jointly on θ and d [7]. To proceed, we assign one stochastic dimension to each component of θ and one to each component of d. Further, we assume an affine transformation between each component of d and the corresponding ξ_i; any realization of d can thus be uniquely associated with a vector of realizations ξ_i. Since the design variables will usually be supported on a bounded domain (e.g., inside some hyper-rectangle), the corresponding ξ_i are endowed with uniform distributions. The associated univariate ψ_i are thus Legendre polynomials. These distributions effectively define a uniform weight function over the design space $\mathcal{D}$ that governs where the L^2-convergent PC expansions should be most accurate.

Constructing the PC expansion involves computing the coefficients $a_{\mathbf{i}}$. This computation generally can proceed via two alternative approaches, intrusive and non-intrusive. The intrusive approach results in a new system of equations that is larger than the original deterministic system, but it needs be solved only once. The difficulty of this latter step depends strongly on the character of the original equations, however, and may be prohibitive for arbitrary nonlinear systems. The non-intrusive approach computes the expansion coefficients by directly projecting the quantity of interest (e.g., the model outputs) onto the basis functions $\Psi_{\mathbf{i}}$. One advantage of this method is that the deterministic solver can be reused and treated as a black box. The deterministic problem then needs to be solved many times, but typically at carefully chosen parameter and design values. The non-intrusive approach also offers flexibility in choosing arbitrary functionals of the state trajectory as observables; these functionals may depend smoothly on ξ even when the state itself has a less regular dependence. Here, we will employ a non-intrusive approach.

Applying orthogonality, the PC coefficients are simply

$$G_{c,\mathbf{i}} = \frac{\mathbb{E}\left[G_c(\theta(\xi), d(\xi))\Psi_{\mathbf{i}}(\xi)\right]}{\mathbb{E}\left[\Psi_{\mathbf{i}}^2(\xi)\right]} = \frac{\int_\Xi G_c(\theta(\xi), d(\xi))\Psi_{\mathbf{i}}(\xi) f(\xi)\, d\xi}{\int_\Xi \Psi_{\mathbf{i}}^2(\xi) f(\xi)\, d\xi}, \tag{21}$$

where $G_{c,\mathbf{i}}$ is the coefficient of $\Psi_{\mathbf{i}}$ for the cth component of the model outputs. Analytical expressions are available for the denominators $\mathbb{E}\left[\Psi_{\mathbf{i}}^2(\xi)\right]$, but the numerators must be evaluated numerically. When the evaluations of the integrand (and hence the forward model) are expensive and n_s is large, an efficient method for numerical integration in high dimensions is essential.

To evaluate the numerators in (21), we employ Smolyak sparse quadrature based on one-dimensional Clenshaw-Curtis quadrature rules [130]. Care must be taken to avoid significant aliasing errors when using sparse quadrature to construct polynomial approximations, however. Indeed, it is advantageous to recast the approximation as a Smolyak sum of constituent full-tensor polynomial approximations, each associated with a tensor-product quadrature rule that is appropriate to its polynomials [131, 74]. This type of approximation may be constructed *adaptively*, thus taking advantage of weak coupling and anisotropy in the dependence of G on θ and d. More details can be found in [74].

At this point, we may substitute the polynomial approximation of G into the likelihood function $f(y|\theta, d)$, which in turn enters the expected information gain estimator (14). This enables fast evaluation of the expected information gain.

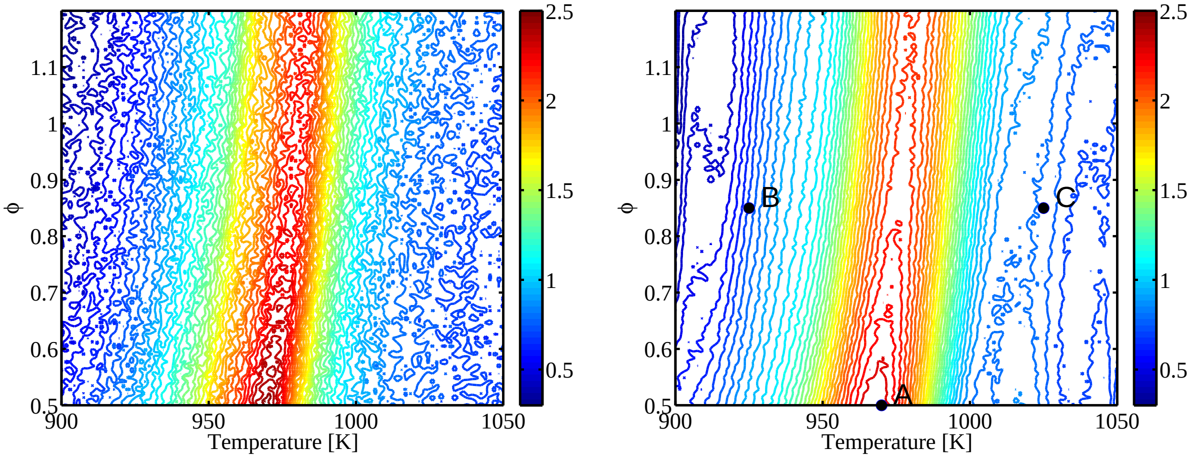
As mentioned earlier, another important contribution of polynomial chaos surrogate is the simple polynomial structure provides easy access to gradient information that can be used for gradient-based stochastic optimization methods such as RM and SAA-BFGS. Details can be found in [9].

■ **Work-To-Date**: From [7], Figure 4 shows the comparison of the expected utility contours in a hydrogen-oxygen autoignition kinetic parameter inference problem, using the full ODE model and PC surrogate. The inference problem is solved at a number of experimental conditions, and the resulting posteriors are shown in Figure 5. Results indicate PC surrogates can achieve high accuracy for both optimal experimental design and parameter inference problems. ■

5.2 Closed-Loop

5.2.1 Greedy

The numerical methods used in the greedy approach are almost exactly the same as those from the open-loop formulation, since greedy design involves a sequence of multiple open-loop sub-problems. But there is one important difference. In the open-loop approach, the prior is usually easily represented or approximated by some parametric form of distribution such as uniform or Gaussian. The posterior from inference is then usually represented by samples, from methods such as MCMC. In the greedy approach, the next experiment uses the previous experiment's posterior as its prior. In particular, we need to sample from this new prior for the next experiment's inference. We do not want to bound by the available MCMC samples, since the number of such samples is limited, and we would not explore certain regions of the parameter space. It would be better if we can approximately represent the distribution in a manner using those samples, such that we can sample from this approximate distribution. Some example include Gaussian mixture model fitting [132], particle representation such as sequential Monte Carlo [133], and random variable mapping [134]. Please refer to Section 7.2.2 for a more detailed discussion.



(a) Full ODE model

(b) PC surrogate

Figure 4: Estimated expected utility contours in a single-experiment hydrogen-oxygen autoignition setting, where the ignition delay time is used to infer two uncertain kinetic parameters. The design variables are the initial temperature and equivalence ratio ϕ. The contours are constructed using the full ODE model and the PC surrogate with total order $p = 12$ and 10^6 isotropic sparse quadrature evaluations. The contours have good agreement, and captures all key features of the surface. The smoother PC surface is a result of a much higher Monte Carlo size being used since the surrogate is much cheaper to evaluate. Inference problems are then solved at experimental conditions A, B, and C to validate the experimental design procedure, with the posteriors shown in Figure 5. More details can be found in [7].

5.2.2 (Approximate) Dynamic Programming

The Bellman equation from the DP formulation (Section 4.2.2) cannot be solved analytically except for a few special classes of problems such as linear-quadratic DP problems. In general, they must be solved numerically. The numerical techniques for solving the DP formulation are known as approximate dynamic programming (ADP) methods. For contrasting with the other optimal experimental design approaches, we will use ADP and DP interchangeably, where whether it is analytical DP or numerical ADP should be clear from context.

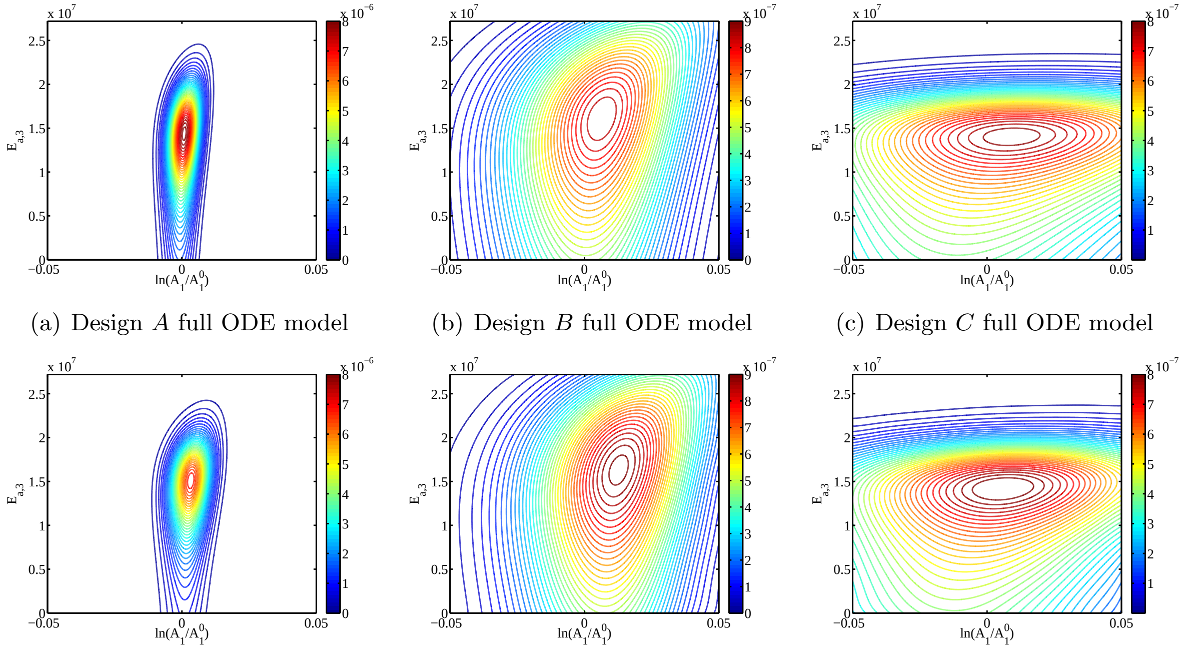
The ADP methods can be categorized into three groups:

1. problem simplification,
2. value function approximation, and
3. other techniques.

Problem Simplification

One approach of ADP is to simplify the DP formulation into something more manageable. Open-loop and greedy designs are of this type, as the former eliminates feedback and the latter ignores future costs. We now discuss other ways to simplify the DP problem to varying degrees.

(a) *Myopic Design*: In a myopic design, Equation (12) only retains the stage cost g_k and ignores all



(a) Design A full ODE model (b) Design B full ODE model (c) Design C full ODE model

(d) Design A PC surrogate (e) Design B PC surrogate (f) Design C PC surrogate

Figure 5: Contours of posterior density of the kinetic parameters, showing the results of inference with data obtained at three different experimental conditions (designs A, B, and C). The top row figures are posteriors constructed using the full ODE model, while the bottom row are those via the PC surrogate with $p = 12$, $n_{\text{quad}} = 10^6$; they show excellent agreement. Note that these results are under a particular realization of the uncertain parameters, but the experimental design framework optimizes for the *expected* information gain, over the entire (prior) distribution of the parameters. It might appear that the optimal design minimized the variance of the posterior from these figures, but in fact the formulation we used utilizes the expected KL divergence as the objective. A small expected KL usually also leads to a small expected variance in the posterior in practice. More details can be found in [7].

future costs. This design is similar to greedy design in concept, but in our current formulation of the optimal experimental design problem (Equations (11) and (12)), myopic design would not be able to capture any information gain until the very last experiment, since our information measure is included only in the terminal cost. In fact, our greedy design corresponds to the myopic design of a different formulation that involves incremental information gain. This formulation is discussed in Section 7.1.3.

(b) *Rolling Horizon*: The rolling horizon method uses a shorter horizon of length $M < N$. As experiments are performed, it solves a new DP problem of horizon M from the new time. The full N-stage DP problem is effectively broken down into a series of smaller, more manageable M-stage DP problems. However, too small of an M might not capture enough future influences, while too large of an M might still be too expensive to solve.

(c) *Certainty Equivalent Control*: Certainty equivalent control simplifies the DP formulation by fixing the random variables y_k at “typical” values $\bar{y}_k$ such that a stochastic DP problem is turned into a deterministic one:

$$\tilde{J}_k(x_k) = \max_{d_k} g_k(x_k, d_k, \bar{y}_k) + \tilde{J}_{k+1}\left(\mathcal{F}(x_k, d_k, \bar{y}_k)\right), \tag{22}$$

where $\tilde{J}_k$ is the optimal cost of the corresponding deterministic problem. Both open-loop and closed-loop formulations yield the same optimal designs for a deterministic problem. This approximation can then be solved via either online (sequential) or offline (batch) approaches. A variation of this technique involves leaving some uncertain quantities as stochastic and fixing some design variables (usually later stages) with a heuristic policy. The resulting value functions $\tilde{J}_k$ can also be used for value function approximation in the fully-stochastic problem. The main (and often fatal) disadvantage of this method is that since g_k and J_{k+1} are almost always nonlinear functions in y_k, the expectation of these functions are not the same as (and in fact can be very different than) the functions applied to the expectation of y_k. These errors can quickly grow with the number of stages.

