Optimum Experimental Design Issues in Functional Neuroimaging Studies

by

Lin Zhou

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved June 2017 by the
Graduate Supervisory Committee:

Ming-Hung Kao, Co-Chair
Bruno Welfert, Co-Chair
Zdzislaw Jackiewicz
Mark Reiser
John Stufken
Jesse Earl Taylor

ARIZONA STATE UNIVERSITY

August 2017

ABSTRACT

Functional magnetic resonance imaging (fMRI) is one of the popular tools to study human brain functions. High-quality experimental designs are crucial to the success of fMRI experiments as they allow the collection of informative data for making precise and valid inference with minimum cost. The primary goal of this study is on identifying the best sequence of mental stimuli (i.e. fMRI design) with respect to some statistically meaningful optimality criteria. This work focuses on two related topics in this research field. The first topic is on finding optimal designs for fMRI when the design matrix is uncertain. This challenging design issue occurs in many modern fMRI experiments, in which the design matrix of the statistical model depends on both the selected design and the experimental subject's uncertain behavior during the experiment. As a result, the design matrix cannot be fully determined at the design stage that makes it difficult to select a good design. For the commonly used linear model with autoregressive errors, this study proposes a very efficient approach for obtaining high-quality fMRI designs for such experiments. The proposed approach is built upon an analytical result, and an efficient computer algorithm. It is shown through case studies that our proposed approach can outperform the existing method in terms of computing time, and the quality of the obtained designs. The second topic of the research is to find optimal designs for fMRI when a wavelet-based technique is considered in the fMRI data analysis. An efficient computer algorithm to search for optimal fMRI designs for such cases is developed. This algorithm is inspired by simulated annealing and a recently proposed algorithm by Saleh *et al.* (2017). As demonstrated in the case studies, the proposed approach makes it possible to efficiently obtain high-quality designs for fMRI studies, and is practically useful.

DEDICATION

To my parents

Mr. Huaqing Zhou and Mrs. Shunhua Zhu

and

to my husband

Arthur

ACKNOWLEDGMENTS

My graduate study has been a long journey. Over the years I have gained invaluable knowledge and skills. I am grateful to everyone who supported me along this journey.

I would like to express my sincere gratitude and appreciation to my co-chairs Dr. Ming-Hung Kao and Dr. Bruno Welfert, for their patient guidance and encouragement. Their depth of knowledge and enthusiasm about research motivated me and drove me to develop my interest and understanding of my research topics.

Dr. Kao has been a tremendous mentor for me. He spent countless hours discussing my research and proofreading my papers. I appreciate his unlimited dedication towards my studies. His ideas, advices and funding during all those years were of incalculable value.

Dr. Welfert has been another fantastic mentor. He is friendly, supportive and helpful. Through numerous meetings, he has always encouraged me to pursue my own interests and has provided insightful discussions about my research throughout my time as his student.

I would also like to extend my appreciation to my committee members: Dr. Zdzislaw Jackiewicz, Dr. Mark Reiser, Dr. John Stufken and Dr. Jesse Earl Taylor, for their valuable questions and suggestions during my prospectus and defense. A special thanks to Dr. Zdzislaw Jackiewicz, he was the first one who motivated me to join the PhD program. Thank you for your encouragement.

I would like to thank Dr. Hans D. Mittelmann, for his motivating discussions and professional advice on optimization problem. I am also very grateful to my anonymous external reviewer, for his valuable comments in my work.

I gratefully acknowledge the funding received towards my PhD which allowed me to pursue my graduate studies from School of Mathematical and Statistical Sciences (SoMSS) at ASU (teaching assistantship) and Dr. Ming-Hung Kao (research assistantship supported by an NSF CAREER grant).

I am deeply grateful to the faculty members who have taught me here at ASU. I thank all the members of staff at SoMSS, especially Graduate Program Coordinators Debbie Olson

and Jennifer May; and IT support Renate Mittelmann. I would also like to thank my fellow graduate students, friends here at ASU who made my graduate life full of joy.

I am forever thankful to my beloved parents, I would not be where I am today without your care and support. A heartfelt thanks go to my husband, Arthur, for his unconditional love and support. I am thankful to all my family and friends who have helped me achieve this goal.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

Table Page

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Humans have explored and discovered our world and acquired tremendous accomplishments, yet the human brain remains a mystery. Recent surveys show that brain diseases affect more people than ever before, and deaths from such diseases increase year by year (Pritchard *et al.*, 2013). Understanding the brain disorders and how our brain works is arguably a crucial task of our time. The importance of this task is also highlighted in the Brain Research through Advancing Innovative Neurotechnologies (BRAIN) Initiative, a multimillion-dollar research project announced by the White House in 2013, and also in the Europe's Human Brain Project that was launched in the same year. The development of functional magnetic resonance imaging (fMRI) technology has led to a better understanding of brain function and disease. It is a relatively inexpensive and completely noninvasive brain mapping technology (Kwong and Chesler, 1995), and is one of the most popular techniques that helps to provide insights into the way brain works. With equipment like fMRI, functional neuroimaging experiments are widely conducted in various research fields such as psychology, neuroscience, and education for studying brain activity in response to mental stimuli; other than non-clinical applications, physicians also use fMRI as a tool for clinical diagnosis, surgery planning and evaluation, etc., see also Lindquist *et al.* (2008), and a special issue on clinical applications of fMRI in Neuropsychology Review, Vol. 17 et al. (2007). The use of fMRI is arguably an important advance in neuroscience, and it has many practical applications.

In an fMRI experiment, the investigator may present a sequence of tens or thousands of mental stimuli (e.g., images or sounds) to the experimental subject. Each stimulus evokes neuronal activity at some regions of the subject's brain. The neuronal activity leads to a change in the concentrations of the oxygenated and deoxygenated blood in the cerebral

vessels. It is measured as the blood oxygenation level dependent (BOLD) signals by an fMRI scanner, which repeatedly scans each of the brain voxels (three-dimensional image elements) every, say, 2 seconds. The BOLD signals collected over time form an fMRI time series. In particular, there will be, say $64 \times 64 \times 30$ voxels, each having a size of, say $3 \times 3 \times 5mm^3$ (Lazar, 2008, Section 2.1.1). An fMRI time series is acquired from each of these brain voxels. These time series are then analyzed by some statistical methods to make inference about the brain activity. In such an analysis, neuroscientists may be interested in studying the hemodynamic response function (HRF) whose characteristics such as the peak, time-to-peak, time-to-onset or shape help to provide some insights into the underlying brain activity evoked by the stimulus (see Section 2.2 for further discussions). Unfortunately, the fMRI data almost always contain high amount of noises, making it impossible to identify the HRFs by visual inspections. Statistical methods allowing researcher to extract useful information from the noisy fMRI data are thus crucial. As an integral part of the statistical process, selecting a high-quality fMRI experimental design, which determines the onset times and the order of stimuli to be presented to an experimental subject, is essential to allow researchers to effectively utilize the limited resource to collect informative data for making a precise and valid inference about the HRF and/or efficiently achieving other study objectives of interest. It is one of the key steps to the success of fMRI studies.

In this work, we study two related topics on optimal fMRI experimental designs for popular linear models with autocorrelated errors. The first topic is on finding optimal designs where the focus is on fMRI experiments with an uncertain design matrix. In traditional fMRI studies, the design matrix of the statistical model for analyzing fMRI data normally can be completely determined by the selected design. However, this no longer holds true for some modern fMRI experiments that aim at investigating the brain activity evoked by the subject's reactions to the stimuli (e.g., the subject's answers to the presented questions). Cordes *et al.* (2012) reported an experiment of this sort. For such experiments, the design matrices will depend not only on the selected design, but also on the subject's reaction. As

the subject's reactions are uncertain at the design stage (before the experiment starts), it is unlikely to have an accurate evaluation of the quality of the designs. This makes it very challenging to select good designs suited to this type of modern experiments. Following Cordes *et al.* (2012), we consider linear regression models with first order autoregressive errors (an AR(1) error model). The goal is to develop an efficient and effective approach for finding high-quality designs to improve the quality of fMRI experiments when the design matrix is uncertain. To that end, we build our design selection approach on an analytical result and an efficient computer algorithm. Specifically, we analytically derive a closed-form expression for our design selection criterion that allows us to evaluate the quality of designs without much computational effort. We then adapt the genetic algorithm (GA) put forward by Kao *et al.* (2009) to search for an fMRI design optimizing the criterion. As demonstrated in our case studies in Chapter 3, our criterion can serve as an inexpensive, but effective surrogate of the design selection criterion proposed by Cordes *et al.* (2012). More importantly, our approach is much faster than that of Cordes *et al.* (2012).

In the past years, research in fMRI grew rapidly, a variety of statistical models have been developed for analyzing fMRI data. One of these models is the wavelet-based model. Another topic of interest in this study is to find a fast algorithm to obtain optimal fMRI designs for such models. Jeong *et al.* (2013) recently proposed a wavelet-based approach for regression models with long memory processes. They show that the discrete wavelet transforms (DWTs) can serve as whitening filter for such processes by simplifying the dense covariance matrix into a sparse form. They also demonstrate that their method is useful for analyzing some fMRI data. Although wavelet-based models are not uncommon in fMRI data analysis, to our knowledge, there is no research article on optimal design studies based on these models. Due to the complexity of the information matrix, which is used to evaluate the quality of designs, it is very challenging to find an optimal design even for simple AR(1) model; see also Kao *et al.* (2009). Thus, an efficient computational algorithm for searching for optimal design from a vast design space is necessary. The need for an efficient compu-

tational approach for obtaining optimal fMRI designs is even greater when the model is as complex as that of Jeong *et al.* (2013). To develop such an efficient approach, we first consider to adapt the approach of Saleh *et al.* (2017) that is a heuristic algorithm for obtaining D-optimal fMRI experimental designs under the traditional, simple models. This approach was inspired by the widely used coordinate-exchange algorithm; see Gotwalt *et al.* (2009); Jones and Goos (2007); Meyer and Nachtsheim (1995). Saleh *et al.* (2017)'s algorithm is shown to be superior than the genetic algorithm of Kao *et al.* (2009). To help alleviating this drawback, we propose another algorithm by incorporating simulated annealing (SA). In Chapter 4, we demonstrate the performance of the new algorithm through case studies. Our results show that the new algorithm can quickly obtain similar D-optimal designs comparing with Saleh *et al.* (2017)'s algorithm. In addition, the newly proposed algorithm is suitable for other optimality criteria, such as A-optimality.

The following chapters are organized as follows. In Chapter 2, we provide background information about fMRI time series and fMRI designs; we then present our methodology for finding fMRI experiments with an uncertain design matrix based on general linear models with AR(1) errors in Chapter 3 and illustrate some case studies to demonstrate the usefulness of our approach. In Chapter 4, we shift our attention to a traditional fMRI experimental setting that the design matrix depends only on the selected design, and can be fully specified at the design stage. But, in contrast to the existing studies on optimal fMRI designs, we consider the advanced wavelet-based models. A fast computer algorithm for obtaining optimal fMRI designs for such a case is proposed. We then present some numerical simulations to show the performance of new algorithm. We note that there exist several different statistical methods for analyzing fMRI data. A comparison of these methods might be of interest, but is beyond the scope of this study. Interested readers are referred to, e.g. Lazar (2008) and Lindquist *et al.* (2008) for an overview of some of these statistical analysis methods. Here, our main focus is on studying optimal experimental designs for two challenging, commonly encountered situations.

Chapter 2

BACKGROUND

2.1 fMRI Experimental Design

In an fMRI study, there might be tens to hundreds of brief stimuli of one or more types presented to the subject at different time points. Each stimulus may last several milliseconds to a few seconds, immediately followed by a period of 'control' such as a rest period. For example, an experiment might involve 1-second pictures of familiar faces, which form the first stimulus type, and 1-second pictures of unfamiliar faces that form the second stimulus type. Each stimulus can possibly appear every τ_{ISI} seconds, where τ_{ISI} is a pre-specified time (e.g., 4 seconds); τ_{ISI} is sometimes called the inter-stimulus-interval. During the period from the offset of a stimulus to the onset of the next one, the subject is exposed to the control (e.g., rest or visual fixation). An experiment can have a duration of several minutes (e.g., 10 minutes). A design for such an experiment is often represented as an ordered sequence of N elements; i.e. $\boldsymbol{d} = \{d_1, ..., d_N\}$, where N is typically tens or hundreds, and is determined by τ_{ISI} and the duration of the experiment. With Q stimulus types, each element d_n in a design is an integer between 0 and Q. For example, a design with $Q = 2$ may look like $\boldsymbol{d} = \{1, 0, 2, 1, ..., 0\}$. The n^{th} position of $\boldsymbol{d}$ corresponds to time $(n-1)\tau_{ISI}$, $n = 1, ..., N$. Time 0 is typically set to the time point when the first valid fMRI measurement is acquired by the fMRI scanner. When $d_n = q > 0$, there is an onset of the q^{th}-type stimulus at the n^{th} time point. When $d_n = 0$, there is no stimulus onset at that time point. For example, a '1' in $\boldsymbol{d}$ may indicate an appearance of a familiar face, a '2' is for an unfamiliar face, and a '0' means that none of these pictures occurs.