(d) *Open-Loop Feedback Control*: Open-loop feedback control involves performing open-loop design on all *remaining* experiments in a sequence of experiments, while only carrying out the immediately next experiment. Upon completing experiment k, we update the state to $x_{k+1} = \mathcal{F}(x_k, d_k, y_k)$, and solve the open-loop tail problem

$$\max_{d_{k+1},\ldots,d_{N-1}} \mathbb{E}\left[J_N(x_N) + \sum_{i=k+1}^{N-1} g_k\left(x_k, d_k, y_k\right)\right] \tag{23}$$

subject to $x_{i+1} = \mathcal{F}(x_i, d_i, y_i), i = k+1, \ldots, N-1$. Once the solution $d_{k+1}, \ldots, d_{N-1}$ is obtained, we conduct the next experiment at d_{k+1} only and repeat the process. The open-loop feedback control is proven to be at least as good as an optimal open-loop policy. Variations include carrying out batches of designs at a time, and mixing open-loop and closed-loop designs (partial open-loop feedback control).

(e) *State Aggregation*: State aggregation [87] involves grouping states that share similar properties or features, and forming a new problem based on the smaller or lower-dimensional aggregated state space. One special case of aggregation for a continuous state space is simply a discretization of the state space.

The formulation typically involves defining the following.

- Aggregation and disaggregation transition probability functions under a given control. These functions must satisfy the probability axioms.
- System equation and stage costs for the aggregation problem.

The main difficulties of this method are its heavy reliance on heuristics and understanding of the original problem in defining the transition probability functions, and the computation of the aggregation system equation and stage costs.

Value Function Approximation

The methods discussed in this subsection center around approximating the value functions J_k. Observing Equation (12), the main difficulty of DP lies within the recursive nature of the J_k

functions. If they can be decoupled, then the exponential grow of cost with respect to N can be reduced to linear growth. This can be achieved by replacing J_k with approximations $\tilde{J}_k$ where $\tilde{J}_k$ can be readily evaluated without the need of computing J_{k+1} or $\tilde{J}_{k+1}$.

There are several options in representing $\tilde{J}_k$ in a continuous state space.

1. Use tabular storage of J_k values at a set of discretization points of the state space. However, the storage requirement grows exponentially with the dimension of the state space, and linearly with horizon.

2. More popularly, parametric functions are used. For example, a linear approximation architecture has the form

$$\tilde{J}_k(x_k) = \sum_i r_{k,i} z_{k,i}(x_k), \tag{24}$$

   where $r_{k,i}$ is the ith coefficient or weight of $\tilde{J}_k$, and $z_{k,i}$ is the ith feature or basis function of the state x_k. The two main tasks are to select the features and to train the weights. The choice of features is a very important and difficult task. While there exists algorithms for selecting the best features from a pool of candidates, it is much harder to extract or create new features from data [135]. The feature functions are often picked based on experience, trial-and-error, and knowledge of the problem structure. Many different options exist for training the weights. A simple and popular method is the least-squares regression (e.g., [136]), effectively solving the optimization problem of

$$\min_{r_k} \| J_k(x_{k,s}) - Z_k(x_{k,s}) r_k \|_2 \tag{25}$$

   where $x_{k,s}$ is a set of sample points (e.g., from Monte Carlo or quadrature), $Z_k(x_{k,s})$ is the matrix containing all the feature functions evaluated at $x_{k,s}$, and r_k is the vector of unknown coefficients. Such an (usually) over-determined system can be easily solved using existing linear algebra routines, for example, via singular value decomposition [136, 137].

3. Take value functions from a related, simpler problem (e.g., using certainty equivalent control, or from simplified constraints or dynamics). However, this also needs to be expressed in a readily-evaluatable form such as tabular storage or parametric form.

We shall denote Π the approximation operator, such that $\tilde{J}_k = \Pi J_k$. For example, Π can be a projection operator.

(a) *Limited Lookahead*: The idea of the limited lookahead method is simply to replace the value functions with approximations. The one-step lookahead policy is obtained by solving

$$\max_{d_k} \mathbb{E}\left[g_k(x_k, d_k, y_k) + \tilde{J}_{k+1}\left(\mathcal{F}(x_k, d_k, y_k)\right)\right] \tag{26}$$

where $\tilde{J}_{k+1}$ is approximation to the true value function J_{k+1}. In other words, one step of DP is performed before an approximation function is used. Similarly, a two-step lookahead policy involves an additional step

$$\tilde{J}_{k+1}(x_{k+1}) = \max_{d_{k+1}} \mathbb{E}\left[g_{k+1}(x_{k+1}, d_{k+1}, y_{k+1}) + \tilde{J}_{k+2}\left(\mathcal{F}(x_{k+1}, d_{k+1}, y_{k+1})\right)\right] \tag{27}$$

where $\tilde{J}_{k+2}$ is now the approximation to the true value function J_{k+2}. Two steps of DP are performed before using an approximation. An ℓ-step lookahead policy extends similarly. However, it is in general not true that a higher ℓ being better.

In the special case where $\tilde{J}_k$ are the value functions of a suboptimal policy applied to the same problem, it is known as the rollout algorithm.

(b) *Rollout Algorithm*: Here $\tilde{J}_k$ are the value functions of some suboptimal policy, called a *base heuristic*. It can be shown that [81], if for all x_k and k,

$$\max_{d_k} \mathbb{E}\left[g_k(x_k, d_k, y_k) + \tilde{J}_{k+1}\left(\mathcal{F}(x_k, d_k, y_k)\right)\right] \geq \tilde{J}_k(x_k), \tag{28}$$

then the value functions $\bar{J}_k$ corresponding to a one-step lookahead policy based on $\tilde{J}_k$ satisfy for all x_k and k

$$\bar{J}_k(x_k) \geq \max_{d_k} \mathbb{E}\left[g_k(x_k, d_k, y_k) + \tilde{J}_{k+1}\left(\mathcal{F}(x_k, d_k, y_k)\right)\right]. \tag{29}$$

If we update our policy iteratively in a "rollout" fashion where $\bar{\mu}_k$ is the base heuristics, then

$$\begin{aligned} \tilde{J}_k &= \mathbb{E}\left[g_k(x_k, \bar{\mu}_k(x_k), y_k) + \tilde{J}_{k+1}\left(\mathcal{F}(x_k, \bar{\mu}_k(x_k), y_k)\right)\right] \\ &\leq \max_{d_k} \mathbb{E}\left[g_k(x_k, d_k, y_k) + \tilde{J}_{k+1}\left(\mathcal{F}(x_k, d_k, y_k)\right)\right], \end{aligned}$$

and the conditions for Equation (28) is satisfied. Hence, rollout policy achieves no worse cost than its base heuristics! In fact, the rollout algorithm is just a single step of policy iteration. If conducted repeatedly, it is guaranteed to converge to the optimal policy under finite number of steps [81].

The choice of base heuristic is quite flexible and can be any valid policy. The two main difficulties of rollout are: evaluating $\tilde{J}_k$ from the base heuristic may be computationally intensive, and representing $\tilde{J}_k$ in a parametric fashion may be inaccurate.

(c) *One-step Lookahead Backward Induction*: A straightforward way to build the value function approximations is to use backward induction based on one-step lookahead. The approximation functions are then

$$\tilde{J}_N(x_N) = \Pi J_N(x_N) \tag{30}$$

$$\tilde{J}_k(x_k) = \Pi \max_{d_k} \mathbb{E}[g_k(x_k, d_k, y_k) + \tilde{J}_{k+1}\left(\mathcal{F}(x_k, d_k, y_k)\right)]. \tag{31}$$

Note that $\tilde{J}_k$ is built based on the previous approximation function $\tilde{J}_{k+1}$. Consequently, the approximation error can quickly aggregate, and in fact potentially at an exponential rate [89]. Once the approximation functions are constructed, the optimal policy can then be approximated by solving

$$\tilde{\mu}_k(x_k) = \arg\max_{d_k} \mathbb{E}[g_k(x_k, d_k, y_k) + \tilde{J}_{k+1}\left(\mathcal{F}(x_k, d_k, y_k)\right)]. \tag{32}$$

■ **Work-To-Date**: The DP numerical results presented in Section 6 are based on the one-step lookahead backward induction. In particular, we use a linear architecture where the feature vectors are polynomials of the state components. The maximization of the expectation is evaluated using stochastic optimization methods described in Section 5.1.2. The projection onto the approximation subspace is done using least-square regression with Monte Carlo samples from the prior. ■

(d) *Forward Trajectory Simulation*: Another way to construct the approximation functions is to use simulated forward trajectories (sample paths). One simple algorithm provided by Powell [83] is shown in Algorithm 1. The method for updating the (continuous) approximation function can be done by using, for example, recursive least squares [90], Kalman filtering [91], and temporal differencing [92].

Algorithm 1: ADP algorithm based on iteratively updating value function approximations from forward trajectory simulation. [83].

```
Initialize $\{\tilde{J}_k^0\}$, perhaps via choices of coefficients $\{r_{k,i}\}$;
Choose an initial state $x_0^1$;
$n = 1$;
for $n = 1, \dots, T$ do
    Choose sample path $\{y_k^n\}$;
    for $k = 0, \dots, N$ do
        Solve $\tilde{J}_k^n(x_k) = \max_{d_k} \mathbb{E}\left[g_k(x_k, d_k, y_k) + \tilde{J}_{k+1}^{n-1}(\mathcal{F}(x_k, d_k, y_k))\right]$ and let $d_k^n$ be the value of $d_k$ that solves the maximization problem;
        Update $\tilde{J}_k^{n-1}(x_k)$ using $\tilde{J}_k^n(x_k)$ ;
    end
end
```

There are several advantages of this construction. First, the forward simulations are cheap to conduct, since Equation (32) only needs to be evaluated at one x_k as we know the realization of y_k observed. Second, there is an automatic exploration of the state space, where the sample paths would visit states that are more likely to occur according to the approximation functions. This phenomenon is known as exploitation. In contrast, the backward induction method has no such information, and the sample points used in regression may cover large regions of the state space that are rarely or never visited. Third, this approach is more flexible in stopping the algorithm, as at any point it will be able to provide a full set of approximation functions. Backward-induction, on the other hand, will only able to provide a full set of approximation functions when it has reached $k = 0$ and thus completed all required computations. Similarly, forward simulation allows further refinement of the approximation functions should the designer find them unsatisfactory, while this cannot be easily achieved with backward induction.

The biggest drawback of this approach is the issue of exploration versus exploitation. As stated earlier, one of the advantages of this method is the exploitation of the approximation functions. However, the sample paths might enter a cycle where the states being visited merely reinforce and amplify the good and bad regions according to the current approximation functions, and the approximations may not converge to the true value functions. Thus, it is crucial to reach a balance between exploitation and exploration by introducing additional and occasional visits to states that appear unlikely to be visited. More discussion can be found in Section 7.2.1.

Other Techniques

(a) *Post-Decision State and Q-Factors*: In the traditional DP formulation, the next state evolves according to the system equation $x_{k+1} = \mathcal{F}(x_k, d_k, y_k)$. It can be beneficial to split the system equation into two stages

$$\begin{aligned} x_k^p &= \mathcal{F}_A(x_k, d_k) && (33)\\ x_{k+1} &= \mathcal{F}_B(x_k^p, y_k) = \mathcal{F}_B(\mathcal{F}_A(x_k, d_k), y_k), && (34) \end{aligned}$$

where x_k^p is the post-decision state of stage k. If this formulation is possible, then we can construct an approximation to the expectation of the value function taking the post-decision state as the argument

$$\tilde{J}_k^p(x_k^p) \approx \mathbb{E}\left[J_k\left(\mathcal{F}_B(x_k^p, y_k)\right)\right]. \tag{35}$$

The advantage of the post-decision state is that it separates the noise variable y_k from the arguments of the system equation, and so we are able to build an approximation function that includes the expectation operator. In the traditional formulation, $\tilde{J}_k(x_k)$ does not have explicit y_k dependence, and so incorporating the expectation within the approximation function would be impossible. Another advantage is that once the approximation functions $\tilde{J}_k^p$ are constructed, we no longer need the system equation $\mathcal{F}$, and only require $\mathcal{F}_A$. This is the so-called "model-free" approach.

Is it always possible to split $\mathcal{F}$ into $\mathcal{F}_A$ and $\mathcal{F}_B$. A neat split may very well depend on the structure and nature of the problem. Generally for any model, one can always let $\mathcal{F}_A(x_k, d_k) = \{x_k, d_k\}$ be a simple concatenation between the state and decision variables. This special case is known as the method of Q-factors, described with detail in Section 7.2.1.

6 Example: Linear-Gaussian Model

In this section, we investigate the optimal experimental design problem for a simple linear-Gaussian model where analytical results are possible to obtain under certain situations. This exercise serves two purposes: (1) to validate numerical results against analytical solutions; and (2) to provide motivation for potential substantial gains in using the closed-loop DP design.