2.2 Hemodynamic Response Function (HRF)

When a selected design is presented to the subject, some voxels of the subject's brain may be activated by the stimuli. Such neuronal activity leads to several physiological changes in cerebral blood vessels. Among different fMRI techniques, the BOLD (blood-oxygenation-level-dependent) fMRI is the most common form of fMRI (Barbee *et al.*, 2012). This technique detects neural activity using the oxygen level in the blood as a marker, and the fMRI scanner picks up a functional signal based on a rise or fall in the concentration of the oxygenated blood in the cerebral blood vessels. This leads to a change in the strength of the local magnetic field around the activated brain regions. The fMRI scanner collects the BOLD signal at regular time points from each of the brain voxels. These signals reflect the fluctuation in the strength of the local magnetic field that help to infer the underlying brain activity. In particular, from each brain voxel, an fMRI measurement is collected every τ_{TR} seconds (e.g., $\tau_{TR} = 2\,\text{seconds}$) to form an fMRI time series. τ_{TR} is sometimes referred to as time-to-repetition. These collected time series are then analyzed for making inference about the brain activity in response to the stimuli. The changes in the blood flow upon stimulus can be described via the hemodynamic response function, HRF, which is a function of time modeling the stimulus-induced change in the concentrations of the oxy- and deoxy-blood in the cerebral blood vessels at a voxel. The HRF typically has a long duration (e.g., 30 seconds) relative to τ_{ISI} and τ_{TR}. It may look like the green curve presented in the top plot of Figure 2.1. It is commonly assumed that, at each voxel, the stimuli of the same type have the same HRF, whereas stimuli of different types may have different HRFs. Consequently, there will be Q possibly different HRFs in cases with Q stimulus types. It is also assumed that overlapping HRFs add up linearly. Figure 2.1 presents an example with two onsets of stimulus of the same type. They give rise to two HRFs (solid green curve and broken red curve in the top plot) of a certain voxel. Since the two stimuli are of the same type, the shapes of the HRFs are the same; furthermore, these two HRFs accumulate linearly to form the solid blue curve as shown in the bottom plot of Figure 2.1.

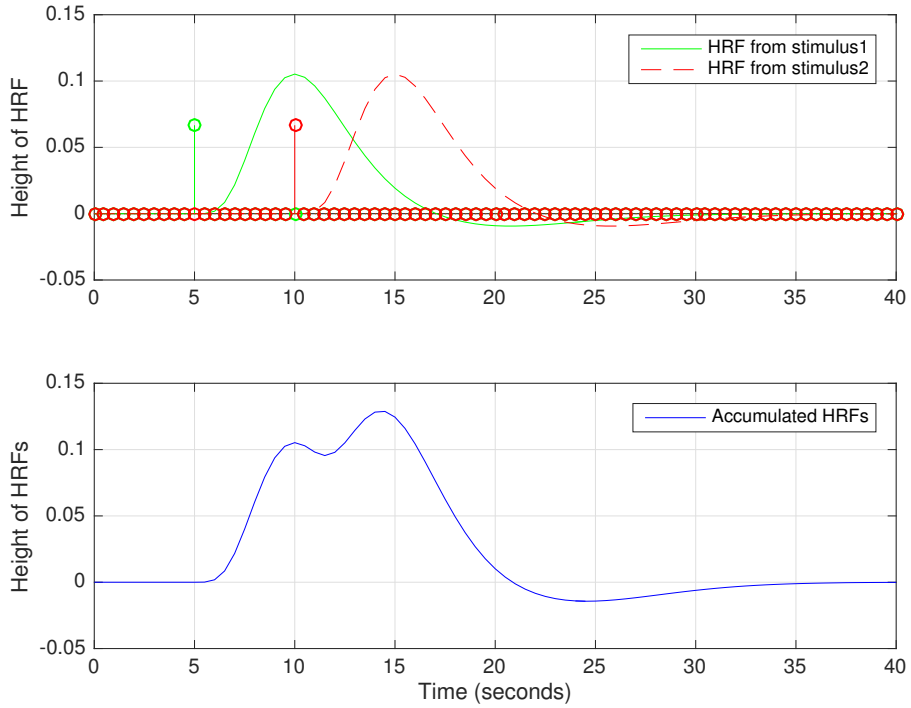

Figure 2.1: Illustration of the Hemodynamic Response Function (HRF): (Top) Two stimuli (bars) with same stimulus type are presented to an subject with 5 s apart; the responses of certain voxel to first stimulus (green solid curve) and second stimulus (red broken curve); (Bottom) the accumulated HRFs (blue curve) is formed by these two HRFs.

To understand the neural-BOLD relationship, there are two major objectives. One study objective of interests is to estimate these Q HRFs, and the other study objective is to detect the regions of brain corresponding to the neural activity in response to the stimulus. The former problem is sometimes referred to as the estimation problem, whereas the latter one is sometimes termed as the detection problem. For the detection problem, the HRF is commonly approximated by the product of an assumed basis function of the HRF and an unknown amplitude. One of the popular basis function for the HRF is the double gamma function which has the following form and is similar to the one proposed in Glover (1999):

$$h^*(t) = \frac{t^5 e^{-t}}{5!} - \frac{1}{6} \cdot \frac{t^{15} e^{-t}}{15!}. \tag{2.1}$$

Typically, $h^*(t)$ is (nearly) zero after 32 s. It is also noteworthy that, with given τ_{ISI} and τ_{TR}, the HRF heights that could possibly contribute to an observed BOLD signal will occur at 0 s and every ΔT seconds after the stimulus onset. Here, the discretization interval ΔT is defined as the greatest time making both $\tilde{\tau}_{ISI} = (\tau_{ISI}/\Delta T)$ and $\tilde{\tau}_{TR} = (\tau_{TR}/\Delta T)$ integers.

2.3 Temporal Autocorrelations in fMRI Time Series

General linear models (GLMs) are commonly used in fMRI data analysis. It was first introduced into neuroimaging by Friston *et al.* (1994). In fMRI time series, researchers observed temporally autocorrelated noises in data due to low-frequency physiological fluctuation (Weisskoff *et al.*, 1993). When there are correlated errors in the model, statistical inferences can be made, e.g., using generalized least squares. Bullmore *et al.* (1996) proposed a pre-whitening approach, assuming the errors follow a first order autoregressive model, or AR(1). Other than AR(1), there are variety of ways to model the noise, such as m^{th} order autoregressive, or AR(m) (Worsley *et al.*, 2002), first order autoregressive (AR) plus white noise model (Purdon and Weisskoff, 1998), first order autoregressive moving-average, ARMA(1,1) model (Purdon *et al.*, 2001), also Zarahn *et al.* (1997) first suggested that temporal autocorrelation in fMRI can be well modeled by $1/f$ process, or long memory process. Among these models, the most commonly used model is probably the general linear model with AR(1) noise. In the next section, we will introduce such a model. Similar models will be considered in the first part of our study. A general linear model with a long memory process that we consider in the second part of the study will also be introduced.

2.4 General Linear Models

The general linear model is commonly used in fMRI data analysis (Worsley and Friston, 1995; Friston *et al.*, 1995; Dale, 1999). In the subsequent subsections, we present some very popular general linear models. Based on these models, we would like to develop some optimal designs.

2.4.1 The First Order Autoregressive Model

When analyzing the fMRI data, it is not uncommon to assume that the noise follows a first order autoregressive process; see also Bullmore *et al.* (1996). Depending on the objective, either the estimation of the HRFs or the detection of brain activated regions, the following linear models are commonly considered (e.g., Liu and Frank, 2004):

$$\boldsymbol{y} = \sum_{k=1}^{K}\sum_{q=1}^{Q} x_{q,k}h_{q,k} + \boldsymbol{S\gamma} + \boldsymbol{\epsilon} = \boldsymbol{X\zeta} + \boldsymbol{S\gamma} + \boldsymbol{\epsilon}, \quad \textit{and} \tag{2.2}$$

$$\boldsymbol{y} = \sum_{k=1}^{K}\sum_{q=1}^{Q} x_{q,k}h_k^*\theta_q + \boldsymbol{S\gamma} + \boldsymbol{\eta} = \boldsymbol{XH\theta} + \boldsymbol{S\gamma} + \boldsymbol{\eta}. \tag{2.3}$$

Here, Model (2.2) is used for estimating HRFs, vector $\boldsymbol{y} = [y_1, ..., y_T]'$ represents the fMRI measurements obtained from a voxel every τ_{TR} seconds. The unknown parameter vector is $\boldsymbol{\zeta} = [\boldsymbol{\zeta}_1', \ldots, \boldsymbol{\zeta}_K']'$, $\boldsymbol{\zeta}_k = [h_{1,k}, ..., h_{Q,k}]'$. The $h_{q,k}$ represents the height of the HRF evoked by q^{th}-type stimulus at the k^{th} time point that contribute to $\boldsymbol{y}$. The pre-specified integer K is sufficiently large so that the height of the HRF is negligible after K time points. In particular, with function (2.1), the length is set to $K = \lfloor 1+(32/\Delta T)\rfloor$, see Kao *et al.* (2009), where ΔT is the discretization interval as defined previously; $\lfloor a\rfloor$ is the integer part of a. The $T \times QK$ matrix $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_K]$, $\boldsymbol{X}_k = [\boldsymbol{x}_{1,k}, \ldots, \boldsymbol{x}_{Q,k}]$ is the 0-1 design matrix. $\boldsymbol{S\gamma}$ is a nuisance term modeling the possible drift/trend of $\boldsymbol{y}$ with $\boldsymbol{S}$ being a specified matrix and $\boldsymbol{\gamma}$ being the corresponding parameter vector. The vector $\boldsymbol{\epsilon}$ consists of the T correlated error terms that are assumed to follow an AR(1) process. Model (2.3) is for detecting brain activation, h_k^* is the k^{th} height of the assumed HRF basis, θ_q represents the response amplitude for the q^{th}-type stimulus. The vector $\boldsymbol{\eta}$ is correlated errors that also are assumed to follow an AR(1) process. The remaining terms are as in Model (2.2).

We note that $\boldsymbol{X}$ is determined by design $\boldsymbol{d}$. An example for construction of design matrix for cases where $\tau_{ISI} = \tau_{TR}$ can be found in Zhou (2014). Here, we present a case where $\tau_{ISI} \neq \tau_{TR}$ in Example 1.

Example 1 *Let the design* $\boldsymbol{d} = \{1, 2, 1, 0, 2, \ldots, 0\}$*, where* $Q = 2$*,* $\tau_{ISI} = 3\,\mathrm{s}$*,* $\tau_{TR} = 1.5\,\mathrm{s}$*, and* $\Delta T = 1.5\,\mathrm{s}$*. To construct* $\boldsymbol{X}_k$*, where* $k = 1, \ldots, K$*;* $K = \lfloor 1 + (32/\Delta T) \rfloor = 22$*, we first construct two indicator vectors* $\boldsymbol{\delta}_1$ *and* $\boldsymbol{\delta}_2$*. The lengths of these indicator vectors are the same as the length of* $\boldsymbol{d}$*. The* n^{th} *element of* $\boldsymbol{\delta}_q$ *is* $\delta_{q,n} = 1$ *when* $d_n = q$*, where* d_n *denotes the* n^{th} *entry of the design* $\boldsymbol{d}$*,* $n = 1, \ldots, N$*. Thus we have* $\boldsymbol{\delta}_1 = \{1, 0, 1, 0, 0, \ldots, 0\}$ *and* $\boldsymbol{\delta}_2 = \{0, 1, 0, 0, 1, \ldots, 0\}$ *respectively. We then represent the onset times of the stimuli in terms of* ΔT*, that is* $\boldsymbol{\omega}_q = \boldsymbol{\delta}_q \otimes [1, \mathbf{0}'_{\tilde{\tau}_{ISI}-1}]'$*, where* $\otimes$ *is the Kronecker product,* $\mathbf{0}_a$ *is a zero vector of length* a *and* $\tilde{\tau}_{ISI} = (\tau_{ISI}/\Delta T) = 2$*;* $q = 1, 2$*. Its length is adjusted to* $N \times \tilde{\tau}_{TR}$ *by adding zeros or leaving out the last few elements. The design matrix is then* $\boldsymbol{X}_k = (\boldsymbol{I}_T \otimes [1, \mathbf{0}'_{\tilde{\tau}_{TR}-1}])\boldsymbol{L}^{k-1}[\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_Q]$*, where* $\boldsymbol{I}_a$ *is an* $a \times a$ *identity matrix and* $\boldsymbol{L}$ *is a lower shift matrix, i.e.,* $\boldsymbol{L} = \begin{bmatrix} \mathbf{0}' & 0 \\ \mathbf{I} & \mathbf{0} \end{bmatrix}$*. In particular,* $\boldsymbol{X}_k$ *can be obtained from shifting the elements of* $\boldsymbol{X}_{k-1}$ *downward by one position, with zeros appearing in the top row; as illustration, we present* $\boldsymbol{X}_1\boldsymbol{\zeta}_1$ *to* $\boldsymbol{X}_3\boldsymbol{\zeta}_3$ *for Model* (2.2)*, note that the first column of* $\boldsymbol{X}_k$ *is for stimulus type* $q = 1$*, and the second column is for stimulus type* $q = 2$*:*

$$\boldsymbol{X}_1\boldsymbol{\zeta}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \begin{bmatrix} h_{1,1} \\ h_{2,1} \end{bmatrix}, \boldsymbol{X}_2\boldsymbol{\zeta}_2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \begin{bmatrix} h_{1,2} \\ h_{2,2} \end{bmatrix}, \boldsymbol{X}_3\boldsymbol{\zeta}_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \begin{bmatrix} h_{1,3} \\ h_{2,3} \end{bmatrix}.$$

□

For Model (2.2), one important design goal is to select an fMRI design $\boldsymbol{d}$ that yields the most precise generalized least square estimate (GLSE) of $\boldsymbol{\zeta}$. For detection problem, the main concern is on studying the strength of brain activation for each stimulus type,

typically through obtaining the GLSE of $\boldsymbol{\theta}$ with Model (2.3). Our target is to identify a $\boldsymbol{d}$ that optimizes some statistically meaningful function of the information matrix of $\boldsymbol{\zeta}$ or $\boldsymbol{\theta}$. Note that the information matrix $\boldsymbol{M}$ is inversely proportional to the covariance of the GLSE of $\boldsymbol{\zeta}$ or $\boldsymbol{\theta}$, and it depends on the design $\boldsymbol{d}$ through the design matrix $\boldsymbol{X}$. Ideally, we would like to find a $\boldsymbol{d}$ that 'maximizes' the numerical value of a real-valued function (e.g. determinant, trace etc.) of the information matrix. However, this goal is often not achievable. A common strategy is thus to find a $\boldsymbol{d}$ that optimizes $\phi(\boldsymbol{M})$ for some real function ϕ; see Chapter 3 for a further discussion of the information matrix $\boldsymbol{M}$ and popularly used ϕ in fMRI. With a selected ϕ, one may utilize a computer algorithm such as the genetic algorithm (GA) of Kao *et al.* (2009) to search for an optimal $\boldsymbol{d}$. A description of an adapted genetic algorithm that we use can be found in Section 2.7.

Another fMRI data analysis method that also draws some attention is to consider a long memory process. In the second part of our study, we consider such a model. In what follows, we provide some background knowledge about this modeling technique.

2.4.2 The Long Memory Processes Model

Long memory processes have been widely observed in nature, Zarahn *et al.* (1997) indicated that the temporal autocorrelation in a BOLD fMRI time series can be well modeled by long memory processes. Specifically, one may consider the following linear models for modeling a BOLD time series:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\zeta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim (0, \Sigma_{\boldsymbol{\epsilon}}); \tag{2.4}$$

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{H}\boldsymbol{\theta} + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim (0, \Sigma_{\boldsymbol{\eta}}). \tag{2.5}$$

Here, Model (2.4) is for estimating HRFs, the design matrix $\boldsymbol{X}$, the observations $\boldsymbol{y}$ and the unknown vector $\boldsymbol{\zeta}$ are the same as in Model (2.2); the highly correlated error terms $\boldsymbol{\epsilon}$ are modeled as a long memory process with mean 0 and covariance matrix $\Sigma_{\boldsymbol{\epsilon}}$. Model (2.5) is used for detecting brain activation, the vector $\boldsymbol{\eta}$ is the correlated errors and is assumed to follow a long memory process with mean 0 and covariance matrix $\Sigma_{\boldsymbol{\eta}}$, the remaining

terms are the same as in Model (2.3). In both models, the long-range correlation feature of long memory processes give rise to dense covariance matrices that make existing methods of statistical inference computationally expensive. The wavelet-based method with Model (2.4) proposed by Jeong *et al.* (2013) simplifies the dense variance-covariance matrix by employing discrete wavelet transforms (DWTs) as whitening filter, it is shown to be suitable for fMRI time series.

Wavelets have been recognized as an useful method for fMRI data analysis, see Ruttimann *et al.* (1998); Brammer (1998) among others. Thus, having knowledge for selecting a good design to obtain the most informative data for such models is necessary. As for fMRI time series, the most important property of DWTs is that the strongly correlated long memory process can be transformed into wavelet coefficients, which are approximately uncorrelated. Specifically, after wavelet transform, the dense covariance matrix can be approximated as a diagonal matrix in wavelet domain. More details can be found in Section 2.6.2. Based on such property, for Models (2.4) and (2.5), we will focus on developing an efficient and effective algorithm for obtaining optimal designs for cases where such a wavelet-based method is applied in fMRI data analysis.

In the next section, we will provide more details about long memory process and the decorrelation property that discrete wavelet transforms have.

2.5 Long Memory Process

Long memory, or long range dependence is a phenomenon that has broadly been observed in many research fields, such as civil engineering (e.g. Hurst (1951)), econometrics (e.g. Baillie (1996)), weather forecasting (e.g. Caporin *et al.* (2009)) among others. In time series data, it refers to the rate of decay of statistical dependence between two time points as we increase the time interval, long memory is the situation when the dependence decays more slowly than an exponential decay. Some well-known long memory processes include fractional Brownian motion (fBm), fractional Gaussian noise (fGn) and autoregressive fractionally integrated moving average (ARFIMA) processes. In the second topic of this work,

we focus on wavelet-based linear models with long memory processes. Here, we provide some background on long memory processes.

Continuous-Time Stochastic Process

Standard fractional Brownian motion (fBm) is a continuous-time Gaussian process $B^H(t)$ on $[0, T]$, which starts at 0, has mean 0 for all t, and has the covariance function (Taqqu *et al.*, 1995):

$$\mathrm{Cov}(B^H(t), B^H(s)) = \mathbb{E}[B^H(t)B^H(s)] = \frac{1}{2}(|t|^{2H} + |s|^{2H} - |t-s|^{2H}). \tag{2.6}$$

The variance of $B^H(t)$ is,

$$\mathrm{Var}(B^H(t), B^H(t)) = |t|^{2H}. \tag{2.7}$$

It is self-similar and characterized by Hurst parameter H assuming any value in the interval $[0, 1]$. The self-similarity means that the process is invariant in distribution under a suitable change of scale. Mathematically, $B^H(t)$ has normal distribution with mean 0, and $\mathrm{Cov}(B^H(at), B^H(as)) = |a|^{2H}\,\mathrm{Cov}(B^H(t), B^H(s))$. Its special case, when $H = \frac{1}{2}$, is standard Brownian motion or Wiener process, denoted by $B(t)$. It is a continuous-time stochastic process with independent Gaussian increments. When $H > \frac{1}{2}$, the process has long-range dependence or long memory.

There are many ways to define fractional Brownian motion. Mandelbrot and Ness (1968) defined fractional Brownian motion as the $(\frac{1}{2} - H)^{th}$ *fractional derivative* or *integral* (in the sense of Weyl (1917)) of Brownian motion. The integral representation for fBm is:

$$\begin{aligned} &B^H(t) - B^H(0) \\ &= \frac{1}{\Gamma(H + 1/2)} \left\{ \int_{-\infty}^{0} \left[(t-u)^{H-\frac{1}{2}} - (-u)^{H-\frac{1}{2}} \right] \mathrm{d}B(u) + \int_{0}^{t} (t-u)^{H-\frac{1}{2}} \mathrm{d}B(u) \right\}, \end{aligned} \tag{2.8}$$

where $B^H(0)$ is the starting value at time 0, Γ is the Gamma function $\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}e^{-x}\mathrm{d}x$, $B(t)$ is Brownian motion, and $0 < H < 1$ is the Hurst parameter.

By definition, the formal derivative of Brownian motion $B'(t)$ is the continuous-time white noise. By analogy, we define the fractional Gaussian noise (fGn) as the derivative of

fractional Brownian motion. It may also be thought of as the $(H-\frac{1}{2})^{th}$ *fractional derivative* of continuous-time white noise. Note that the fractional Gaussian noise reduces to white noise when $H = \frac{1}{2}$.

Fractional Brownian motion and its increments fractional Gaussian noise are both continuous-time stochastic processes. However, some data set such as fMRI time series are discrete data, Hosking (1981) proposed a model for discrete-time version of fGn, named fractionally differenced Gaussian noise, we will call it discrete fractional Gaussian noise (dfGn) hereafter.

Discrete-Time Stochastic Process

Discrete fractional Gaussian noise (dfGn) can be used to model discrete time series that has long-range dependence and self-similarity. Here, we will describe the derivation of such noise that is proposed by Hosking (1981). It is also known as ARFIMA processes.

The discrete time analogue of Brownian motion, $\{x_t\}$, can be defined as $\nabla x_t = (1 - \mathcal{B})x_t = \epsilon_t$, where $\mathcal{B}$ is the backward shift operator defined by $\mathcal{B}x_t = x_{t-1}$ and ϵ_t are independent identically distributed random variables. The first difference of $\{x_t\}$ is the discrete-time white noise process $\{\epsilon_t\}$.

By analogy with the above definition of continuous-time white noise, we defined the discrete fractional Gaussian noise with parameter H to be the $(H-\frac{1}{2})^{th}$ *fractional difference* of discrete-time white noise. The fractional difference operator ∇^{d^*} is defined by a binomial series:

$$\nabla^{d^*} = (1-\mathcal{B})^{d^*} = \sum_{k=0}^{\infty} \binom{d^*}{k} (-\mathcal{B})^k = 1 - d^*\mathcal{B} - \frac{1}{2}d^*(1-d^*)\mathcal{B}^2 - \dots. \tag{2.9}$$

Since $\binom{n}{j} = (-1)^j \binom{j-n-1}{j}$ and $\binom{n}{j} = \frac{\Gamma(n+1)}{\Gamma(j+1)\Gamma(n-j+1)}$, then we have

$$(1-\mathcal{B})^{d^*} = \sum_{k=0}^{\infty} (-1)^k \binom{d^*}{k} \mathcal{B}^k = \sum_{k=0}^{\infty} \binom{k-d^*-1}{k} \mathcal{B}^k = \sum_{k=0}^{\infty} \frac{\Gamma(k-d^*)}{\Gamma(k+1)\Gamma(-d^*)} \mathcal{B}^k,$$

where the difference parameter $d^* = H - \frac{1}{2}$, so that the continuous-time fractional Gaussian noise has as its discrete-time analogue the process $\nabla^{d^*} x_t = \epsilon_t$, where $\{\epsilon_t\}$ is a white noise process.

Thus, discrete fractional Gaussian noise process $\{x_t\}$ is defined as

$$\nabla^d x_t = \epsilon_t, \text{ where } \epsilon_t \sim (0, \sigma_\epsilon^2).$$

The spectral density of $\{x_t\}$ is $s(\omega) = \sigma_\epsilon^2 (2\sin(\frac{1}{2}\omega))^{-2d^*}$ for $0 < \omega \leq \pi$, $\omega = 2\pi f$ (Hosking, 1981). Note that $\{\epsilon_t\}$ is a white noise, we have $\sigma_\epsilon^2 = 1$.