Consider the following 1D linear-Gaussian model

$$y_k = d_k\theta + \epsilon_k, \tag{36}$$

where k is the experiment index, y_k are the noisy measurements, $d_k \in [1, 10]$ are the design variables, θ is the uncertain parameter we would like to infer, and $\epsilon_k \sim \mathcal{N}(0, \sigma_{\epsilon,k}^2)$ are zero-mean Gaussian noise variables. Given a prior of $\theta \sim \mathcal{N}(s_0, \sigma_0^2)$, then all subsequent conditional distributions (posteriors) will also be Gaussian due to conjugacy. We shall let the state of the DP formulation to be the mean and variance of these distributions: $x_k = (s_k, \sigma_k^2)$. The system equation is then a simple application of Bayes' Theorem:

$$x_{k+1} = (s_{k+1}, \sigma_{k+1}^2) = \mathcal{F}(x_k, d_k, y_k) = \left(\frac{y_k d_k \sigma_k^2 + s_k \sigma_{\epsilon,k}^2}{d_k^2\sigma_k^2 + \sigma_{\epsilon,k}^2}, \frac{\sigma_k^2\sigma_{\epsilon,k}^2}{d_k^2\sigma_k^2 + \sigma_{\epsilon,k}^2}\right). \tag{37}$$

Specifically, let there be a total of $N = 2$ experiments, and the prior be $\mathcal{N}(7, 9)$. The desired objective to be maximized is the final expected information gain (expected KL divergence) plus an experimental cost in the form of a quadratic penalty in the controls: $g_k(x_k, d_k, y_k) = -0.01d_k^2$.

6.1 Constant Noise Variance

Consider when $\sigma_{\epsilon,k}^2 = 1$ is a constant. Then, the expected KL divergence is a function of the posterior variance only. Since the posterior variance progresses deterministically given d_k (see Equation (37)), the problem is no longer stochastic, and we expect the open-loop and closed-loop analytical results to be the same.

6.1.1 Analytical Results

Open-Loop The open-loop expected utility function can be expressed as

$$\begin{aligned} U(d_0, d_1) &= \mathbb{E}_{y_0,y_1|d_0,d_1}\left[D_{\mathrm{KL}}\left(f(\cdot|y_0,y_1,d_0,d_1)||f(\cdot)\right) + g(f(\cdot),d_0,d_1,y_0,y_1)\right] \\ &= \mathbb{E}_{y_0,y_1|d_0,d_1}\left[\mathbb{E}_{\theta|y_0.y_1,d_0,d_1}\left[\ln\left[\frac{f(\theta|y_0,y_1,d_0,d_1)}{f(\theta)}\right]\right] - 0.01(d_0^2+d_1^2)\right] \\ &= \mathbb{E}_{\theta|d_0,d_1}\left[\mathbb{E}_{y_0,y_1|\theta,d_0,d_1}\left[\ln\left[\frac{f(\theta|y_0,y_1,d_0,d_1)}{f(\theta)}\right]\right]\right] - 0.01(d_0^2+d_1^2), \end{aligned} \qquad (38)$$

where the last equality is due to the application of conditional expectation. In the derivation, we take advantage of the linearity of expectation, and use the following formulas

$$\mathbb{E}_{\theta|d_0,d_1}\left[\theta\right] = s_0, \qquad \mathbb{E}_{\theta|d_0,d_1}\left[\theta^2\right] = \sigma_0^2 + s_0^2 \qquad (39)$$

$$\mathbb{E}_{y_i|\theta,d_i}\left[y_i\right] = \theta d_i, \qquad \mathbb{E}_{y_i|\theta,d_i}\left[y_i^2\right] = \sigma_{\epsilon,k}^2 + \theta^2 d_i^2, \qquad (40)$$

where $i = 0, 1$ indicates the experiment number, and y_i are assumed independent conditioned on θ and d_i. The final analytical expected utility surface is plotted in Figure 6(a). The optimal designs (red dashed line in Figure 6(a)) follow the formulas

$$\boxed{1 \leq d_0^* \leq \frac{\sqrt{440}}{3}, \qquad d_1^* = \frac{\sqrt{449 - 9(d_0^*)^2}}{3}.} \qquad (41)$$

The corresponding optimal expected utility value of $U^* \approx 2.5557$.

Greedy For the greedy approach, we proceed in the same manner as open-loop, taking expectations and using the relationships in Equations (39) and (40). However, we do this for one experiment at a time.

The greedy policy is

$$\boxed{d_0^* = \frac{\sqrt{440}}{3}, \qquad d_1^* = 1.} \qquad (42)$$

This design is the single point along the bottom of the domain on the open-loop optimal design front (red dashed line) in Figure 6(a), and we have the same optimal expected utility of $U^* \approx 2.5557$.

Dynamic Programming In the DP formulation, since we are interested in the expected information gain of the experiments with respect to θ, the terminal cost is chosen to be the KL divergence between the final and initial densities on θ

$$J_N(x_N) = \int_{\mathcal{H}} h(x_N) \ln\left(\frac{h(x_N)}{h(x_0)}\right) d\theta = \frac{1}{2}\left[\frac{\sigma_N^2}{\sigma_0^2} + \frac{(s_N - s_0)^2}{\sigma_0^2} - \ln\left(\frac{\sigma_N^2}{\sigma_0^2}\right) - 1\right], \qquad (43)$$

where the second equality is the special case for 1D Gaussians, and

$$h(x_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left[-\frac{(\theta - s_k)^2}{2\sigma_k^2}\right] \tag{44}$$

transforms the state x_k to its corresponding conditional density function. The Bellman equation is

$$J_k(x_k) = \max_{d_k} \mathbb{E}\left[g_k(x_k, d_k, y_k) + J_{k+1}(\mathcal{F}(x_k, d_k, y_k))\right], \qquad k = 0, \ldots, N-1. \tag{45}$$

Note that this is not a linear-quadratic DP model because the system equation (Bayes' Theorem) is not linear, and the log term in the terminal cost makes it non-quadratic.

We proceed backwards by first substituting J_2 into J_1, and again exploit the linearity of expectation and use the following formulas

$$\mathbb{E}_{y_1|s_1,\sigma_1^2,d_1}\left[y_1\right] = d_1 s_1, \qquad \mathbb{E}_{y_1|s_1,\sigma_1^2,d_1}\left[y_1^2\right] = \sigma_{\epsilon,k}^2 + d_1^2(\sigma_1^2 + s_1^2). \tag{46}$$

The optimal policy is

$$\boxed{1 \le d_0^* \le \frac{\sqrt{440}}{3}, \qquad d_1 = \mu_1^*(x_1) = \frac{\sqrt{50\sigma_1^2 - 1}}{\sigma_1}.} \tag{47}$$

Since μ_1^* only depends on σ_1^2, which is a deterministically evolution from x_0 given d_0, we can plot d_1^* as a function of d_0, shown in Figure 7:

$$d_1 = \frac{\sqrt{449 - 9(d_0^*)^2}}{3}. \tag{48}$$

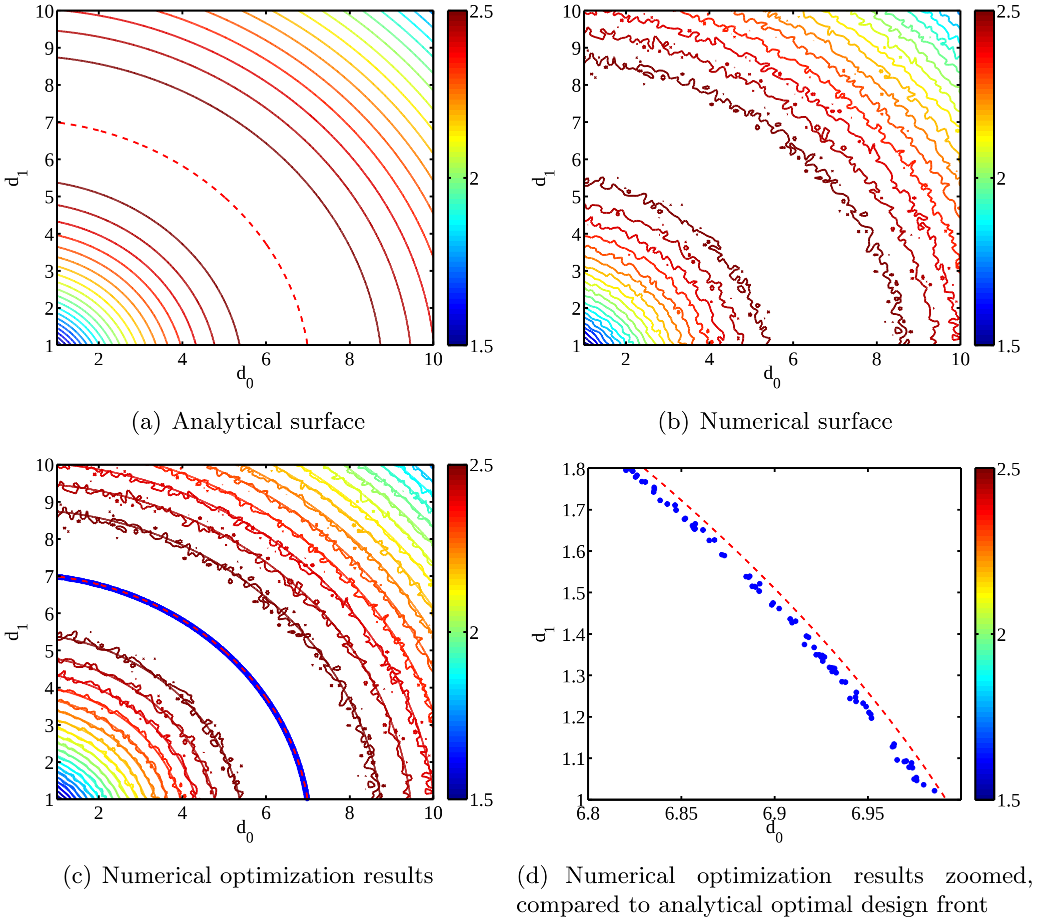
This optimal policy is identical to the open-loop optimal design. The optimal expected total reward is again $J_{0,\pi^*}(x_0 = (7, 9)) \approx 2.5557$.

6.1.2 Numerical Results

The optimal experimental design problem is solved using open-loop, greedy, and DP approaches with the numerical tools described in Section 5. Specifically, RM is used for stochastic optimization in open-loop and greedy designs. One-step lookahead backward induction is the chosen ADP technique, where the approximate value functions are degree 4 polynomials in the state variable constructed using a simple least-squares regression. We will refer to all results based on the DP *formulation* as "DP" for simplicity, even if ADP were used.

The open-loop expected utility surfaces are shown in Figure 6. The contours from analytical and numerical methods have excellent agreement. In Figure 6(c), there appears to be a "thick blue line" near the analytical optimal design front. This thick line is actually the converged results of 5000 stochastic optimization runs. Figure 6(d) provides a zoomed-in view, showing the optimization results lie closely to the true optimal design front, but slightly biased due to the finite inner-loop sample size M. As M increases, this bias is decreased.

The open-loop, greedy, and DP numerical optimization designs are superimposed with the open-loop analytical expected utility surface in Figure 8. It is possible to express the closed-loop optimal policy in terms of d_0 because this is a deterministic problem, and the optimal policy is independent of the noise y_0. As expected, rewards from simulated trajectories shown in Figure 9 are almost the same for all three design approaches, and they have excellent agreement with the analytical optimal value. The greedy method pinpoints its d_0 with ease, but has difficulty in finding the $d_1 = 1$. This is because the second experiment's objective function is very flat in a large region near the optimum, such that the bias and Monte Carlo noise can greatly shift the numerical optimum away from the true optimum. The DP results only capture one particular region of the full optimal front. This is due to the polynomial approximation used in approximating the value function, where even a slight approximation error would lead to the favoring of a localized region due to the flatness of the expected utility across the optimal front.



(a) Analytical surface

(b) Numerical surface

(c) Numerical optimization results

(d) Numerical optimization results zoomed, compared to analytical optimal design front

Figure 6: Open-loop expected utility surfaces for the constant variance case. The red dashed lines are the open-loop optimal design fronts.

6.2 State-Dependent Noise Variance

We now consider the noise variance $\sigma^2_{\epsilon,k} = \exp(0.5 s_k)$ that depends on the state. Since the mean is a function of the measurement, we expect open- and closed-loop approaches to yield different results.

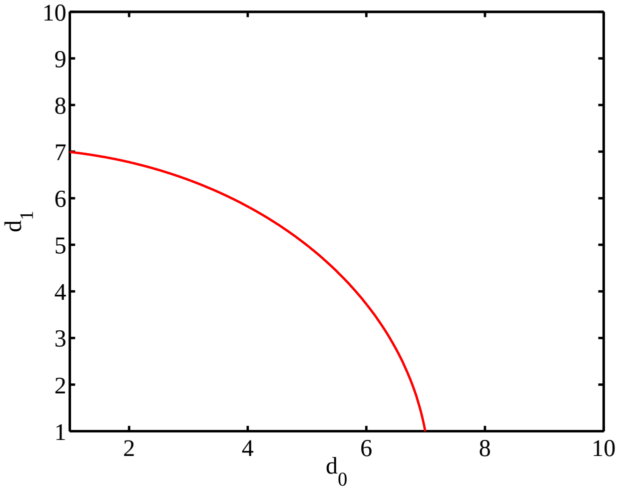
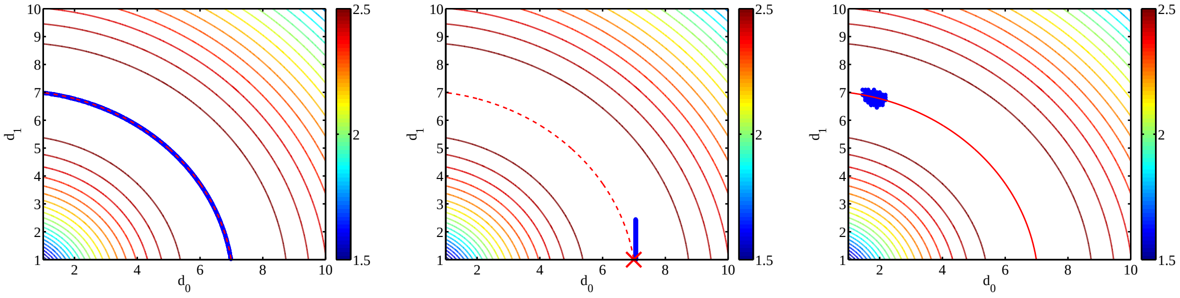


Figure 7: Analytical DP policy for the constant noise variance case.