Discrete fractional Gaussian noise can also be defined as the increments of fractional Brownian motion, by

$$x_t = B^H(t+1) - B^H(t),$$

then the covariance of dfGn is

$$\begin{aligned}
\text{Cov}(x_t, x_s) &= \text{Cov}(B^H(t+1) - B^H(t), B^H(s+1) - B^H(s)) \\
&= \mathbb{E}[(B^H(t+1) - B^H(t))(B^H(s+1) - B^H(s))] \\
&= \mathbb{E}[B^H(t+1)B^H(s+1) - B^H(t+1)B^H(s) - B^H(t)B^H(s+1) + B^H(t)B^H(s)] \\
&= \frac{1}{2}(|t+1-s|^{2H} + |t-s-1|^{2H} - 2\,|t-s|^{2H}).
\end{aligned}$$

Now, let $t - s = k$, the autocovariance function is

$$\begin{aligned}
\gamma(k) &= \text{Cov}(x_t, x_s) \\
&= \frac{1}{2}(|k+1|^{2H} + |k-1|^{2H} - 2k^{2H}] \qquad (2.10) \\
&= \frac{1}{2}k^{2H}\left(\left|\frac{k+1}{k}\right|^{2H} + \left|\frac{k-1}{k}\right|^{2H} - 2\right) \\
&= \frac{1}{2}k^{2H}\left(\left|1+\frac{1}{k}\right|^{2H} + \left|1-\frac{1}{k}\right|^{2H} - 2\right), \qquad (2.11)
\end{aligned}$$

by writing down the Taylor series for $|1 + 1/k|^{2H} + |1 - 1/k|^{2H} - 2$, (2.11) becomes

$$\begin{aligned}
\gamma(k) &\approx \frac{1}{2}k^{2H}[2H(2H-1)k^{-2}] \\
&= [H(2H-1)]k^{2H-2}. \qquad (2.12)
\end{aligned}$$

2.5.1 Simulation of Discrete Fractional Gaussian Noise (dfGn)

Jeong *et al.* (2013) considered a linear regression model with long memory, more specifically, error term was modeled as discrete fractional Gaussian noise. Additionally, Bullmore

et al. (2001); Fadili and Bullmore (2002); Bullmore *et al.* (2004) also consider long memory errors in fMRI using discrete fractional Gaussian noise (dfGn). There are many methods for simulating dfGn, Jean-Francois (2007) describe several popular methods to simulate dfGn and fBm. Circulant embedding method (Wood and Chan, 1994; Dietrich and Newsam, 1997) is an efficient way to simulate discrete fractional Gaussian noise. In the second part of the work, we use this method to simulate dfGn. This method extracts the square root of the autocovariance matrix of the discrete fractional Gaussian noise as in (2.10), denoted by $\boldsymbol{G}$. To generate a vector with N random variables with mean 0 and covariance matrix $\boldsymbol{G}$, Wood and Chan (1994) proposed to embed $\boldsymbol{G}$ in an $M \times M$ circulant matrix $\boldsymbol{C}$, where $M = 2^g$ for some integer g, and $M \geq 2(N-1)$. By generating a random vector $\boldsymbol{y} = (y_0, y_1, \ldots, y_{m-1})' \sim N_m(0, \boldsymbol{C})$, then, with appropriate construction of $\boldsymbol{C}$, we have $\boldsymbol{y} = (y_0, y_1, \ldots, y_{n-1})' \sim N_n(0, \boldsymbol{G})$.

The circulant matrix $\boldsymbol{C}$ can be constructed as

$$\boldsymbol{C} = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_{m-1} \\ c_{m-1} & c_0 & c_1 & \cdots & c_{m-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & c_3 & \cdots & c_0 \end{bmatrix},$$

where $c_0 = 0$ and

$$c_m = \begin{cases} \gamma(m), & m = 1, 2, \ldots, N-1 \\ \gamma(M-m), & m = N-1, \ldots, M-1 \end{cases},$$

$\gamma(\cdot)$ is defined in (2.10). Note that the top left corner of $\boldsymbol{C}$ is equal to $\boldsymbol{G}$. We can construct a nonnegative definite matrix $\boldsymbol{C}$ by choosing an appropriate value of M, see Wood and Chan (1994). $\boldsymbol{C}$ can be decomposed as $\boldsymbol{C} = \boldsymbol{Q}\Lambda\boldsymbol{Q}^*$, where $\boldsymbol{Q}$ is defined as an unitary matrix $\boldsymbol{Q} = (q_{jk})_{j,k=0}^{M-1}$:

$$q_{jk} = \frac{1}{\sqrt{M}} exp\left\{-2\pi i \frac{jk}{M}\right\},$$

with $i = \sqrt{-1}$; $\boldsymbol{Q}^*$ is the conjugate transpose of $\boldsymbol{Q}$. $\Lambda = diag(\lambda_0, \lambda_1, \cdots, \lambda_{M-1})$, $\lambda_k = \sum_{j=0}^{M-1} c_j exp\left\{-2\pi i \frac{jk}{M}\right\}$, it is the diagonal matrix of eigenvectors of C. To generate random

vector $\boldsymbol{y}$ with mean 0 and covariance C, given $\boldsymbol{Q}$ is unitary, we can define $\boldsymbol{y} = \boldsymbol{Q}\Lambda^{1/2}\boldsymbol{Q}^*\boldsymbol{z}$ with $\boldsymbol{z} \sim N(0, I)$, and $(y_0, \ldots, y_N - 1) \sim N(0, \boldsymbol{G})$. More details can be found in Wood and Chan (1994); Jean-Francois (2007). From (2.12), the covariances of discrete fractional Gaussian noise are positive for $H > 0.5$, negative for $H < 0.5$ and zero for $H = 0.5$. Figure 2.2 gives simulated dfGn for four different values of the Hurst parameter. For $H = 0.25$, the negative correlation sample has high variability, for $H = 0.95$, there are clearly periods and less variability.

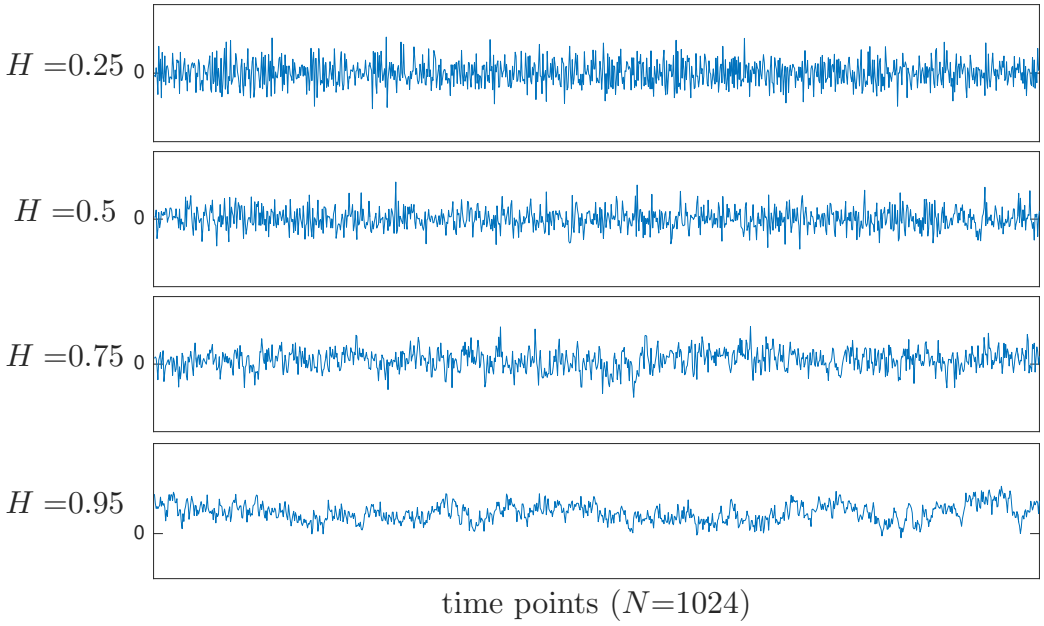

Figure 2.2: Samples of discrete fractional Gaussian noise (dfGn) with $N = 1024$ for $H = 0.25$, $H = 0.5$, $H = 0.75$, $H = 0.95$.

2.5.2 Simulation of fMRI Data with Long Memory Errors

To have a better understanding of fMRI experiment and fMRI time series, we simulate a simple fMRI design with random stimulus onsets. Specifically, we simulate a random design with one stimulus type, the length of the design is set to 512. In Figure 2.3, the blue sticks in the top plot shows the onset times of the stimuli in second (s), the inter-stimulus interval τ_{ISI} is set to 0.5 s, that is, there are two stimulus onsets per second. The other two plots of

Figure 2.3 present the simulated fMRI signal (middle) and the simulated fMRI signal with long memory errors (bottom). fMRI time series is simulated with the general linear model (Friston *et al.*, 1994), $y(t) = X(t)\beta + \epsilon(t)$, where the error term follows a long memory process (dfGn), with the Hurst parameter $H = 0.75$. We set $\beta = 2$, $X(t)$ is obtained by the convolution of the onset stimuli with an HRF, where HRF is a double Gamma function as in (2.1).

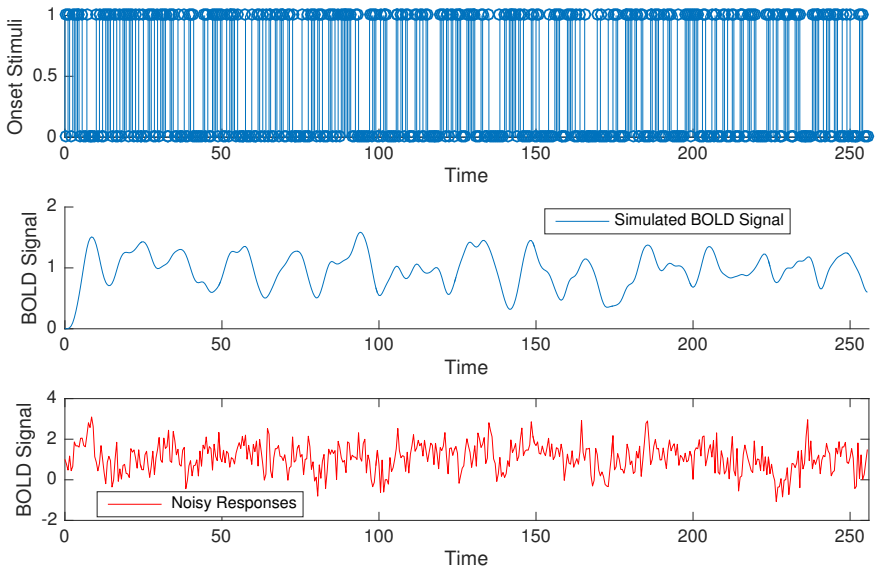

Figure 2.3: Simulation of fMRI data with long memory: simulated fMRI time series with $N = 512$, $\tau_{ISI} = 0.5\,\mathrm{s}$, $Q = 1$. Top plot shows the random generated onset times for each stimulus (time (s)), middle plot shows the convolved signals, bottom plot shows the response signal with added long memory errors with $H = 0.75$.

2.5.3 Hurst Parameter and Estimation

Hurst parameter H provides a measurement of a random process with an underlying trend has some degree of autocorrelation. For $H = 0.5$, the increments of the process are uncorrelated white noise. There exist plenty of methods to measure the Hurst parameter described in various literatures. A good guide about measuring the Hurst parameter can

be found in Clegg (2006). The methods described below are some well-known techniques which have been used to estimate Hurst parameter for some time.

2.5.3.1 R/S Statistics (Rescaled Range Method)

R/S statistics is the original method that Hurst (1951) proposed to measure the long memory, it is used by Mandelbrot and Wallis (1969); Mandelbrot and Taqqu (1979).

Let $x_1, \ldots, x_N$ be a time series with partial sum $Y(n) = \sum_{j=1}^{n} x_j$, $n = 1, \ldots, N$ and $Q_0 = 0$, next, let $S_{k,n} = Y(k) - \frac{k}{n}Y(n)$, $k = 0, 1, \ldots, n$ be the adjusted partial sums. The adjusted range is defined as

$$R_n = \max_{0 \le k \le n} S_{k,n} - \min_{0 \le k \le n} S_{k,n}.$$

In order to study the properties that are independent of the scale, R_n is standardized by its standard deviation:

$$S_n = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(x_j - \bar{x})^2},$$

where $\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_j$ is the sample mean. For fGn and ARFIMA processes, the observations appeared to be well represented by the relation:

$$[R/S] = \mathbb{E}(R_n/S_n) = Cn^H \text{ as } n \to \infty,$$

it is called the rescaled adjusted range or R/S-statistic. Thus, the Hurst parameter can be estimated by

$$\log(R/S) = \log(C) + H(\log(n)).$$

We thus can estimate the Hurst parameter H by least square method.