(a) Open-loop

(b) Greedy

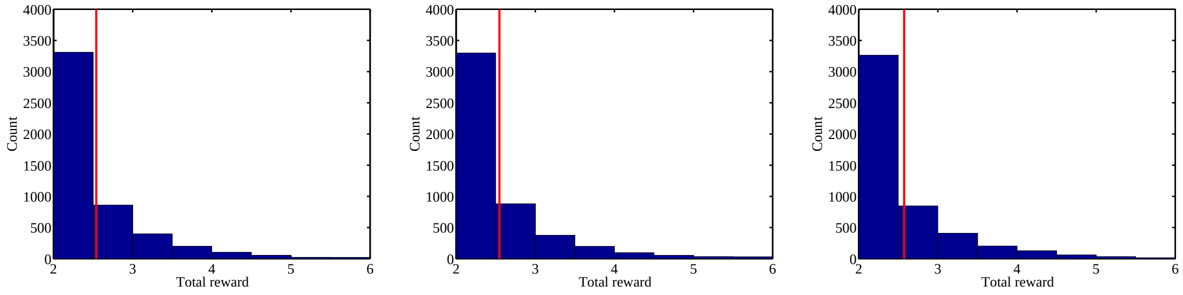
(c) DP

Figure 8: Final designs of 5000 optimization runs for the constant variance case, plotted on top of the open-loop analytical expected utility surface. The red dashed lines in Figures 8(a) and 8(b) are the open-loop optimal design fronts, the red × in Figure 8(b) is the optimal greedy policy, and the red solid line in Figure 8(c) is the DP optimal policy.

6.2.1 Analytical Results

We will not show the analytical equations in for this case as they become very messy. Instead, the policies will be plotted. While available, we do not present the analytical expected utility values for open-loop and greedy designs to avoid confusion, because the objectives used in those approaches are now different than that in DP for a noise model that depends on the state and cannot be compared. One may view their objectives as quantities that guide towards the true closed-loop optimal. Instead, we are interested in how they perform under the expected rewards function (Equation (9)) used for closed-loop design.

Open-Loop The analytical and numerical expected utility surfaces are the same as in Figure 6 except that the contour values are much lower due to the larger noise variance and the contour lines are slightly shifted. This is expected since the only difference to the constant variance case is a different variance value.



(a) Open-loop (mean: 2.54) (b) Greedy (mean: 2.55) (c) DP (mean: 2.57)

Figure 9: Final reward (utility) values of 5000 optimization runs for the constant variance case. The red vertical line indicates the mean value.

Greedy The greedy policy is as follows. For the first experiment, $d_0 \approx 6.806$; and for the second experiment, d_1 as a function of the first experiment's measurement y_0 is plotted as the blue dashed line in Figure 10.

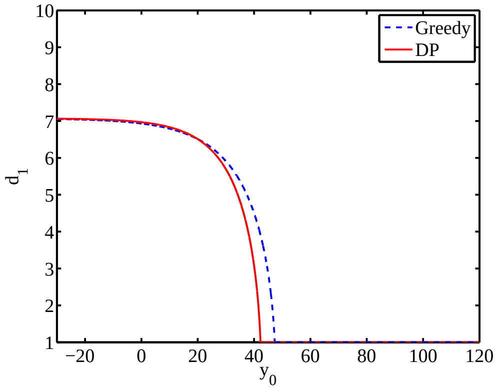


Figure 10: Analytical optimal designs for the second experiment using greedy and DP for the state-dependent variance case.

Dynamic Programming While the analytical form of $J_1(x_1)$ can be derived, it is not practical for $J_0(x_0)$ because $J_1(x_1)$ is a piecewise function. Instead, we will take a "pseudo-analytical" approach and evaluate the following expectation and maximization numerically:

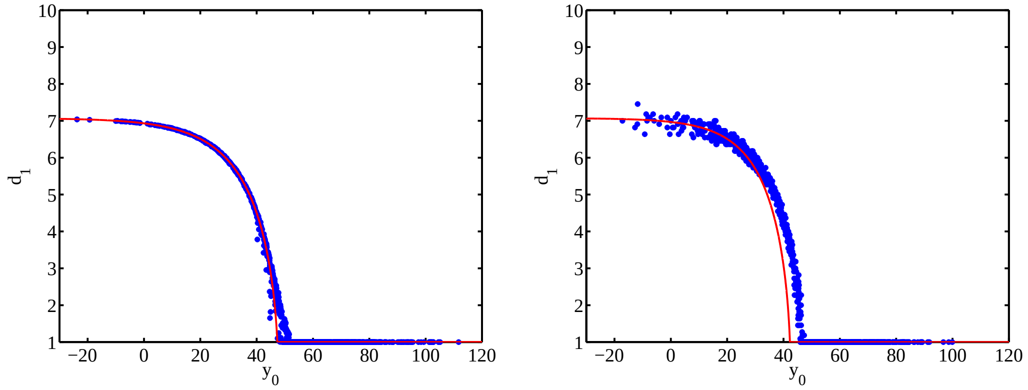
$$J_0(x_0) = \max_{d_0} \mathbb{E}\left[g_0(x_0, d_0, y_0) + J_1(\mathcal{F}(x_0, d_0, y_0))\right]. \tag{49}$$

Specifically, we perform the maximization using a grid search, and take the expectation with a Monte Carlo sample size of 10^8 to ensure that the sampling noise does not affect the maximization (the standard error of our results is on the order of 10^{-5}).

The optimal policy is as follows. For the first experiment, $d_0^* \approx 5.297$; for the second experiment, d_1^* plotted as a function of y_0 is shown as the red solid line in Figure 10, with an optimal expected reward of ≈ 1.010.

6.2.2 Numerical Results

The numerical results for the design of the second experiment for greedy and DP are superimposed with their analytical counterparts in Figure 11. Unlike the previous constant variance example, this is no longer a deterministic problem, and we expect an advantage in utilizing the results from the first experiment in designing the second experiment. Consequently, the optimal designs cannot be shown on a d_0-d_1 plane, but must be expressed as a function of the state x_1 (i.e., a policy). In this example, because we have fixed the prior x_0 and only have a single optimal value for d_0, we are able to plot d_1 as a function of the scalar y_0. In the figure, the DP optimization results doe not match the analytical designs exactly, due to the approximation error in the value function.

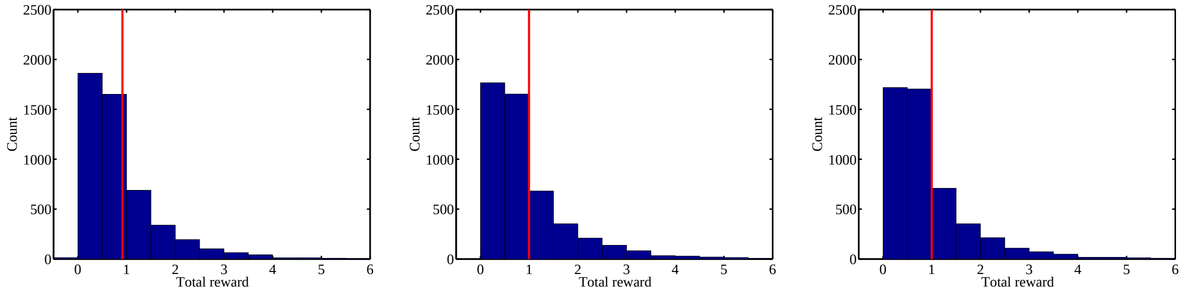


(a) Greedy Numerical

(b) DP Numerical

Figure 11: Final designs of 5000 optimization runs for the state-dependent variance case. The red line in Figure 11(a) represents the optimal greedy designs, and the red line in Figure 11(b) represents the optimal DP designs.

Another effect is that the closed-loop designs (greedy and DP) are expected to be different (and better, at least for DP) than the open-loop design. Indeed, rewards from simulated trajectories are shown in Figure 12, and the open-loop mean is about 10% lower than the greedy and DP designs (both of which are within 1% of the analytical optimal). The greedy is able to keep up with DP for this case, but this is not in general true for all noise structures that only dependent on the state.



(a) Open-loop (mean: 0.914)

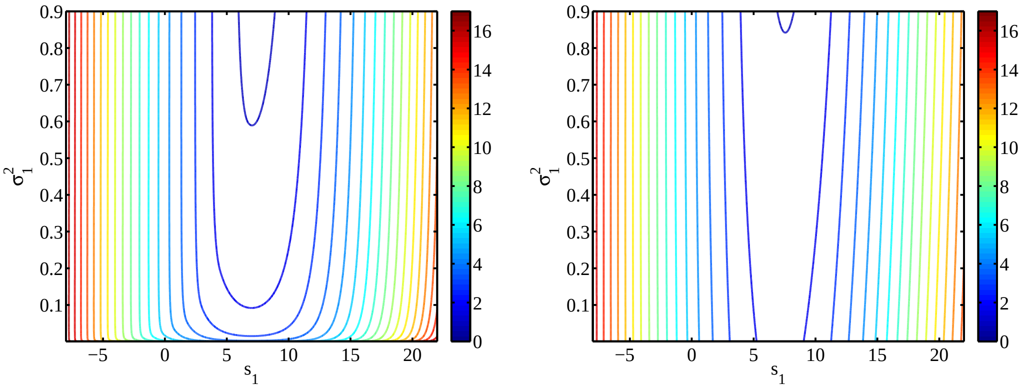
(b) Greedy (mean: 0.997)

(c) DP (mean: 0.999)

Figure 12: Final reward (utility) values of 5000 optimization runs for the state-dependent variance case. The red vertical line indicates the mean value.

In arriving at the numerical DP results, a one-step lookahead DP method is used, where only

$J_1(x_1)$ is approximated since $J_2(x_2)$ (i.e., Equation (43)) can be evaluated exactly and cheaply, and $J_0(x_0)$ is not needed since x_0 is fixed. The analytical form and polynomial approximation are shown in Figure 13. The approximation function does a good job in most of the region of interest[2] except for the parts of low σ_1^2. However, the states x_1 actually encountered in the simulated trajectories (Figure 14) concentrate only near the higher values of σ_1^2. This result demonstrates the importance of state measure: if this measure is known (or perhaps explored iteratively and adaptively), the approximation function can be made more accurately and efficiently according to this measure.



(a) $J_1(x_1)$ analytical

(b) $J_1(x_1)$ numerical

Figure 13: Analytical form and polynomial approximation to the DP value function $J_1(x_1)$ for the state-dependent variance case.

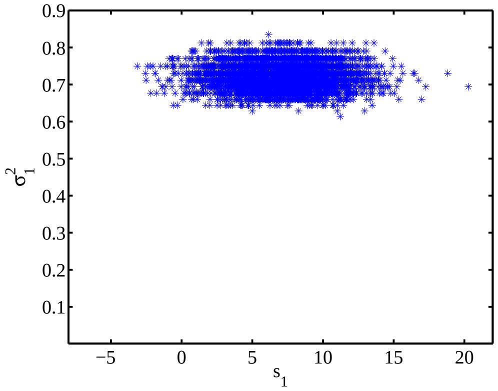


Figure 14: State x_1 encountered in the simulated trajectories in the numerical DP method for the state-dependent variance case.

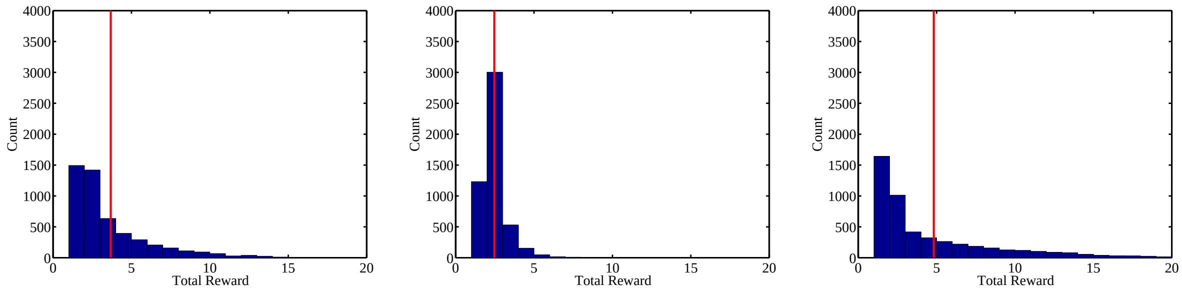
6.3 State- and Design-Dependent Noise Variance

Finally, we consider a more complicated noise variance of $\sigma_{\epsilon,k}^2 = (\frac{\sqrt{7}}{5} d_k^{s_k - 6.5})^2$. This structure is complicated enough to prohibit any analytical derivation; we will only provide numerical results.

[2] The region of interest is determined from experimentation and analytical derivation (e.g., for the range of σ_1^2).

Rewards from simulated trajectories are shown in Figure 15. The open-loop outperforms greedy by 32%, while DP further outperforms open-loop by another 34%. Fully solving the closed-loop optimal design problem can indeed be greatly beneficial over other design approaches.

Some intuition may be obtained for the DP results from the form of the variance. The variance monotonically increases with d_1 if s_1 is greater than 6.5, and decreases otherwise. Since how the mean changes does not depend on the magnitude of d_0, the DP formulation correctly applies the minimal d_0 in the first experiment (since there is a quadratic stage cost on d_k), and then chooses either the smallest or largest d_1 depending on the outcome of the first experiment. The open-loop design is unable to capture any effect that involves a change dependent on any experimental outcome as there is no feedback. The greedy design is also unable to capture this phenomenon because it does not know if we have any future experiments, and would try to recover the maximum information gain immediately, rather than "making a sacrifice" in the first experiment in order to extract an even higher gain from the second experiment.



(a) Open-loop (mean: 3.68)

(b) Greedy (mean: 2.78)

(c) DP (mean: 4.94)

Figure 15: Final reward (utility) values of 5000 optimization runs for the state- and design-dependent variance case. The red vertical line indicates the mean value.

7 Future Work

The primary objectives of this thesis are outlined in Section 2, and restated here:

1. to rigorously formulate the optimal closed-loop experimental design problem, for the purpose of Bayesian parameter inference, using a dynamic programming approach; and
2. to develop the numerical tools that find the optimal experimental conditions in a computationally feasible manner.

Here we describe in more detail the future work to be done in achieving these objectives. A proposed schedule can be found in Section 8.2.

7.1 First Objective: Closed-Loop Design Formulation

The majority of the first objective is completed, as we have rigorously formulated the optimal experimental design problem using open-loop, greedy, and DP approaches in Section 4. There remains, however, interesting questions related to the formulation.

In many practical situations, the total number of experiments is not fixed. Instead, experiments may be added or reduced due to funding adjustments, change of managerial direction or political influence, or simply that we have gained enough information from our experiments. The DP formulation in Section 4.2.2 assumes a fixed finite horizon of experiments, and does not offer the flexibility to accommodate these situations. We will explore some techniques that may be more used from this perspective.

In Section 7.1.1, we present the stopping rule formulation where an experimenter might want to stop the project once enough information has been gathered. Section 7.1.2 describes remedies under which an unexpected change in the number of experiments takes place. Finally, options in directly constructing a formulation robust to early termination are discussed in Section 7.1.3.

7.1.1 Intended Stopping Rule

Sometimes a project can be terminated once enough information has been extracted from the experiments. Such a termination is within the decision power of the designer or experimenter, and a stopping rule can be set.

In a finite horizon setting, we can set N to be the maximum possible number of experiments or simply some large number (although blindly choosing a large N can lead to substantial unnecessary computational costs) and use the Bellman equation

$$J_k(x_k) = \begin{cases} \max\left\{\max_{d_k} \mathbb{E}\left[g_k(x_k,d_k,y_k) + J_{k+1}\left(\mathcal{F}(x_k,d_k,y_k)\right)\right], \int_{\mathcal{H}} \ln\left[\frac{x_k}{x_0}\right] x_k \, d\theta\right\} & \text{if } x_k \neq T \\ 0 & \text{if } x_k = T \end{cases} \tag{50}$$

$$J_N(x_N) = \begin{cases} \int_{\mathcal{H}} \ln\left[\frac{x_N}{x_0}\right] x_N \, d\theta & \text{if } x_N \neq T \\ 0 & \text{if } x_N = T \end{cases}, \tag{51}$$

where T is the absorbing terminal state, and the 2nd argument in the max function in Equation (50) corresponds to the decision of terminating the project.

Suppose there is no upper limit for N, such as in the weather balloon example described in the Introduction section, the problem can also be formed in a infinite horizon framework:

$$J^*(x) = \max\left\{\max_{d} \mathbb{E}\left[g(x,d,y) + J^*\left(\mathcal{F}(x,d,y)\right)\right], \int_{\mathcal{H}} \ln\left[\frac{x}{x_0}\right] x \, d\theta\right\}. \tag{52}$$

Stationarity assumptions need to be imposed in order to guarantee convergence [81].

7.1.2 Unexpected Changes to the Number of Experiments

Here we describe situations where the number of experiments is changed due to factors beyond the control of the planner or experimenter (e.g., funding adjustments).

Supposed you have completed 4 experiments of a 10-experiment series originally planned using DP. Unforeseen increase in funding has allowed you to add 3 new experiments. What should you do? First, there is nothing you can do about the 4 experiments already performed. They are in the past. What matters is where you are now, your current state x_4. Next, in order to *fully* capture the future gains, a new policy must be created to take into account this change. If time is not an issue, we can simply repeat the DP computations for a 9-experiment horizon starting at the old x_4. However, this would be an expensive task, and the gains of creating the new policy may not be substantial enough to warrant repeating the full DP computations. This may especially be the case if the experiments are of the same structure inferring the same parameters [34], or if only a small change is made compared to the original total (e.g., 1 experiment added to the original total of 100). These factors, combined with often limited time in forming the new policy, present a need for *updating* or *modifying* the original policy on the fly (perhaps suboptimally), instead of starting anew.

One possible solution is to use rollout. First, we can construct a starting base policy by simply appending a heuristics to the remaining original policy

$$\pi_{\text{init}} = \left\{\mu^*_{4,\text{old}}, \dots, \mu^*_{9,\text{old}}, \mu_{10}, \dots, \mu_{12}\right\}, \tag{53}$$

where $\mu_{10}, \dots, \mu_{12}$ can be, for example, the same as $\mu^*_{9,\text{old}}$. Rollout algorithm (Equation (30)) can then be used repeatedly to improve from this base policy. Intuitively, using part of the previously-optimal policy should lead to a starting policy that is closer to the new optimal, and thus faster convergence. Numerically, we still need to overcome the challenges of parameterizing policy and value functions.

In extreme situations where the number of experiments fluctuates frequently, even an online update method may be too expensive. It would then be more advantageous to address this from the fundamentals, and use a more robust formulation to start with.

7.1.3 Formulations Robust to Horizon Change

Stopping Problem The first solution we introduce is a stopping problem formulation. Here is an example in a finite horizon setting. We introduce a new, augmented state variable $x_k = [x_{k,t}, x_{k,n}]$ where $x_{k,t} \in \{0, T_1, T_2\}$ is a "tag" component for keeping track of termination, and $x_{k,n} = f(\theta|I_k)$ is the usual conditional density function. The values for $x_{k,t}$ will be explained as we proceed through the example. We start from an initial project of $N = 8$ experiments, with the understanding that a project review will take place after the completion of 4 and 6 experiments, and each review will have a probability of p_f of terminating the project. The initial state's tag is $x_{0,t} = 0$ to represent that it is not in the termination state. For the first three experiments ($k = 0, \dots, 2$), the value functions are

$$J_k(x_k) = \max_{d_k} \mathbb{E}\left[g_k(x_k, d_k, y_k) + J_{k+1}\left(\mathcal{F}_k(x_k, d_k, y_k)\right)\right], \tag{54}$$

and the system equation is

$$x_{k+1} = \mathcal{F}_k(x_k, d_k, y_k) = \left[0, \frac{f(y_k|\theta, d_k) x_{k,n}}{f(y_k|d_k, I_k)}\right]. \tag{55}$$

For the fourth experiment, while $J_3(x_3)$ still takes the form of Equation (54), the system equation is now a stochastic function

$$x_4 = \mathcal{F}_4(x_3, d_3, y_3) = \left\{ \begin{array}{cc} \left[0, \frac{f(y_3|\theta,d_3)x_{3,n}}{f(y_3|d_3,I_3)}\right] & \text{w.p. } (1-p_f) \\ \left[T_1, \frac{f(y_3|\theta,d_3)x_{3,n}}{f(y_3|d_3,I_3)}\right] & \text{w.p. } p_f \end{array} \right. , \tag{56}$$

since there is a 25% chance of terminating the project. Here T_1 is the tag for a special (temporary) termination state (before accounting the information gain). The next value function is

$$J_4(x_4) = \left\{ \begin{array}{cc} \max_{d_4} \mathbb{E}\left[g_4(x_4, d_4, y_4) + J_5\left(\mathcal{F}(x_4, d_4, y_4)\right)\right] & \text{if } x_{4,t} \neq T_1 \\ \int_{\mathcal{H}} \ln\left[\frac{x_{4,n}}{x_{0,n}}\right] x_{4,n}\, d\theta & \text{if } x_{4,t} = T_1 \end{array} \right. , \tag{57}$$

where if $x_{4,n} = T_1$ then we will now account for the information gain, and then immediately move to the T_2 tag at the next stage (i.e., termination state after accounting for information gain) via

$$x_5 = \mathcal{F}_5(x_4, d_4, y_4) = \begin{cases} \left[0, \frac{f(y_4|\theta,d_4)x_{4,n}}{f(y_4|d_4,I_4)}\right] & \text{if } x_{4,t} \neq T_1 \\ [T_2, 0] & \text{if } x_{4,t} = T_1 \end{cases} . \tag{58}$$

Note that T_2 is an absorbing state, such that if $x_{k,t} = T_2$ then $x_{l,t} = T_2$ for all $l > k$. For the next stage, we have

$$J_5(x_5) = \begin{cases} \max_{d_5} \mathbb{E}\left[g_5(x_5, d_5, y_5) + J_6\left(\mathcal{F}(x_5, d_5, y_5)\right)\right] & \text{if } x_{5,t} \neq T_2 \\ 0 & \text{if } x_{5,t} = T_2 \end{cases} , \tag{59}$$

and $\mathcal{F}_6(x_5, d_5, y_5)$ follows Equation (56) (unless $x_5 = T_2$, in which case $x_6 = T_2$). Similarly, we have $J_6(x_6)$ having the same form as Equation (57), $\mathcal{F}_7(x_6, d_6, y_6)$ as Equation (58), $J_7(x_7)$ as Equation (59), $\mathcal{F}_8(x_7, d_7, y_7)$ as Equation (56) but with $p_f = 1$ (mandatory termination), and the terminal cost being

$$J_N(x_N) = \begin{cases} \int_{\mathcal{H}} \ln\left[\frac{x_{N,n}}{x_{0,n}}\right] x_{N,n}\, d\theta & \text{if } x_{N,t} = T_1 \\ 0 & \text{if } x_{N,t} = T_2 \end{cases} . \tag{60}$$

If there is no mandatory termination and the review occurs perpetually, an analogous infinite horizon formulation can be used.

A slightly more complicated situation would be when a project review reveals whether experiments are added or reduced effective at a future date. This variable will be random before the review date, but observed once that date passes, and must be augmented as part of the noise variable y_k. The system equations above then must take its observed value into account.

Greedification The greedy design formulation is naturally independent of the design horizon due to its myopia, and is thus robust against any changes to the total number of experiments. The downside is that greedy is a suboptimal policy. A mix between greedy and DP formulations can be used to attain some of both properties: robustness and optimality. For example, the N experiments may be broken into groups, where within each group a DP policy is sought; and between the larger groups, a greedy policy governs. Such a formulation is also numerically advantageous since now smaller DP problems need to be solved.

Incremental Information Measure Looking at the DP reward functions Equations (11) and (12), all of the information gain is in the terminal rewards, which means the policy expects the information reward to be only realized at the end of the project. If we want to create a policy that guards against early termination, then we can simply build that into the objective function by dividing the information gain into a incremental manner across the experiments.

Specifically, we assign the incremental information measures in stage rewards $g_k(x_k, d_k, y_k)$. In a two-experiment example, the incremental formulation would be

$$\max_{d_0}\left\{\mathbb{E}\left[\int_{\mathcal{H}} \ln\left[\frac{x_1}{x_0}\right] x_1\, d\theta + \max_{d_1}\left\{\mathbb{E}\left[\int_{\mathcal{H}} \ln\left[\frac{x_2}{x_1}\right] x_2\, d\theta\right]\right\}\right]\right\}, \tag{61}$$

subject to the system equation $x_{k+1} = \mathcal{F}(x_k, d_k, y_k)$. The resulting policy would not be the same as the one from Section 4.2.2 since the incremental KL terms are not additive. The policy we obtain would be one that is more robust against early termination, with the tradeoff of having an objective that is not truly optimal in the expected information gain sense, and that a KL term needs to be evaluated at every experiment which can be computationally expensive.

7.2 Second Objective: ADP Solution Methods

The second objective concerns the numerical tools for solving the optimal experimental design problem. It will be tackled in three parts: ADP, state representation, and application problem.

7.2.1 New ADP Algorithm

Foremost and central to the theme of the thesis, is the development of an ADP algorithm that solves the optimal sequential experimental design problem in a manner that is *accurate* and *computationally feasible*. To this aim, we shall start by making incremental improvements upon existing ADP methods such as those described in Section 5.2.2.