2.5.3.2 Aggregated Variance Method

Aggregated variance method is described in Beran (1994), it aggregates the time series over blocks of size m, and calculates the variance of the aggregated dataset. For example, for a

fBm process $\{B^H(t), t \geq 0\}$, i.e., for dfGn, $X(n) = B^H(n+1) - B^H(n)$, the variance of the increment process is denoted by σ_X^2. Now, let $s < t$, and $t - s = k$ we have

$$\begin{aligned} \mathrm{Var}(B^H(t) - B^H(s)) &= \mathbb{E}[(B^H(t) - B^H(s))^2] - \mathbb{E}[B^H(t) - B^H(s)]^2 \\ &= \mathbb{E}[B^H(t-s)] \\ &= |k|^{2H} \sigma_X^2. \end{aligned}$$

The m-averaged process $X^{(m)} = (X^{(m)}(1), X^{(m)}(2), \dots)$ of a given series is

$$X^{(m)}(i) = \frac{1}{m} \sum_{n=(i-1)m+1}^{im} X(n),$$

where $i = 1, 2, \dots, N/m$, m and i are positive integers. The variance of this aggregated process is

$$\mathrm{Var}(X^{(m)}) = \frac{1}{N/m} \sum_{i=1}^{N/m} (X^{(m)}(i) - \bar{X})^2.$$

For large $\frac{N}{m}$ and m, the variance of the sample mean $\mathrm{Var}(X^{(m)}) \propto m^{2H-2}$. Thus, the log-log plot of sample variance have a slope of $2H - 2$.

We use MATLAB functions to estimate the Hurst exponents with both R/S statistics method and aggregated variance method. The complete code can be found in Appendix. We generate dfGn with $H = 0.75$ of length 2^x, where $x = 8, \dots, 24$, and estimate the Hurst parameter with both methods. The results are shown in Figure 2.4. As mentioned in Clegg (2006), R/S statistics usually underestimates the Hurst parameter. These methods usually perform better with the simulated data than real data.

2.6 Wavelet Transform

Wavelet, meaning 'little' wave, is a mathematical function that used to represent data (signal or image). It has been proven as a powerful tool for analysis long memory processes, see Wornell and Oppenheim (1992), McCoy and Walden (1996). In fMRI research, different wavelet-based methods have proposed for fMRI data analysis (Ruttimann *et al.*, 1998; Brammer, 1998; Costafreda *et al.*, 2009), a good review of wavelets and fMRI can be found

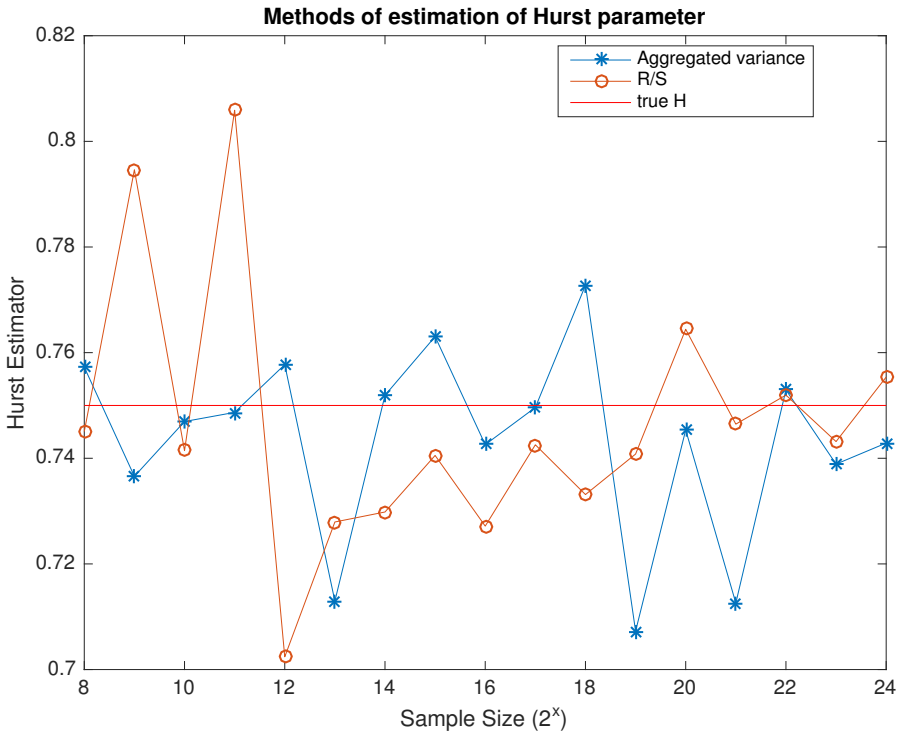

Figure 2.4: Estimators of Hurst parameter with R/S statistics and aggregated variance method, dfGn were generated with length 2^x, where $x = 8, \ldots, 24$, and Hurst parameter $H = 0.75$.

in Bullmore *et al.* (2004). In the present study, following Jeong *et al.* (2013)'s model, we are interested in Discrete Wavelet Transform (DWT) which simplify the variance-covariance matrix of the data with long memory. There are many literature on wavelets and their properties. Here we give a brief introduction of Discrete Wavelet Transform with a simple example on how to construct the wavelet matrix and some useful properties that are relevant to our problem.

2.6.1 Discrete Wavelet Transform

In wavelet analysis, time series signal can be represented as a set of orthogonal wavelet basis functions, so-called 'mother' wavelet and 'father' wavelet (or scaling function). *Haar*

wavelets and *Daubechies* wavelets are two commonly used DWTs. In this section, we describe the procedure for computing Discrete Wavelet Transform.

Let the initial data sequence denoted by $\{y_1, y_2, \ldots, y_N\} = \{y_{0,1}, y_{0,2}, \ldots, y_{0,N}\} = \{y_{m,n}\}$, where $n = 1, \ldots, N$, and $N = 2^J$, it is said to be at level $m = 0$ and scale $2^m = 2^0 = 1$, $m = 0, \ldots, J-1$. The following description explains the standard DWT. At each level or scale, the sequence $\{y_{m,n}\}$ is filtered with a low-pass filter (scaling filter) $\{g_l\}$ and high-pass filter (wavelet filter) $\{h_l\}$, $l = 0, \ldots L-1$ respectively, where L is the length of the filter. That is

$$\{y_{m,n}\} \rightarrow \begin{cases} \text{low-pass} & \text{downsample} & \text{approximation} \\ u_{m+1,i} = \sum_{l=0}^{L-1} g_l y_{m,l+2i-1} & \rightarrow \quad \downarrow 2 \quad \rightarrow & \{u_{m+1,i}\} \\ v_{m+1,i} = \sum_{l=0}^{L-1} h_l y_{m,l+2i-1} & \rightarrow \quad \downarrow 2 \quad \rightarrow & \{v_{m+1,i}\} \\ \text{high-pass} & \text{downsample} & \text{detail} \end{cases} . \quad (2.13)$$

The sequence $\{y_{m,n}\}$ is then 'downsampled' by a factor of 2, as $\{u_{m+1,i}\}$ and $\{v_{m+1,i}\}$ with length $\frac{N}{2^{m+1}}$ respectively, where $i = 1, \ldots, 2^{J-m-1}$. Sequence $\{u_{m+1,i}\}$ is known as 'scaling' coefficients, it provides an approximation of the sequence; $\{v_{m+1,i}\}$ is a sequence of wavelet coefficients, providing detail information of the sequence. The scaling coefficients $\{u_{m+1,i}\}$ then pass to the next level as input to obtain a vector of scaling coefficients and wavelet coefficient by repeating the operation (2.13). The process continues until we obtain one scaling coefficient and one wavelet coefficient at level J, that is $u_{J,1}$ and $v_{J,1}$. Together with the other wavelet coefficients obtained from the previous $J-1$ steps, which are $v_{m+1,i}$, where $m = 1, \ldots, J-2$ and i is as defined earlier. Thus, after J steps,

$$y : \{y_1, y_2, \ldots, y_N\} \rightarrow \{u_{J,1}, v_{J,1}, v_{J-1,1}, v_{J-1,2}, \ldots, v_{1,1}, \ldots, v_{1,2^{J-1}}\} = z. \quad (2.14)$$

Example 2 describe the procedure of applying Harr wavelet transform to the data up to the maximum scale.

Example 2 *Let $N = 2^J = 4$, thus $J = 2$. We use 'Haar wavelets' to illustrate the process. In particular, the scaling function (g_l) and wavelet function (h_l) coefficients are defined as:*

$$\begin{cases} \textit{Scaling function coefficients:} \quad h_0 = 0.5; \quad h_1 = 0.5 \\ \textit{Wavelet function coefficients:} \quad g_0 = 0.5; \quad g_1 = -0.5 \end{cases} . \tag{2.15}$$

In matrix form:

$$\begin{bmatrix} h_0 & h_1 & 0 & 0 & \cdots \\ 0 & 0 & h_0 & h_1 & \cdots \\ g_0 & g_1 & 0 & 0 & \cdots \\ 0 & 0 & g_0 & g_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} .$$

Then the discrete wavelet transform (DWT) can be described as follows:

Let the initial data vector be $[y_{0,n}] = [y_{0,1}, y_{0,2}, y_{0,3}, y_{0,4}]'$. *The first step of the forward Haar transform for* $[y_{0,t}]$ *is*

$$w_0[y_{0,n}] = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} y_{0,1} \\ y_{0,2} \\ y_{0,3} \\ y_{0,4} \end{bmatrix} = \begin{bmatrix} (y_{0,1} + y_{0,2})/2 \\ (y_{0,3} + y_{0,4})/2 \\ \hdashline (y_{0,1} - y_{0,2})/2 \\ (y_{0,3} - y_{0,4})/2 \end{bmatrix} = \begin{bmatrix} u_{1,1} \\ u_{1,2} \\ \hdashline v_{1,1} \\ v_{1,2} \end{bmatrix} : [y_{1,n}];$$

thus, at level $m = 0$, *we obtain* $\{y_{1,i}\}$ *and* $\{d_{1,i}\}$ *with each length* $\frac{J}{2^{m+1}} = 2$. *Next step, we multiple the* $y_{1,n}$ *values by a* 4×4 *transform matrix:*

$$w_1[y_{1,n}] = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{1,1} \\ u_{1,2} \\ v_{1,1} \\ v_{1,2} \end{bmatrix} = \begin{bmatrix} (u_{1,1} + u_{1,2})/2 \\ \hdashline (u_{1,1} - u_{1,2})/2 \\ \hdashline v_{1,1} \\ \hdashline v_{1,2} \end{bmatrix} = \begin{bmatrix} u_{2,1} \\ v_{2,1} \\ v_{1,1} \\ v_{1,2} \end{bmatrix} .$$

This complete the wavelet process by generating single average $u_{2,1}$ *and detail* $v_{2,1}$ *at level* $J = 2$.

□

We can define wavelet transform matrix

$$W = w_m w_{m-1} \dots w_0, \tag{2.16}$$

where w_m is wavelet transform matrix at level m. where $u_{J,1}$ is the scaling coefficient, and the rest are called wavelet coefficients.

2.6.2 Discrete Wavelet Transform and Long Memory Process

The highly correlated long memory processes lead a dense variance-covariance matrix of the data. As whitening filters, discrete wavelet transforms allow to simplify the variance-covariance matrix. In this section, we provide some details about covariance structure after applying wavelet transform to the data with long memory processes.

Discrete wavelet transform can represent in matrix form as

$$\boldsymbol{z} = W\boldsymbol{y}, \tag{2.17}$$

where vector $\boldsymbol{z}, \boldsymbol{y}$ are given in (2.14), and wavelet transform matrix W was defined in (2.16). So with a given covariance matrix of $\boldsymbol{y}$, Σ_y, the covariance matrix of $\boldsymbol{z}$, can be written as:

$$\Sigma_z = W\Sigma_y W'. \tag{2.18}$$

As reported in McCoy and Walden (1996), the DWT coefficients of long memory processes (dfGn) can approximately uncorrelated. So the covariance matrix in (2.18) can be written as a diagonal matrix

$$\begin{bmatrix} S_{p+1} & & & & & & \\ & S_p & & & & & \\ & & \begin{pmatrix} S_{p-1} & \\ & S_{p-1} \end{pmatrix} & & & & \\ & & & \ddots & & & \\ & & & & \begin{pmatrix} S_1 & & \\ & \ddots & \\ & & S_1 \end{pmatrix} & \end{bmatrix}. \tag{2.19}$$

Here S_{p+1} and S_p, where $p = 1, \ldots, J$ are the variance of the scaling coefficient and wavelet coefficients as in (2.14):

$$S_p = \mathrm{Var}\{v_{m+1,i}\} \text{ for } p = 1, \ldots, J; \quad S_{p+1} = \mathrm{Var}\{u_{J,1}\},$$

where m, i are same as in (2.14).