Q-Factors Q-factor is a special case of post-decision state method discussed in Section 5.2.2, where the post-decision state is taken to be the augmented state-control pairs

$$x_k^p = \mathcal{F}_1(x_k, d_k) \equiv (x_k, d_k)\,. \tag{62}$$

The advantage of this transformation is that it does not rely on any special structure of the system equation $\mathcal{F}(x_k, d_k, y_k)$, hence it is possible to execute in all DP problems. The Q-factors themselves are defined to be

$$Q_k(x_k, d_k) \equiv \mathbb{E}\left[g_k(x_k, d_k, y_k) + J_{k+1}\left(\mathcal{F}(x_k, d_k, y_k)\right)\right]. \tag{63}$$

The Bellman equation (Equations (11) and (12)) can now be equivalently restated as

$$\begin{aligned} Q_{N-1}(x_{N-1}, d_{N-1}) &= \mathbb{E}\left[g_{N-1}(x_{N-1}, d_{N-1}, y_{N-1}) + J_N(\mathcal{F}(x_{N-1}, d_{N-1}, y_{N-1}))\right] && (64)\\ Q_k(x_k, d_k) &= \mathbb{E}\left[g_k(x_k, d_k, y_k) + \max_{d_{k+1}} Q_{k+1}\left(\mathcal{F}(x_k, d_k, y_k), d_{k+1}\right)\right] && (65) \end{aligned}$$

and

$$J_0(x_0) = \max_{d_0} Q_0(x_0, d_0). \tag{66}$$

Q-factor is a transformation technique that takes advantage of the recursive nature of the DP problem; it does not change or approximate the problem in anyway.

How can Q-factors be advantageous? Instead of approximating value functions $J_k(x_k)$, we now approximate Q-factors $Q_k(x_k, d_k)$. From a numerical perspective, we no longer need to evaluate the optimization of an expectation, but instead only need to take the expectation of an optimization. First, stochastic optimization methods can now be avoided, and instead more developed deterministic optimization techniques may be used. Second, assuming the Q-factor approximations are constructed via $\Pi\mathbb{E}$, where Π is a projection onto an approximating subspace. Then, $\hat{\Pi}\hat{\mathbb{E}}$, where Monte Carlo approximations are used for the projection and expectation, is an unbiased estimator due to the linearity of Monte Carlo. In contrast, $\hat{\Pi}\max\hat{\mathbb{E}}$ used in value function approximations are biased due to the nonlinearity of max. Third, $\hat{\Pi}\hat{\mathbb{E}}$ can now use more efficient methods that heavily rely on function regularity, such as sparse quadrature. Quadrature methods would not work well for noisy function, which would be the case for $\hat{\Pi}\max\hat{\mathbb{E}}$ in constructing $\hat{J}$ functions due to stochastic optimization.

Forward Trajectory Simulation The ADP algorithm presented by Powell [83] (see Section 5.2.2, Algorithm 1)) produces sample paths from forward trajectory simulations, and uses these sample paths to adaptively update the value function approximations. Its main advantages over methods based on backward-induction in constructing the approximation functions are: (1) exploitation of the approximation function and automatically explore the state space; (2) construction of approximation functions not directly dependent on approximations from previous stage, and thus can avoid the potential exponential growth in approximation error from backward induction; and (3) flexibility in stopping or continue refining the approximation functions.

As we have learned from Section 6.2.2 (particularly Figure 14), knowledge about the state measure can be extremely advantageous in building the approximation functions. We envision the new ADP method to be developed being based on forward trajectory simulation, as one aspect of its novelty lies within a more efficient construction of the approximate value functions through the use of a state measure approximation.

As mentioned in Section 5.2.2, exploitation and exploration need to be balanced. Here is an illustration of the unwanted consequences if only exploitation were used. We start with some initialization of the value function approximations, which may not be close to the true value functions. In these initial forms, some regions of the state will have higher values while others have low values. Due to the maximization in Bellman equation, the simulated trajectories will tend to go to the next states that have high values of the approximation function. The approximations are then mostly updated at states in which we have obtained data, and the unvisited "low-value" states will remain low value. As a result, those states will remain unvisited, and we would not be able to find out about the regions we wrongly assumed to be low value in the initialization. As a result, exploration needs to be added to provide the opportunity for us to find these mistakes.

One approach to attain exploration is by treating the approximation values of the physical states in an uncertain (e.g., probabilistic) setting. Exploration then depends not only on its mean, but also on how confident we feel about that value. We might opt to visit a state that is highly uncertain over one with a high value. Various statistical techniques may be employed, such as Gaussian process [66]. The policy governing exploration, however, is likely to be a result of heuristics and experimentation.

Inference We have so far focused on constructing accurate value function approximations and finding the optimal policy. However, the ultimate goal of the experiments is parameter inference. Indeed, how good of a job we have done is measure by how well we do in inferring the uncertain parameters, and this is reflected by our choice of the objective function. While it is true that accurate value function approximations and optimization would yield a good policy, but a good policy does not require them. In fact, we may spend substantial resources in improving our approximation techniques while only attaining insignificant improvement in the objective. Would it then be possible to find another indicator to guide us in finding a "good enough" policy? The answer potentially lies within iterative methods, which is another reason to pursue the forward trajectory simulation method. It remains an open question and is a future direction of research.

7.2.2 State Representation

The major numerical difficulty in the closed-loop formulations (e.g., greedy and DP) is to represent the state variable x_k, which are probability density functions. We are interested in problems beyond linear-Gaussian models, and a general density function cannot be represented exactly by a finite-dimensional vector. It then must be approximately represented.

One popular technique is Gaussian mixture model (GMM) (e.g., [132])

$$f(\theta|I_k) = \sum_{m=1}^{M} w_m \mathcal{N}\left(\theta|\mu_m, \Sigma_m, I_k\right), \tag{67}$$

where μ_m is the mean vector of the mth Gaussian, Σ_m is the covariance matrix, and w_m are the relative weights. The fitting of these parameters can be done efficiently using the Expectation-Maximization (EM) algorithm [138]. However, the choice M is not always clear, and often needs to be adjusted in an automated manner via data clustering algorithms [139].

Another technique is to use a nonparametric (particle) representation. In particular, particle filters (a.k.a. sequential Monte Carlo) [133] provides a natural infrastructure for representing and updating our state variable. This method, however, would not be suitable for high dimensions as the number of particles required for obtaining reasonably accurate representations would be enormous.

Finally, instead of trying to capture the distribution directly, one can focus on capturing the *mapping* or transformation from a reference random variable to the random variable corresponding to the current state [134]. Although the density function is no longer available, but all that is required in our experimental design formulation is to be able to sample the random variable, which is available from this method. The basis functions in value function approximations can no longer depend on properties of the density function such as mean or higher moments. However, we can simply create new features that depend on the parameters of the mapping instead. How well these features work in ADP is yet to be explored. This newly introduced method provides an alternative to MCMC for solving Bayesian inference problems while requiring substantially less computation effort.

Due to the numerous research topics in this thesis project, we will focus mainly on the development of the ADP algorithm. For the state representation problem, we will for the most part select an existing method that works well in the sequential Bayesian inference setting for our closed-loop experimental design formulation.

7.3 Model Problems and Applications

We would like to apply the framework and tools developed on a series of model and application problems. Through these, we will address practical questions such as how much more beneficial is DP over other design methods such as open-loop and greedy, and if the advantage increases or decreases as the number of experiments is expanded. The answer is not trivial: on the one hand, only DP is able to fully capture the expected effects of future, while on the other hand, a longer horizon would lead to an accumulation of value function approximation error, and repeated experiments can lead to diminishing return. These questions naturally bring out the need of a stopping rule, with a simple formulation described in Section 7.1.1.

To test our methods, we start with a series of simple model problems both to assess the performance of our tools and to extract intuition and insight from the results. We then move on application problems with different levels of complexity. First, optimally designing experiments with fixed form and stationary parameters are tested on a chemical combustion system. Then, in the setting of optimal sensor placement in a diffusion field for source inversion, we explore the optimal design framework applies to experiments with varying form and both stationary and non-stationary parameters. Finally, we describe an above-and-beyond full-blown scenario that possesses the closest resemblance to a real life situation.

7.3.1 Simple Model Problems

The simple model problems we have chosen are the linear-Gaussian problems described in Section 6. Specifically, we have explored variants where the Gaussian noise has constant, state-dependent, and state- and control-dependent variance structures. The first two allow the derivation of analytical results for open-loop, greedy, and DP designs, and allowed us to not only compare the performance between the different design approaches, but also to the exact optimal design and value. Moreover, these simple examples provided motivation for the superiority of DP design, and intuition and insight on why an approach chose its design or policy and which areas we need to focus on. For detailed formulation, implementation, results and discussions, please see Section 6.

7.3.2 Experiments with Fixed Form and Stationary Parameters

The chemical combustion problem is the first example described in Section 1, where the goal is to determine experimental conditions such as temperature and pressure that are optimal for inferring kinetic parameters from measurements at different times. When the parameter to be inferred, and measurement quantity and times are fixed, we have experiments of a fixed form and stationary parameters. This provides with us the simplest type of problem in which we can test our tools on a realistic application model. We investigated this problem in the open-loop framework, and used the opportunity to compare and validate the performance between alternative numerical methods, in particular stochastic optimization and polynomial chaos surrogates. Our next step is to apply sequential experimental design to this problem. Full details of this work can be found in [7].

7.3.3 Experiments with Varying Form and Stationary Parameters

For the next three subsections, let us consider the situation when a biological weapon is released as a result of a terrorist attack. As the toxic contaminant diffuses from its source, it endangers the health and safety of everyone around it. Having ordered appropriate evacuations, the authority must now find the contaminant source and eliminate it as soon as possible! It would be far too dangerous to send personnel into the contamination zone. Luckily, a number of remote-controlled robotic vehicles are available. But due to their slow movement, a full visual search for the source is out of the question. We must use a limited sequence of measurements of the contaminant concentration to infer its source location. When and where should these measurements be made so the source is determined most accurately and quickly? Our work can be used to answer these questions.

We will describe a few different levels of simplification to this complicated situation. In a very simple one, the physical process can be described by the 2D scalar diffusion equation in a rectangular domain. A sequence of measurements can then be made, and we would like to find the optimal measuring locations as well as times such that the expected information gain on the source location is maximized. The source is assumed stationary in this case. In this type of problem, the experimental form changes with time, as we would be making measurements at different times. The author has carried out investigation in the open-loop design and with fixed measurement times of this problem in [9].

7.3.4 Experiments with Varying Form and Non-Stationary Parameters

In the next level of complication, the source is assumed to be moving (perhaps being carried on a truck) according to some known model (e.g., random walk) with unknown parameters—the uncertain source location is now a function of time. The objective may need to be modified, for example, to account for the inference of the random walk model parameters, or the uncertainty in prediction of the truck location at a future time. Such a problem can be related to the field of filtering, which can provide insights and lend existing techniques. The measurement times also can be chosen, but under constraints or penalties depending on the distance from the previous measurement (i.e., due to movement speeds of robotic vehicles). A prescribed or changing convection field can also be easily incorporated. This is the highest level of complexity we aim for in this thesis.

7.3.5 Above and Beyond

A truly realistic model would be fully 3D and include building topology and road geometries. While such complexities make the physical simulation much harder, the problem is not conceptually much different. A more challenging objective would be to tie the project to path planning (i.e., for the sensor robot to also catch the source), and to use game theory to intercept an intelligent and evasive moving source.

8 Milestones and Proposed Schedule

8.1 Milestones

Completed	
2010/01	qualifying exams
2010/06	S.M. degree
2012/04	thesis committee formation
2012/12	minor approval
2012/12	course requirement
Future	
2013/02	thesis proposal defense
2014/06	Ph.D. defense and graduation

8.2 Proposed Schedule

Completed	
2008/09–2011/08	open-loop design for multiple experiments; gradient-free stochastic optimization methods (SPSA and NM); combustion application; contents in JCP paper [7]
2011/06–2012/05	gradient-based stochastic optimization methods (RM and SAA-BFGS); diffusion source-inversion application; contents in IJUQ paper [9]
2011/11–2013/01	closed-loop DP design formulation; analytic solutions and preliminary numerical ADP results for simple linear-Gaussian cases
Future	
2013/02–2013/07	*ADP*: literature review, implement existing techniques (e.g., Q-factors), create new techniques (state measure exploration and simultaneous inference)
2013/08–2013/11	*state representation*: literature review, compare and choose methods to represent the states (PDFs) in experimental design context
2013/12–2014/03	combine tools together, run cases on the application problem
2014/04–2014/06	write and defend thesis

References

[1] Frenklach, M., "Transforming data into knowledge—Process Informatics for combustion chemistry," *Proceedings of the Combustion Insitute*, Vol. 31, 2007, pp. 125–140.

[2] Davidson, D. F. and Hanson, R. K., "Interpreting Shock Tube Ignition Data," *International Journal of Chemical Kinetics*, Vol. 36, No. 9, 2004, pp. 510–523.