2.6.2.1 Variance of Wavelet Coefficients

Recall that in Section 2.5, the power spectral density function of discrete fractional Gaussian noise is $s(\omega) = (2\sin(\frac{1}{2}\omega))^{-2d^*}$, that is,

$$S(f) = (2\sin \pi f)^{-2d^*},$$

where $d^* = 1 - H$ and $0 < d^* < 0.5$, the signal is stationary and has long memory. We have $S(f) \sim f^{-2d^*}$ as $f \to 0$, since $\sin^{-2d^*}(\pi f) \approx \pi f^{-2d^*}$ when $f \to 0$.

At level m, the bandpass variance B_m in the frequency octaves $\left[-2^{-m}, -2^{-m-1}\right] \cup \left[2^{-m-1}, 2^{-m}\right]$ can be written as

$$\begin{aligned}
B_m &= 2\int_{2^{-m-1}}^{2^{-m}} (2\sin(\pi f))^{-2d^*} df \\
&= 2 \times 2^{-2d^*} \int_{2^{-m-1}}^{2^{-m}} \sin^{-2d^*}(\pi f) df \\
&\approx 2 \times 2^{-2d^*} \int_{2^{-m-1}}^{2^{-m}} (\pi f)^{-2d^*} df \\
&= 2 \times (2\pi)^{-2d^*} \int_{2^{-m-1}}^{2^{-m}} f^{-2d^*} df \\
&= 2 \times (2\pi)^{-2d^*} 2^{-m(1-2d^*)} \frac{1 - 2^{2d^*-1}}{1 - 2d^*}.
\end{aligned}$$

Suppose over each such frequency octave, we wish to replace by a constant power spectrum, with value S_m, then the bandpass variance is

$$\int_{2^{-m-1}}^{2^{-m}} S_m df = S_m 2^{-m-1}.$$

In order to get equal bandpass variance $2 \times S_m 2^{-m-1} = B_m$, thus,

$$\begin{aligned} S_m &= 2^m B_m \\ &= 2 \times (2\pi)^{-2d^*} 2^{2md^*} \frac{1 - 2^{2d^*-1}}{1 - 2d^*} \\ &= 2(2\pi)^{-2d^*} \frac{1 - 2^{2d^*-1}}{1 - 2d^*} \times 2^{2md^*} \\ &= C 2^{2md^*}, \end{aligned} \tag{2.20}$$

where C is a constant.

Follow Jeong *et al.* (2013) approach, we omit the variance of scaling coefficient S_{p+1} and will not discuss the details here, some information can be found in McCoy and Walden (1996).

2.7 Existing Algorithms for Obtaining fMRI Designs

In this section, we present a brief review on some popular algorithms or methods for finding fMRI designs that optimize a given optimality criterion. To search for a good design over the large space of fMRI designs, Wager and Nichols (2003) proposed to use a genetic algorithm (GA). Later, based on the knowledge of the performance of some fMRI designs and the features of genetic algorithms, Kao *et al.* (2009) proposed an improved algorithm that not only is fast, but also can effectively obtain high-quality fMRI designs. Following these works, genetic algorithms have been applied in obtaining fMRI designs for many real experiments, see Maus *et al.* (2010); Cordes *et al.* (2012); Delzell *et al.* (2012) and Zhou (2014) among others.

Besides GAs, some other computationally efficient algorithms have been proposed in recent years. For example, Kao and Mittelmann (2011) proposed a fast algorithm by combining hill climbing algorithm with cyclic permutation method for constructing high-quality fMRI designs. Saleh *et al.* (2017) proposed another algorithm for optimizing D-optimal fMRI designs; their algorithm is built upon the coordinate-exchange algorithm of Meyer and Nachtsheim (1995) and hill climbing algorithm. Both algorithms can outperform GAs

under certain experimental settings.

In this study, we adapt Kao *et al.* (2009)'s genetic algorithm for our first research topic. Here, we give a brief description of our adapted GA in Algorithm 1:

Algorithm 1 Adapted genetic algorithm

Step 1. Generate 20 designs as parents, which include block designs of various block sizes, random designs, m-sequence based designs and mixed designs. Obtain their design efficiencies.

Step 2. With probability proportional to the value of design efficiency, randomly draw 10 pairs of distinct designs with replacement from the 20 parent designs.

Step 3. For each design pair, randomly select a cut-point and exchange the corresponding fractions before the cut-point of the two designs to obtain a pair of offspring designs.

Step 4. Randomly select a portion (1%) of elements of the offspring designs, and randomly perturb these elements by replacing them with integers randomly generated from the discrete uniform distribution over $0, 1, 2, \ldots, Q$.

Step 5. Randomly generate 4 immigrant designs from mixed designs.

Step 6. Obtain the design efficiencies for the offspring and immigrant designs.

Step 7. In the current pool of the parent, offspring and immigrant designs, select the best 20 designs to form the parents of the next generation.

Step 8. Repeat Steps 2-7 until no significant improvement in the design efficiency can be expected; i.e. use the second stopping rule of Kao (2009). Keep track of the best design over generations.

Note that a mixed design in Steps 1 and 5 is formed by a fraction of a 16 s-on-16 s-off block design followed by a fraction of an m-sequence (or a random design if the m-sequence

is unavailable). To obtain such a design, we randomly select a cut-point and replace all the elements of the block design after the cut-point by the corresponding elements of an m-sequence (or a random design).

We then shift our focus to the development of an efficient algorithm for a wavelet-based model as the second topic. Inspired by Saleh *et al.* (2017), we proposed an algorithm by combining a simulated annealing algorithm with an exchange algorithm. We now give a brief introduction about a simulated annealing algorithm in the next subsection.

2.7.1 Simulated Annealing Algorithm

Simulated annealing is a popular algorithm in stochastic optimization, it is proposed by Kirkpatrick *et al.* (1983), Černỳ (1985) independently. It has been applied to many areas since then, such as very-large-scale integrated (VLSI) circuit computer chips, communication technology, network design, a review of the algorithm and its application can be found in Tan and Raghavan (2008).

Simulated annealing algorithms mimic the process of heating a material to melt and then carefully control the cooling scheme to reduce its defects and increase the size of crystals. Starting at a high temperature (melted state), the physical process of annealing reducing the temperature gradually to reach its minimum energy state (thermal equilibrium) at each temperature. Mathematically, Let $U(x)$ be the objective function that we would like to optimize (maximize or minimize). Let the starting temperature T_0 be large, the cooling scheme of simulated annealing is presented in Algorithm 2, where i is the initial solution of the problem, j is a candidate solution. At each temperature T, the solid reaches to thermal equilibrium which is determined by the Boltzmann distribution:

$$Prob(E) = \frac{1}{Z(T)} exp(-E/(k_B T)), \tag{2.21}$$

where E is the energy of the current state, k_B is the Boltzmann constant, Z is the partition function. As T decreases, the Boltzmann distribution concentrates on the state with lowest energy.

Algorithm 2 Pseudocode for simulated annealing algorithm

```
1  Initialization:
2  T_0, R (Reduction factor), i
3  T = T_0;
4  while stop criteria not met do
5      pick a neighbor j
6      Calculate ΔE = E(j) − E(i)
7      if ΔE > 0 then
8          i = j                                  ▷ accept the improvement
9      else
10         p = exp(−ΔE/T)
11         if p > rand(0, 1) then
12             i = j                              ▷ accept the worsening
13         end if
14     end if
15     T = R * T
16 end while
```

In a simulated annealing algorithm, the constant k_B is normally omitted. At a high T, the algorithm performs a coarse search over the entire space and identifies a 'subspace' where a good solution might be found. As T becomes increasingly small, the algorithm focuses on the 'subspace', aiming at the best solution in that subspace. The probability of accepting a worse solution, which is usually set to $exp(-\frac{\Delta E}{T})$ for minimization problem. The temperature T is started with a relatively high value such that the probability of acceptance is high at the beginning of the search to avoid being trapped in a local optimum. As T decreases, the probability becomes smaller until it reaches to zero. We consider an algorithm of this type in our second topic to be presented in Chapter 4.

Chapter 3

DESIGNS OF FMRI EXPERIMENTS WITH AN UNCERTAIN DESIGN MATRIX

3.1 A Motivating Example

In most of the traditional fMRI experimental settings, the design matrix such as the $\boldsymbol{X}$ matrix in (2.2) and (2.3) can be determined at the design stage. But, this might not be the case for some modern fMRI experiments. Here, our focus is on a case where the design matrix $\boldsymbol{X}$ in the model is uncertain before data collection. A study of this sort is reported by Cordes *et al.* (2012). Specifically, Cordes *et al.* (2012) reported an fMRI experiment for studying brain activity evoked by the mental stimuli (pictures) presented to her/him, and the subject's reaction to each of these pictures. In their pilot study, the subjects were asked to study a list of pictures that presented asymmetrically on a vertical axis before an fMRI experiment. During the experiment, some of the pictures were presented with the same or opposite left/right orientation, interlaced with new pictures. Consequently, there are three stimulus types, namely (1) studied pictures with the 'same' orientation; (2) studied pictures with the 'different' orientation; and (3) 'new' pictures. The subjects were asked to select an answer among 'same', 'different' and 'new' for each picture. During the experiments, the subject's answers are recorded along with the BOLD signals of the subject's brain collected by the fMRI scanner. The main objective of Cordes *et al.* (2012)'s experiments is to study the brain activity with respect to the stimulus-response pairs. Since the subject's reactions are uncertain at the design stage (before the experiment starts), it is unlikely to have an accurate evaluation of the quality of designs. More precisely, the uncertainty of the design matrix $\boldsymbol{X}$ will give an unknown information matrix $\boldsymbol{M}$. The value of the optimality criterion $\phi(\boldsymbol{M})$ for evaluating the quality of designs is thus unavailable. Note that these terms will depend on the subject's probabilistic behavior in selecting the answers during the experiment and are thus unknown at the design stage (see also Section 3.2).

This uncertainty makes it very challenging to select good designs for this type of modern experiments.

To tackle this design issue, Cordes *et al.* (2012) first approximate the probabilities of the subject's answers to each type of pictures. For each candidate design $\boldsymbol{d}$, they then simulate, say, 100 realizations of the subject's answers to obtain 100 realizations of the design matrix. This results in 100 realizations of the ϕ-values for $\boldsymbol{d}$. A summary statistic such as the median or mean of these 100 ϕ-values is obtained to evaluate the goodness of $\boldsymbol{d}$. Conceptually, this is similar to use $\mathbb{E}[\phi(\boldsymbol{M})|\boldsymbol{d}]$ to evaluate the goodness of $\boldsymbol{d}$, where the expectation $\mathbb{E}[\cdot|\boldsymbol{d}]$ is taken over the conditional probability of the subject's answer to each given stimulus type. Since a closed form of $\mathbb{E}[\phi(\boldsymbol{M})|\boldsymbol{d}]$ is in general unavailable, a Monte Carlo simulation is considered to approximate this criterion. Unfortunately, the Monte Carlo simulation is time consuming, and it needs to be repeated for every candidate design. The procedure thus requires much computational effort as reported in Cordes *et al.* (2012). An efficient approach is called for.

3.2 Methodology

3.2.1 The General Linear Models

Following Cordes *et al.* (2012), extensions of Models (2.2) and (2.3) are considered for modeling the data collected from experiments having the above mentioned settings. For simplicity, we assume that the subject can have R possible responses for every stimulus. We now describe the statistical models for the estimation and detection problems.

3.2.1.1 A Linear Model for Estimation

For estimating the HRFs, the model that we consider is

$$\boldsymbol{y} = \sum_{k=1}^{K}\sum_{q=1}^{Q}\sum_{r=1}^{R} \boldsymbol{x}_{r,q,k} h_{r,q,k} + \boldsymbol{S}\boldsymbol{\gamma} + \boldsymbol{\epsilon} = \boldsymbol{X}\boldsymbol{\zeta} + \boldsymbol{S}\boldsymbol{\gamma} + \boldsymbol{\epsilon}. \tag{3.1}$$

This is a extension of Model (2.2). In particular, the $T \times RQK$ design matrix $\boldsymbol{X}$ consists of column vectors $\boldsymbol{x}_{r,q,k}$'s with all elements either 0 or 1, and $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_K]$, $\boldsymbol{X}_k = [\boldsymbol{X}_{1,k}, \ldots, \boldsymbol{X}_{Q,k}]$, $\boldsymbol{X}_{q,k} = [\boldsymbol{x}_{1,q,k}, \ldots, \boldsymbol{x}_{R,q,k}]$. The t^{th} element of $\boldsymbol{x}_{r,q,k}$ is $((\boldsymbol{x}_{r,q,k}))_t = 1$ if $h_{r,q,k}$ contributes to y_t. Here, each unknown parameter $h_{r,q,k}$ represents the HRF of the event that the subject selects the r^{th} answer to a stimulus of the q^{th} type evaluated at the stimulus onset time $h_{r,q,1}$ and the subsequent, regularly spaced $(K-1)$ time points $[h_{r,q,2}, \ldots, h_{r,q,K}]$. $\boldsymbol{\zeta}$ is a vector of $h_{r,q,k}$ with $\boldsymbol{\zeta} = [\boldsymbol{\zeta}_1', \ldots, \boldsymbol{\zeta}_K']'$, $\boldsymbol{\zeta}_k = [\boldsymbol{\zeta}_{1,k}', \ldots, \boldsymbol{\zeta}_{Q,k}']'$ and $\boldsymbol{\zeta}_{q,k} = [h_{1,q,k}, \ldots, h_{R,q,k}]'$. The remaining terms are the same as in Model (2.2). With Model (3.1), parametric functions of interest would be $\boldsymbol{C}_h\boldsymbol{\zeta}$ for a coefficient matrix $\boldsymbol{C}_h$. For illustration purposes, we will consider $\boldsymbol{C}_h = \boldsymbol{I}_{RQK}$, the estimation of the heights in $\boldsymbol{\zeta}$. Our proposed method can be easily adopted to accommodate other $\boldsymbol{C}_h$.

3.2.1.2 A Linear Model for Detection

The model that we consider for detection is

$$\boldsymbol{y} = \sum_{k=1}^{K}\sum_{q=1}^{Q}\sum_{r=1}^{R} \boldsymbol{x}_{r,q,k} h_k^* \theta_{r,q} + \boldsymbol{S}\boldsymbol{\gamma} + \boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{H}\boldsymbol{\theta} + \boldsymbol{S}\boldsymbol{\gamma} + \boldsymbol{\eta}. \tag{3.2}$$

Here, $\boldsymbol{H} = \boldsymbol{h}^* \otimes \boldsymbol{I}_{RQ}$, the $K \times 1$ vector $\boldsymbol{h}^*$ is the assumed basis function of HRF, which may be determined by function (2.1). The corresponding coefficient $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_Q]'$; $\boldsymbol{\theta}_q = [\theta_{1,q}, \ldots, \theta_{R,q}]'$ with $\theta_{r,q}$ represents the response amplitude for q^{th}-type stimulus with r^{th} answer. The vector $\boldsymbol{\eta}$ is the correlated error terms that are assumed to follow an AR(1) process. All remaining terms of Model (3.2) are as in Model (3.1).

3.2.2 Design Criterion

We would like to find a design that gives the most precise generalized least squares estimator (GLSE) of the parametric function of interest. The goodness of a design will be evaluated by a functional ϕ of the information matrix $\boldsymbol{M}$ of the parametric function of interest. We defer the details of the information matrix to the next subsection. Here we describe the design selection criteria that we consider.

We apply two commonly used criteria in fMRI design studies, namely the A- and D-optimality criteria (Dale, 1999; Wager and Nichols, 2003; Liu and Frank, 2004; Kao *et al.*, 2009; Maus *et al.*, 2010). Extending our method to other optimality criteria should be straightforward. These two criteria are defined as follows:

$$\phi(\boldsymbol{M}) = \begin{cases} \mathcal{R}/\mathrm{tr}(\boldsymbol{M}^{-1}), \text{ for } A\text{-optimality;} \\ \det(\boldsymbol{M})^{1/\mathcal{R}}, \text{ for } D\text{-optimality.} \end{cases} \tag{3.3}$$

where $\boldsymbol{M}$ is nonsingular. We set $\phi(\boldsymbol{M}) = 0$ when $\boldsymbol{M}$ is singular, $\mathcal{R}$ is set to RQK for Model (3.1) and RQ for Model (3.2), which corresponds to the number of the parameters of interest.

We now provide some additional discussions on the information matrix $\boldsymbol{M}$.

3.2.3 Information Matrices and the Expectations

The information matrices $\boldsymbol{M}$ are:

$$\begin{cases} \boldsymbol{M}(\boldsymbol{\zeta}) = \boldsymbol{X}'\boldsymbol{V}'(\boldsymbol{I}_T - \mathcal{P}_{\boldsymbol{VS}})\boldsymbol{VX}, \text{ for estimation Model (3.1);} \\ \boldsymbol{M}(\boldsymbol{\theta}) = \boldsymbol{H}'\boldsymbol{X}'\boldsymbol{V}'(\boldsymbol{I}_T - \mathcal{P}_{\boldsymbol{VS}})\boldsymbol{VXH}, \text{ for detection Model (3.2).} \end{cases} \tag{3.4}$$

$\mathcal{P}_{\boldsymbol{VS}} = \boldsymbol{VS}(\boldsymbol{S}'\boldsymbol{V}'\boldsymbol{VS})^{-}\boldsymbol{S}'\boldsymbol{V}'$ is the orthogonal projection on the vector space spanned by the column vectors of $\boldsymbol{VS}$, with $\boldsymbol{B}^{-}$ denoting a generalized inverse of a matrix $\boldsymbol{B}$. $\boldsymbol{V}$ is a whitening matrix so that $\mathrm{Cov}(\boldsymbol{V\epsilon}) = \sigma_1^2\boldsymbol{I}_T$, $\mathrm{Cov}(\boldsymbol{V\eta}) = \sigma_2^2\boldsymbol{I}_T$ for Models (3.1) and (3.2) respectively. All the remaining terms are as described in Models (3.1) and (3.2). The information matrix $\boldsymbol{M}$ depends on the selected design $\boldsymbol{d}$. But, in contrast to the traditional settings, this $\boldsymbol{M}$ will also depend on the subject's answers $\boldsymbol{\alpha}$. In particular, $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_N\}$ with $\alpha_n \in \{0, 1, 2, \ldots, R\}$, and $n = 1, \ldots, N$. Here, α_n represents the subject's answer to d_n. We set $\alpha_n = 0$ when $d_n = 0$; i.e. when there is no stimulus presentation, the subject does not respond. In addition, we set α_n can be $1, 2, ...,$ or R when $d_n \neq 0$. Since the subject's responses are not available at the design stage, the design matrix $\boldsymbol{X}$ and the information matrices of equation (3.4) cannot be determined at that stage. Consequently, $\phi(\boldsymbol{M})$ is unavailable at the design stage.

3.2.4 The Proposed Approach

As previously mentioned, the method used by Cordes *et al.* (2012) for obtaining optimal designs for the current setting is very time consuming. Zhou (2014) proposed a very efficient alternative method. But the application of her method is only restricted to the estimation problems with $\tau_{ISI} = \tau_{TR}$. Further developments are needed for other practical situations. We now briefly describe the method of Zhou (2014) and provide useful extensions to other cases, including both estimation and detection problems with or without $\tau_{ISI} = \tau_{TR}$.

The key idea of Zhou (2014) is by utilizing the ϕ-value of the expected information matrix, namely $\phi(\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}])$, instead of $\mathbb{E}[\phi(\boldsymbol{M})|\boldsymbol{d}]$; note that the latter criterion is essentially the one considered by Cordes *et al.* (2012). The expectation is taken over the conditional probability of the subject's answer α_n's to the presented stimulus given the design $\boldsymbol{d}$. Following Cordes *et al.* (2012), we consider α_n's as independent random variables, and the expectation of the information matrix $\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]$ is taken over $p(\alpha_n = r|d_n = q) = p(r|q)$, which can be viewed as the prior conditional probability of $\boldsymbol{\alpha}$ given design $\boldsymbol{d}$. In other words, $p(r|q)$ is the probability that the subject selects the r^{th} answer for a q^{th}-type stimulus. The prior probability can be approximated from, say, a pilot study. We then obtain a design that maximizes $\phi_1 \equiv \phi(\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}])$ for given $p(r|q)$. Here, ϕ is a specified optimality criterion, in our case, it refers to one of the criteria in equation (4.7). Another possibility is to consider a criterion similar to that of Cordes *et al.* (2012), namely $\phi_2 \equiv \mathbb{E}[\phi(\boldsymbol{M})|\boldsymbol{d}]$.

A major advantage of using ϕ_1 is that a closed form of $\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]$ can be derived. The main idea for deriving this closed form expression is by making use of the formula for the expectation of a quadratic form of a random variable. We now present the expressions of our analytical results in Theorem 3.2.1 and Corollary 3.2.2. Note that, when $\tau_{ISI} = \tau_{TR}$, these results reduce to those of Zhou (2014). Our results are thus more general than Zhou (2014).

Theorem 3.2.1 *For Model* (3.1)*, define the partitioned matrix*

$$((((\mathbb{E}[\boldsymbol{X}'_{p,i}\boldsymbol{A}\boldsymbol{X}_{q,j}|\boldsymbol{d}]))_{p,q=1,...,Q}))_{i,j=1,...,K} = \mathbb{E}[\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}|\boldsymbol{d}] = \mathbb{E}[\boldsymbol{M}|\boldsymbol{d}];$$

we have $\mathbb{E}[\boldsymbol{X}'_{p,i}\boldsymbol{A}\boldsymbol{X}_{q,j}] = \mathrm{T}_q \times [diag(\boldsymbol{\pi}_q) - \boldsymbol{\pi}_q\boldsymbol{\pi}'_q] + \boldsymbol{\pi}_q\boldsymbol{\pi}'_q\boldsymbol{U}_{q,q}$ *where* $q = 1, \ldots, Q$*; and* $\mathbb{E}[\boldsymbol{X}'_{p,i}\boldsymbol{A}\boldsymbol{X}_{q,j}] = \boldsymbol{\pi}_q\boldsymbol{\pi}'_q\boldsymbol{U}_{p,q}$ *where* $p \neq q$*;* $p, q = 1, \ldots, Q$*.*

In Theorem 3.2.1, $\boldsymbol{A} = \boldsymbol{V}'(\boldsymbol{I}_T - \mathcal{P}_{\boldsymbol{VS}})\boldsymbol{V}$; $diag(\boldsymbol{\pi}_q)$ is a diagonal matrix whose diagonal elements are those in $\boldsymbol{\pi}_q = [p(1|q), \ldots, p(R|q)]'$; $T_q = trace\{\boldsymbol{A}_{i,j}diag(\boldsymbol{\omega}_q)\}$, and $\boldsymbol{U}_{p,q} = \boldsymbol{\omega}'_p\boldsymbol{A}_{i,j}\boldsymbol{\omega}_q$ where $\boldsymbol{A}_{i,j} = \boldsymbol{D}'_i\boldsymbol{A}\boldsymbol{D}_j$ and $\boldsymbol{D}_k = \boldsymbol{I}_T \otimes [1, \boldsymbol{0}'_{\tilde{\tau}_{TR}-1}]\boldsymbol{U}\tilde{\boldsymbol{L}}^{k-1}$, $\tilde{\boldsymbol{L}}$ is a lower shifting matrix, more details for $\boldsymbol{U}$ and $\tilde{\boldsymbol{L}}$ can be found in (3.6) and (3.7); $\boldsymbol{\omega}_q$ is as in Example 1.

For detection of the brain activation, the expectation of the information matrix can be obtained by replacing the design matrix $\boldsymbol{X}$ with $\boldsymbol{X}\boldsymbol{H}$ where $\boldsymbol{H}$ is defined in Model (3.2).

Corollary 3.2.2 *We have* $\mathbb{E}[\boldsymbol{H}'\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}\boldsymbol{H}|\boldsymbol{d}] = \boldsymbol{H}'\,\mathbb{E}[\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}|\boldsymbol{d}]\boldsymbol{H}$ *for Model* (3.2)*, where* $\mathbb{E}[\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}|\boldsymbol{d}]$ *can be obtained from Theorem 3.2.1.*

We now explain how these two results were obtained in the next subsection.

3.2.5 Evaluating the Expected Information Matrix

3.2.5.1 Expected Information Matrix for Estimation

Without loss of generality, we assume that $\boldsymbol{X}$ in Model (3.1) have the form of $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_K]$, where for $k = 1, ..., K$, and for $q = 1, ..., Q$, $\boldsymbol{X}_k = [\boldsymbol{X}_{1,k}, \ldots, \boldsymbol{X}_{Q,k}]$ with dimension $N \times RQ$, and $\boldsymbol{X}_{q,k} = (\boldsymbol{x}_{1,q,k}, \ldots, \boldsymbol{x}_{R,q,k})$. The vector $\boldsymbol{x}_{r,q,k}$ is defined as in Models (3.1) and (3.2). Since most optimality criteria ϕ are invariant to a simultaneous permutation of rows and columns, a rearrangement of the columns of $\boldsymbol{X}$ will not change the value of $\phi(\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}])$. We have also derived a closed form for the expected information matrix by setting $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_Q]$, where $\boldsymbol{X}_q$ contains all the RK vectors $\boldsymbol{x}_{r,q,k}$ of the same q, for $r = 1, \ldots, R$ and $k = 1, \ldots, K$. We omit this latter result because, compared with the former arrangement of $\boldsymbol{X}$, it tends to take more CPU time when the closed form

derived by the latter choice of $\boldsymbol{X}$ is considered. Results for cases $\tau_{ISI} = \tau_{TR}$ can be found in Zhou (2014), here we extend to the cases with $\tau_{ISI} \neq \tau_{TR}$.