[3] Baulch, D. L., Cobos, C. J., Cox, R. A., Frank, P., Hayman, G., Just, T., Kerr, J. A., Murrells, T., Pilling, M. J., Troe, J., Walker, R. W., and Warnatz, J., "Evaluated Kinetic Data for Combustion Modeling. Supplement I," *Journal of Physical and Chemical Reference Data*, Vol. 23, No. 6, 1994, pp. 847–1033.

[4] Baulch, D. L., Bowman, C. T., Cobos, C. J., Cox, R. A., Just, T., Kerr, J. A., Pilling, M. J., Stocker, D., Troe, J., Tsang, W., Walker, R. W., and Warnatz, J., "Evaluated Kinetic Data for Combustion Modeling: Supplement II," *Journal of Physical and Chemical Reference Data*, Vol. 34, No. 3, 2005, pp. 757–1398.

[5] Reagan, M. T., Najm, H. N., Ghanem, R. G., and Knio, O. M., "Uncertainty Quantification in Reacting-Flow Simulations Through Non-Intrusive Spectral Projection," *Combustion and Flame*, Vol. 132, No. 3, 2003, pp. 545–555.

[6] Phenix, B. D., Dinaro, J. L., Tatang, M. A., Tester, J. W., Howard, J. B., and McRae, G. J., "Incorporation of Parametric Uncertainty into Complex Kinetic Mechanisms: Application to Hydrogen Oxidation in Supercritical Water," *Combustion and Flame*, Vol. 112, 1998, pp. 132–146.

[7] Huan, X. and Marzouk, Y. M., "Simulation-Based Optimal Bayesian Experimental Design for Nonlinear Systems," *Journal of Computational Physics*, Vol. 232, No. 1, 2013, pp. 288–317.

[8] Shea, D. A. and Lister, S. A., "The BioWatch Program: Detection of Bioterrorism," Tech. Rep. RL 32152, Congressional Research Service Report, 2003.

[9] Huan, X. and Marzouk, Y. M., "Gradient-Based Stochastic Optimization Methods in Bayesian Experimental Design," 2013, submitted.

[10] "Radiosonde Observations," NOAA National Weather Service, http://www.ua.nws.noaa.gov/factsheet.htm.

[11] Atkinson, A. C. and Donev, A. N., *Optimum Experimental Designs*, Oxford Statistical Science Series, Oxford University Press, 1992.

[12] Chaloner, K. and Verdinelli, I., "Bayesian Experimental Design: A Review," *Statistical Science*, Vol. 10, No. 3, 1995, pp. 273–304.

[13] Box, G. E. P. and Lucas, H. L., "Design of Experiments in Non-Linear Situations," *Biometrika*, Vol. 46, No. 1/2, 1959, pp. 77–90.

[14] Chu, Y. and Hahn, J., "Integrating Parameter Selection with Experimental Design Under Uncertainty for Nonlinear Dynamic Systems," *AIChE Journal*, Vol. 54, No. 9, 2008, pp. 2310–2320.

[15] Ford, I., Titterington, D. M., and Christos, K., "Recent Advances in Nonlinear Experimental Design," *Technometrics*, Vol. 31, No. 1, 1989, pp. 49–60.

[16] Clyde, M. A., *Bayesian Optimal Designs for Approximate Normality*, Ph.D. thesis, University of Minnesota, 1993.

[17] Müller, P., "Simulation Based Optimal Design," *Bayesian Statistics 6: Proceedings of the Sixth Valencia International Meeting*, Oxford University Press, 1998, pp. 459–474.

[18] Guest, T. and Curtis, A., "Iteratively Constructive Sequential Design of Experiments and Surveys with Nonlinear Parameter-Data Relationships," *Journal of Geophysical Research*, Vol. 114, No. B04307, 2009, pp. 1–14.

[19] Lindley, D. V., "On a Measure of the Information Provided by an Experiment," *The Annals of Mathematical Statistics*, Vol. 27, No. 4, 1956, pp. 986–1005.

[20] Lindley, D. V., *Bayesian Statistics, A Review*, Society for Industrial and Applied Mathematics (SIAM), 1972.

[21] Loredo, T. J., "Rotating Stars and Revolving Planets: Bayesian Exploration of the Pulsating Sky," *Bayesian Statistics 9: Proceedings of the Nineth Valencia International Meeting*, Oxford University Press, 2010, pp. 361–392.

[22] Sebastiani, P. and Wynn, H. P., "Maximum Entropy Sampling and Optimal Bayesian Experimental Design," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, Vol. 62, No. 1, 2000, pp. 145–157.

[23] Loredo, T. J. and Chernoff, D. F., "Bayesian Adaptive Exploration," *Statistical Challenges of Astronomy*, Springer, 2003, pp. 57–69.

[24] van den Berg, J., Curtis, A., and Trampert, J., “Optimal Nonlinear Bayesian Experimental Design: an Application to Amplitude Versus Offset Experiments,” *Geophysical Journal International*, Vol. 155, No. 2, November 2003, pp. 411–421.

[25] Ryan, K. J., “Estimating Expected Information Gains for Experimental Designs With Application to the Random Fatigue-Limit Model,” *Journal of Computational and Graphical Statistics*, Vol. 12, No. 3, September 2003, pp. 585–603.

[26] Terejanu, G., Upadhyay, R. R., and Miki, K., “Bayesian Experimental Design for the Active Nitridation of Graphite by Atomic Nitrogen,” *Experimental Thermal and Fluid Science*, Vol. 36, 2012, pp. 178–193.

[27] Mosbach, S., Braumann, A., Man, P. L. W., Kastner, C. A., Brownbridge, G., and Kraft, M., “Iterative Improvement of Bayesian Parameter Estimates for an Engine Model by Means of Experimental Design,” Tech. Rep. 104, c4e-Preprint Series, University of Cambridge, 2011.

[28] Russi, T., Packard, A., Feeley, R., and Frenklach, M., “Sensitivity Analysis of Uncertainty in Model Prediction,” *The Journal of Physical Chemistry A*, Vol. 112, No. 12, 2008, pp. 2579–2588.

[29] Müller, P. and Parmigiani, G., “Optimal Design via Curve Fitting of Monte Carlo Experiments,” *Journal of the American Statistical Association*, Vol. 90, No. 432, December 1995, pp. 1322–1330.

[30] Clyde, M. A., Müller, P., and Parmigiani, G., “Exploring Expected Utility Surfaces by Markov Chains,” Tech. Rep. 95-39, Duke University, 1995.

[31] Müller, P., Sansó, B., and De Iorio, M., “Optimal Bayesian Design by Inhomogeneous Markov Chain Simulation,” *Journal of the American Statistical Association*, Vol. 99, No. 467, September 2004, pp. 788–798.

[32] Hamada, M., Martz, H. F., Reese, C. S., and Wilson, A. G., “Finding Near-Optimal Bayesian Experimental Designs via Genetic Algorithms,” *The American Statistician*, Vol. 55, No. 3, August 2001, pp. 175–181.

[33] Guestrin, C., Krause, A., and Singh, A. P., “Near-Optimal Sensor Placements in Gaussian Processes,” *Proceedings of the 22nd International Conference on Machine Learning*, 2005, pp. 265–272.

[34] Krause, A., McMahan, B., Guestrin, C., and Gupta, A., “Robust Submodular Observation Selection,” *Journal of Machine Learning Research*, Vol. 9, 2008, pp. 2761–2801.

[35] Huan, X., *Accelerated Bayesian Experimental Design for Chemical Kinetic Models*, Master’s thesis, Massachusetts Institute of Technology, 2010.

[36] Cavagnaro, D. R., Myung, J. I., Pitt, M. A., and Kujala, J. V., “Adaptive Design Optimization: A Mutual Information-Based Approach to Model Discrimination in Cognitive Science,” *Neural Computation*, Vol. 22, No. 4, 2012, pp. 887–905.

[37] Solonen, A., Haario, H., and Laine, M., “Simulation-Based Optimal Design Using a Response Variance Criterion,” *Journal of Computational and Graphical Statistics*, Vol. 21, No. 1, 2012, pp. 234–252.

[38] Müller, P., Berry, D. A., Grieve, A. P., Smith, M., and Krams, M., "Simulation-based sequential Bayesian design," *Journal of Statistical Planning and Inference*, Vol. 137, No. 10, 2007, pp. 3140 – 3150.

[39] Drovandi, C. C., McGree, J., and Pettitt, A. N., "A sequential Monte Carlo algorithm to incorporate model uncertainty in Bayesian sequential design," *Journal of Computational and Graphical Statistics*, 2012.

[40] von Toussaint, U., "Bayesian inference in physics," *Rev. Mod. Phys.*, Vol. 83, 2011, pp. 943–999.

[41] Brockwell, A. E. and Kadane, J. B., "A Gridding Method for Bayesian Sequential Decision Problems," *Journal of Computational and Graphical Statistics*, Vol. 12, No. 3, 2003, pp. 566–584.

[42] Christen, J. and Nakamura, M., "Sequential stopping rules for species accumulation," *Journal of Agricultural, Biological, and Environmental Statistics*, Vol. 8, 2003, pp. 184–195.

[43] Wathen, J. K. and Christen, J. A., "Implementation of Backward Induction for Sequentially Adaptive Clinical Trials," *Journal of Computational and Graphical Statistics*, Vol. 15, No. 2, 2006, pp. 398–413.

[44] Gautier, R. and Pronzato, L., "Sequential Design and Active Control," *Lecture Notes-Monograph Series*, Vol. 34, 1998, pp. 138–151.

[45] Nelder, J. A. and Mead, R., "A Simplex Method for Function Minimization," *The Computer Journal*, Vol. 7, No. 4, 1965, pp. 308–313.

[46] Kiefer, J. and Wolfowitz, J., "Stochastic Estimation of the Maximum of a Regression Function," *The Annals of Mathematical Statistics*, Vol. 23, No. 3, 1952, pp. 462–466.

[47] Spall, J. C., "An Overview of the Simultaneous Perturbation Method for Efficient Optimization," *Johns Hopkins APL Technical Digest*, Vol. 19, No. 4, 1998, pp. 482–492.

[48] Kushner, H. and Yin, G., *Stochastic approximation and recursive algorithms and applications*, Applications of mathematics, Springer, 2003.

[49] Shapiro, A., "Asymptotic Analysis of Stochastic Programs," *Annals of Operations Research*, Vol. 30, No. 1, 1991, pp. 169–186.

[50] Robbins, H. and Monro, S., "A Stochastic Approximation Method," *The Annals of Mathematical Statistics*, Vol. 22, No. 3, 1951, pp. 400–407.

[51] Healy, K. and Schruben, L. W., "Retrospective Simulation Response Optimization," *Proceedings of the 1991 Winter Simulation Conference*, 1991, pp. 901–906.

[52] Gürkan, G., Özge, A. Y., and Robinson, S. M., "Sample-Path Optimization in Simulation," *Proceedings of the 1994 Winter Simulation Conference*, 1994, pp. 247–254.

[53] Kleywegt, A. J., Shapiro, A., and Homem-de-Mello, T., "The Sample Average Approximation Method for Stochastic Discrete Optimization," *SIAM Journal on Optimization*, Vol. 12, No. 2, 2002, pp. 479–502.

[54] Ahmed, S. and Shapiro, A., "The Sample Average Approximation Method for Stochastic Programs with Integer Recourse," *Georgia Institute of Technology Technical Report*, 2002.

[55] Verweij, B., Ahmed, S., Kleywegt, A. J., Nemhauser, G., and Shapiro, A., "The Sample Average Approximation Method Applied to Stochastic Routing Problems: A Computational Study," *Computational Optimization and Applications*, Vol. 24, No. 2, 2003, pp. 289–333.

[56] Benisch, M., Greenwald, A., Naroditskiy, V., and Tschantz, M., "A Stochastic Programming Approach to Scheduling in TAC SCM," *Proceedings of the 5th ACM Conference on Electronic Commerce*, 2004, pp. 152–159.

[57] Greenwald, A., Guillemette, B., Naroditskiy, V., and Tschantz, M., "Scaling Up the Sample Average Approximation Method for Stochastic Optimization with Applications to Trading Agents," *Agent-Mediated Electronic Commerce. Designing Trading Agents and Mechanisms*, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2006, pp. 187–199.

[58] Schütz, P., Tomasgard, A., and Ahmed, S., "Supply Chain Design Under Uncertainty Using Sample Average Approximation and Dual Decomposition," *European Journal of Operational Research*, Vol. 199, No. 2, 2009, pp. 409–419.

[59] Norkin, V., Pflug, G., and Ruszczyski, A., "A Branch and Bound Method for Stochastic Global Optimization," *Mathematical Programming*, Vol. 83, No. 1, 1998, pp. 425–450.

[60] Mak, W.-K., Morton, D. P., and Wood, R. K., "Monte Carlo Bounding Techniques for Determining Solution Quality in Stochastic Programs," *Operations Research Letters*, Vol. 24, No. 1–2, 1999, pp. 47–56.

[61] Chen, H. and Schmeiser, B. W., "Retrospective Approximation Algorithms for Stochastic Root Finding," *Proceedings of the 1994 Winter Simulation Conference*, 1994, pp. 255–261.

[62] Chen, H. and Schmeiser, B., "Stochastic Root Finding via Retrospective Approximation," *IIE Transactions*, Vol. 33, No. 3, 2001, pp. 259–275.

[63] Eldred, M. S., Giunta, A. A., and Collis, S. S., "Second-Order Corrections for Surrogate-Based Optimization with Model Heuristics," *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004, AIAA paper 2004-4457.

[64] Frangos, M., Marzouk, Y., Willcox, K., and van Bloemen Waanders, B., *Computational Methods for Large-Scale Inverse Problems and Quantification of Uncertainty*, chap. Surrogate and reduced-order modeling: a comparison of approaches for large-scale statistical inverse problems, Wiley, 2010.

[65] Kennedy, M. C. and O'Hagan, A., "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, Vol. 63, No. 3, 2001, pp. 425–464.

[66] Rasmussen, C. E., *Gaussian processes for machine learning*, MIT Press, 2006.

[67] Wiener, N., "The Homogeneous Chaos," *American Journal of Mathematics*, Vol. 60, No. 4, 1938, pp. 897–936.

[68] Ghanem, R. and Spanos, P., *Stochastic Finite Elements: A Spectral Approach*, Springer, 1991.

[69] Xiu, D. and Karniadakis, G. E., "The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations," *SIAM Journal of Scientific Computing*, Vol. 24, No. 2, 2002, pp. 619–644.

[70] Debusschere, B. J., Najm, H. N., Pébay, P. P., Knio, O. M., Ghanem, R. G., and Le Maître, O. P., "Numerical Challenges in the Use of Polynomial Chaos Representations for Stochastic Processes," *SIAM Journal on Scientific Computing*, Vol. 26, No. 2, 2004, pp. 698–719.

[71] Najm, H. N., "Uncertainty Quantification and Polynomial Chaos Techniques in Computational Fluid Dynamics," *Annual Review of Fluid Mechanics*, Vol. 41, No. 1, 2009, pp. 35–52.

[72] Xiu, D., "Fast Numerical Methods for Stochastic Computations: A Review," *Communications in Computational Physics*, Vol. 5, No. 2-4, 2009, pp. 242–272.

[73] Le Maître, O. P. and Knio, O. M., *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics*, Springer, 2010.

[74] Conrad, P. and Marzouk, Y., "Adaptive Smolyak Pseudospectral Approximation," *SIAM Journal on Scientific Computing*, 2012, submitted.

[75] Berkooz, G., Holmes, P., and Lumley, J., "The proper orthogonal decomposition in the analysis of turbulent flows," *Annual review of fluid mechanics*, Vol. 25, No. 1, 1993, pp. 539–575.

[76] Sirovich, L., "Turbulence and the dynamics of coherent structures. I-Coherent structures. II-Symmetries and transformations. III-Dynamics and scaling," *Quarterly of applied mathematics*, Vol. 45, 1987, pp. 561–571.

[77] Bui-Thanh, T., Willcox, K., and Ghattas, O., "Model Reduction for Large-Scale Systems with High-Dimensional Parametric Input Space," *SIAM Journal on Scientific Computing*, Vol. 30, 2007, pp. 3270–3288.

[78] Lazzara, D. S., *Modeling and sensitivity analysis of aircraft geometry for multidisciplinary optimization problems*, Ph.D. thesis, Massachusetts Institute of Technology, 2010.

[79] Bellman, R., "Bottleneck problems and dynamic programming," *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 39, No. 9, 1953, pp. 947.

[80] Athans, M., "The role and use of the stochastic linear-quadratic-Gaussian problem in control system design," *IEEE Transactions on Automatic Control*, Vol. 16, No. 6, 1971, pp. 529–552.

[81] Bertsekas, D. P., *Dynamic Programming and Optimal Control*, Athena Scientific, 2nd ed., 2000.

[82] Morari, M. and Lee, J. H., "Model predictive control: past, present and future," *Computers & Chemical Engineering*, Vol. 23, No. 4-5, 1999, pp. 667 – 682.

[83] Powell, W. B., *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, Wiley-Interscience, 2007.

[84] Bertsekas, D. P. and Tsitsiklis, J. N., *Neuro-dynamic programming*, Athena Scientific, 1996.

[85] Sutton, R. and Barto, A., *Reinforcement learning: An introduction*, Vol. 1, Cambridge Univ Press, 1998.

[86] Rüde, U., *Mathematical and computational techniques for multilevel adaptive methods*, SIAM, 1993.

[87] Singh, S., Jaakkola, T., and Jordan, M., "Reinforcement learning with soft state aggregation," *Advances in neural information processing systems*, 1995, pp. 361–368.

[88] Puterman, M., *Markov decision processes: Discrete stochastic dynamic programming*, John Wiley & Sons, Inc., 1994.

[89] Tsitsiklis, J. N. and Van Roy, B., "Regression Methods for Pricing Complex American-Style Options," *IEEE Transactions on Neural Networks*, Vol. 12, No. 4, July 2001, pp. 694–703.

[90] Young, P. C., *Recursive Estimation and Time-Series Analysis: An Introduction for the Student and Practitioner*, Springer-Verlag, 2011.

[91] Kalman, R. et al., "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, Vol. 82, No. 1, 1960, pp. 35–45.

[92] Sutton, R. S., "Learning to predict by the methods of temporal differences," *Machine Learning*, Vol. 3, 1988, pp. 9–44.

[93] Dayan, P., "The convergence of TD(λ) for general λ," *Machine Learning*, Vol. 8, 1992, pp. 341–362.

[94] Tesauro, G., "Practical issues in temporal difference learning," *Machine Learning*, Vol. 8, 1992, pp. 257–277.

[95] Thrun, S., "The role of exploration in learning control," *Handbook of intelligent control: neural, fuzzy and adaptive approaches. Van Nostrand Reinhold, Florence, Kentucky*, 1992.

[96] Van Roy, B., Bertsekas, D., Lee, Y., and Tsitsiklis, J., "A neuro-dynamic programming approach to retailer inventory management," *Proceedings of the 36th IEEE Conference on Decision and Control*, Vol. 4, 1997, pp. 4052–4057.

[97] Powell, W., George, A., Bouzaiene-Ayari, B., and Simao, H., "Approximate dynamic programming for high dimensional resource allocation problems," *Proceedings. 2005 IEEE International Joint Conference on Neural Networks*, Vol. 5, 2005, pp. 2989–2994.

[98] Watkins, C., *Learning from Delayed Rewards*, Ph.D. thesis, King's College, 1989.

[99] Watkins, C. and Dayan, P., "Q-learning," *Machine learning*, Vol. 8, No. 3, 1992, pp. 279–292.

[100] Berger, J. O., *Statistical Decision Theory and Bayesian Analysis*, Springer, 2nd ed., 1985.

[101] Sivia, D. S. and Skilling, J., *Data Analysis: a Bayesian Tutorial*, Oxford University Press, 2006.

[102] Ginebra, J., “On the Measure of the Information in a Statistical Experiment,” *Bayesian Analysis*, Vol. 2, No. 1, 2007, pp. 167–212.

[103] Cover, T. M. and Thomas, J. A., *Elements of Information Theory*, John Wiley & Sons, Inc., 2nd ed., 2006.

[104] MacKay, D. J. C., *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2002.

[105] Long, Q., Scavino, M., Tempone, R., and Wang, S., “Fast estimation of expected information gains for Bayesian experimental designs based on Laplace approximations,” *Computer Methods in Applied Mechanics and Engineering*, 2012.

[106] Sebastiani, P. and Wynn, H. P., “Bayesian Experimental Design and Shannon Information,” *In 1997 Proceedings of the Section on Bayesian Statistical Science*, 1997, pp. 176–181.

[107] Khan, S., Bandyopadhyay, S., Ganguly, A. R., Saigal, S., Erickson, D. J., Protopopescu, V., and Ostrouchov, G., “Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data,” *Phys. Rev. E*, Vol. 76, 2007, pp. 026209.

[108] Shewry, M. C. and Wynn, H. P., “Maximum entropy sampling,” *J. Applied Statistics*, Vol. 14, 1987, pp. 165–170.

[109] Darken, C. and Moody, J. E., “Note on Learning Rate Schedules for Stochastic Optimization,” *NIPS*, 1990, pp. 832–838.

[110] Benveniste, A., Métivier, M., and Priouret, P., *Adaptive algorithms and stochastic approximations*, Applications of mathematics, Springer-Verlag, 1990.

[111] Polyak, B. and Juditsky, A., “Acceleration of Stochastic Approximation by Averaging,” *SIAM Journal on Control and Optimization*, Vol. 30, No. 4, 1992, pp. 838–855.

[112] Spall, J. C., “Implementation of the Simultaneous Perturbation Algorithm for Stochastic Optimization,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 34, No. 3, 1998, pp. 817–823.

[113] Spall, J. C., “A Stochastic Approximation Algorithm for Large-Dimensional Systems in the Kiefer-Wolfowitz Setting,” *Proceedings of the 27th IEEE Conference on Decision and Control*, Vol. 2, December 1988, pp. 1544–1548.

[114] Spall, J. C., “Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation,” *IEEE Transactions on Automatic Control*, Vol. 37, No. 3, March 1992, pp. 332–341.

[115] He, Y., Fu, M. C., and Marcus, S. I., “Convergence of simultaneous perturbation stochastic approximation for nondifferentiable optimization,” *IEEE Transactions on Automatic Control*, Vol. 48, No. 8, 2003, pp. 1459–1463.

[116] Maryak, J. L. and Chin, D. C., “Global Random Optimization by Simultaneous Perturbation Stochastic Approximation,” *Johns Hopkins APL Technical Digest*, Vol. 25, No. 2, 2004, pp. 91–100.

[117] Nocedal, J. and Wright, S. J., *Numerical Optimization*, Springer, 2nd ed., 2006.

[118] Barton, R. R. and Ivey Jr., J. S., "Nelder-Mead Simplex Modifications for Simulation Optimization," *Management Science*, Vol. 42, No. 7, 1996, pp. 954–973.

[119] Spall, J. C., *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, John Wiley & Sons, Inc., 2003.

[120] Ho, Y. C. and Cao, X., "Perturbation Analysis and Optimization of Queueing Networks," *Journal of Optimization Theory and Applications*, Vol. 40, 1983, pp. 559–582.

[121] Glasserman, P., *Gradient Estimation via Perturbation Analysis*, Springer, 1990.

[122] Asmussen, S. and Glynn, P., *Stochastic Simulation: Algorithms and Analysis*, Vol. 57, Springer Verlag, 2007.

[123] Hosder, S., Walters, R., and Perez, R., "A Non-Intrusive Polynomial Chaos Method For Uncertainty Propagation in CFD Simulations," *44th AIAA Aerospace Sciences Meeting and Exhibit*, 2006, AIAA paper 2006-891.

[124] Walters, R. W., "Towards Stochastic Fluid Mechanics via Polynomial Chaos," *41st Aerospace Sciences Meeting and Exhibit*, 2003, AIAA paper 2003-413.

[125] Xiu, D. and Karniadakis, G. E., "A New Stochastic Approach to Transient Heat Conduction Modeling with Uncertainty," *International Journal of Heat and Mass Transfer*, Vol. 46, 2003, pp. 4681–4693.

[126] Marzouk, Y. M., Najm, H. N., and Rahn, L. A., "Stochastic Spectral Methods for Efficient Bayesian Solution of Inverse Problems," *Journal of Computational Physics*, Vol. 224, No. 2, June 2007, pp. 560–586.

[127] Marzouk, Y. M. and Xiu, D., "A Stochastic Collocation Approach to Bayesian Inference in Inverse Problems," *Communications in Computational Physics*, Vol. 6, No. 4, October 2009, pp. 826–847.

[128] Marzouk, Y. M. and Najm, H. N., "Dimensionality Reduction and Polynomial Chaos Acceleration of Bayesian Inference in Inverse Problems," *Journal of Computational Physics*, Vol. 228, No. 6, April 2009, pp. 1862–1902.

[129] Cameron, R. H. and Martin, W. T., "The Orthogonal Development of Non-Linear Functionals in Series of Fourier-Hermite Functionals," *The Annals of Mathematics*, Vol. 48, No. 2, 1947, pp. 385–392.

[130] Clenshaw, C. W. and Curtis, A. R., "A method for Numerical Integration on an Automatic Computer," *Numerische Mathematik*, Vol. 2, 1960, pp. 197–205.

[131] Constantine, P. G., Eldred, M. S., and Phipps, E. T., "Sparse Pseudospectral Approximation Method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 229–232, No. 1, 2012, pp. 1–12.

[132] Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer-Verlag, 2006.

[133] Doucet, A., de Freitas, N., and Gordon, N., *Sequential Monte Carlo Methods in Practice*, Springer, 2001.

[134] Moselhy, T. A. E. and Marzouk, Y. M., "Bayesian inference with optimal maps," *Journal of Computational Physics*, Vol. 231, No. 23, 2012, pp. 7815 – 7850.

[135] Guyon, I., "An introduction to variable and feature selection," *Journal of Machine Learning Research*, Vol. 3, 2003, pp. 1157–1182.

[136] Trefethen, L. N. and Bau, D., *Numerical Linear Algebra*, SIAM: Society for Industrial and Applied Mathematics, 1997.

[137] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D., *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 3rd ed., 1999.

[138] Dempster, A. P., Laird, N. M., and Rubin, D. B., "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 39, No. 1, 1977, pp. 1–38.

[139] Jain, A. K., Murty, M. N., and Flynn, P. J., "Data clustering: a review," *ACM Comput. Surv.*, Vol. 31, No. 3, 1999, pp. 264–323.