We now derive the expectation of the information matrix $\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]$. The expectation is taken over $p(r|q)$, the conditional probability when subject selects the r^{th} answer for a stimulus of the q^{th} type. Here, we assume that $p(r|q)$ remains the same throughout the experiment, and the subject's answer only depends on the current stimulus, and is independent of his/her answers to the previous stimuli; we also assume that, for each stimulus, the subject selects one answer from the R possible answers, and if there is no stimulus, the subject does not respond, and $p(r = 0|q = 0) = 1$. Our results can easily be extended to a more general case such as $p(r|0) > 0$ for $r = 1, \ldots, R$, and/or $p(0|q) > 0$ for $q = 1, \ldots, Q$. For convenience, we also use $\boldsymbol{A}$ to represent $\boldsymbol{V}'(\boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{VS}})\boldsymbol{V}$. Consequently, $\boldsymbol{M} = \boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}$. The main idea is then to make use of the formula for the expectation of a quadratic form as presented in Ch.5 of Rencher and Schaalje (2008). We now present some details of our derivations of $\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]$. We note that all the expectations (and covariances) are conditional on the design $\boldsymbol{d}$. For simplicity, we write $\mathbb{E}[\cdot]$ (and $\text{Cov}(\cdot)$) instead of $\mathbb{E}[\cdot|\boldsymbol{d}]$ (and $\text{Cov}(\cdot|\boldsymbol{d})$) in this subsection.

First, we define $\boldsymbol{\delta}_q$ as the 0-1 indicator vector for the q^{th}-type stimulus. Specifically, the n^{th} element $(\boldsymbol{\delta}_q)_n$ of $\boldsymbol{\delta}_q$ is 1 if the corresponding d_n in the design $\boldsymbol{d} = \{d_1, \ldots, d_N\}$ is q^{th} stimulus type; otherwise, $(\boldsymbol{\delta}_q)_n = 0$. Similarly, we define $\boldsymbol{\delta}_{r,q}$ as the 0-1 indicator vector for the subject selects the r^{th} answer to the q^{th}-type stimulus.

Then, for $\tau_{ISI} \neq \tau_{TR}$, we define $\boldsymbol{\omega}_q$ as the onset times of the q^{th}-type stimulus in terms of ΔT, that is,

$$\boldsymbol{\omega}_q = \boldsymbol{\delta}_q \otimes [1, \mathbf{0}'_{\tilde{\tau}_{ISI}-1}]',$$

and we have

$$\boldsymbol{\omega}_{r,q} = \boldsymbol{\delta}_{r,q} \otimes [1, \mathbf{0}'_{\tilde{\tau}_{ISI}-1}]'. \tag{3.5}$$

Note that the dimension of both $\boldsymbol{\omega}_q$ and $\boldsymbol{\omega}_{r,q}$ are $N\tilde{\tau}_{ISI} \times 1$ (Recall that N is the length of the design). $\tilde{\tau}_{ISI}$ and $\tilde{\tau}_{TR}$ are as defined in Section 2.2.

Now, take $\tilde{\tau}_{TR}$ into consideration, we can write

$$\boldsymbol{X}_k = (\boldsymbol{I}_T \otimes [1, \mathbf{0}'_{\tilde{\tau}_{TR}-1}])\boldsymbol{U}\tilde{\boldsymbol{L}}^{k-1}[\boldsymbol{\omega}_{1,1}, \ldots, \boldsymbol{\omega}_{R,1}, \ldots, \boldsymbol{\omega}_{1,Q}, \ldots, \boldsymbol{\omega}_{R,Q}],$$

where

$$\tilde{\boldsymbol{L}} = \begin{pmatrix} \mathbf{0}'_{N\tilde{\tau}_{ISI}-1} & 0 \\ \boldsymbol{I}_{N\tilde{\tau}_{ISI}-1} & \mathbf{0}_{N\tilde{\tau}_{ISI}-1} \end{pmatrix}; \tag{3.6}$$

$$\boldsymbol{U} = \begin{cases} [\mathbf{I}_{T\tilde{\tau}_{TR}}, \mathbf{O}_{T\tilde{\tau}_{TR},(N\tilde{\tau}_{ISI}-T\tilde{\tau}_{TR})}], & T\tilde{\tau}_{TR} < N\tilde{\tau}_{ISI}; \\ \begin{pmatrix} \mathbf{I}_{N\tilde{\tau}_{ISI}} \\ \mathbf{O}_{(T\tilde{\tau}_{TR}-N\tilde{\tau}_{ISI}),N\tilde{\tau}_{ISI},} \end{pmatrix}, & T\tilde{\tau}_{TR} > N\tilde{\tau}_{ISI}; \\ \mathbf{I}_{T\tilde{\tau}_{TR}}, & T\tilde{\tau}_{TR} = N\tilde{\tau}_{ISI}; \end{cases} \tag{3.7}$$

and $\boldsymbol{O}_{a,b}$ is the a-by-b zero matrix; see also Saleh *et al.* (2017). The dimension of $\boldsymbol{X}_k$ is $T \times RQ$ where T is the number of MR scans. Let $\boldsymbol{I}_T \otimes [1, \mathbf{0}'_{\tilde{\tau}_{TR}-1}]\boldsymbol{U}\tilde{\boldsymbol{L}}^{k-1} = \boldsymbol{D}_k$, the dimension of $\boldsymbol{D}_k$ is $T \times N\tilde{\tau}_{ISI}$, so that

$$\boldsymbol{x}_{r,q,k} = \boldsymbol{D}_k\boldsymbol{\omega}_{r,q}. \tag{3.8}$$

Then, for $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_K]$, the expectation of $\boldsymbol{M}$ for Model (3.1) can be written as:

$$\begin{aligned} \mathbb{E}[\boldsymbol{M}] &= \mathbb{E}[\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}] \\ &= \mathbb{E}[(\boldsymbol{X}'_1, ..., \boldsymbol{X}'_K)'\boldsymbol{A}(\boldsymbol{X}_1, ..., \boldsymbol{X}_K)] \\ &= \begin{pmatrix} \mathbb{E}[\boldsymbol{X}'_1\boldsymbol{A}\boldsymbol{X}_1] & \cdots & \mathbb{E}[\boldsymbol{X}'_1\boldsymbol{A}\boldsymbol{X}_K] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[\boldsymbol{X}'_K\boldsymbol{A}\boldsymbol{X}_1] & \cdots & \mathbb{E}[\boldsymbol{X}'_K\boldsymbol{A}\boldsymbol{X}_K] \end{pmatrix} \\ &= ((\mathbb{E}[\boldsymbol{X}'_i\boldsymbol{A}\boldsymbol{X}_j]))_{i,j=1,\ldots,K}. \end{aligned} \tag{3.9}$$

We then write down the $RQ \times RQ$ submatrix $\mathbb{E}[\boldsymbol{X}_i'\boldsymbol{A}\boldsymbol{X}_j]$, that is

$$\begin{aligned}\mathbb{E}[\boldsymbol{X}_i'\boldsymbol{A}\boldsymbol{X}_j] &= \mathbb{E}[(\boldsymbol{X}_{1,i}', ..., \boldsymbol{X}_{Q,i}')'\boldsymbol{A}(\boldsymbol{X}_{1,j}, ..., \boldsymbol{X}_{Q,j})] \\ &= \begin{pmatrix} \mathbb{E}[\boldsymbol{X}_{1,i}'\boldsymbol{A}\boldsymbol{X}_{1,j}] & \cdots & \mathbb{E}[\boldsymbol{X}_{1,i}'\boldsymbol{A}\boldsymbol{X}_{Q,j}] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[\boldsymbol{X}_{Q,i}'\boldsymbol{A}\boldsymbol{X}_{1,j}] & \cdots & \mathbb{E}[\boldsymbol{X}_{Q,i}'\boldsymbol{A}\boldsymbol{X}_{Q,j}] \end{pmatrix} \qquad (3.10) \\ &= ((\mathbb{E}[\boldsymbol{X}_{p,i}\boldsymbol{A}\boldsymbol{X}_{q,j}]))_{p,q=1,...,Q}, \qquad (3.11)\end{aligned}$$

where the dimension of each submatrix $\mathbb{E}[\boldsymbol{X}_{p,i}\boldsymbol{A}\boldsymbol{X}_{q,j}]$ is $R \times R$. In addition, $\mathbb{E}[\boldsymbol{X}_{p,i}\boldsymbol{A}\boldsymbol{X}_{q,j}]$ can be written as:

$$\begin{pmatrix} \boldsymbol{x}_{1,p,i}'\boldsymbol{A}\boldsymbol{x}_{1,q,j} & \boldsymbol{x}_{1,p,i}'\boldsymbol{A}\boldsymbol{x}_{2,q,j} & \cdots & \boldsymbol{x}_{1,p,i}'\boldsymbol{A}\boldsymbol{x}_{R,q,j} \\ \boldsymbol{x}_{2,p,i}'\boldsymbol{A}\boldsymbol{x}_{1,q,j} & \boldsymbol{x}_{2,p,i}'\boldsymbol{A}\boldsymbol{x}_{2,q,j} & \cdots & \boldsymbol{x}_{2,p,i}'\boldsymbol{A}\boldsymbol{x}_{R,q,j} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_{R,p,i}'\boldsymbol{A}\boldsymbol{x}_{1,q,j} & \boldsymbol{x}_{R,p,i}'\boldsymbol{A}\boldsymbol{x}_{2,q,j} & \cdots & \boldsymbol{x}_{r,p,i}'\boldsymbol{A}\boldsymbol{x}_{R,q,j} \end{pmatrix}_{R\times R}. \qquad (3.12)$$

We then substitute (3.8) into the element in (3.12),

$$\begin{aligned}\mathbb{E}[\boldsymbol{x}_{r,q,i}'\boldsymbol{A}\boldsymbol{x}_{u,p,j}] &= \mathbb{E}[(\boldsymbol{D}_i\boldsymbol{\omega}_{r,q})'\boldsymbol{A}(\boldsymbol{D}_j\boldsymbol{\omega}_{u,p})] \\ &= \mathbb{E}[\boldsymbol{\omega}_{r,q}^{'}\boldsymbol{D}_i'\boldsymbol{A}\boldsymbol{D}_j\boldsymbol{\omega}_{u,p}].\end{aligned}$$

Define $\boldsymbol{A}_{i,j} = \boldsymbol{D}_i'\boldsymbol{A}\boldsymbol{D}_j$, we have

$$\mathbb{E}[\boldsymbol{x}_{r,q,k}'\boldsymbol{A}\boldsymbol{x}_{u,p,k}] = \mathbb{E}[\boldsymbol{\omega}_{r,q}^{'}\boldsymbol{A}_{i,j}\boldsymbol{\omega}_{u,p}]. \qquad (3.13)$$

We now present the expectation of quadratic forms $\mathbb{E}[\boldsymbol{X}_{p,i}\boldsymbol{A}\boldsymbol{X}_{q,j}]$ in (3.10):

1. For $p = q$:

   Case 1: $u = r$,

$$\mathbb{E}[\boldsymbol{x}_{r,q,1}] = \mathbb{E}[\boldsymbol{x}_{r,p,1}] = \mathbb{E}[\boldsymbol{x}_{u,q,1}] = \mathbb{E}[\boldsymbol{x}_{u,p,1}] = p(r|q)\boldsymbol{\omega}_q;$$

$$\mathrm{Cov}(\boldsymbol{x}_{u,p,1}, \boldsymbol{x}_{r,q,1}) = \mathrm{Cov}(\boldsymbol{x}_{r,q,1}, \boldsymbol{x}_{r,q,1}) = p(r|q)(1 - p(r|q))diag(\boldsymbol{\omega}_q);$$

$$\begin{aligned}\operatorname{tr}[\boldsymbol{A}_{i,j}\operatorname{Cov}(\boldsymbol{x}_{u,p,1},\boldsymbol{x}_{r,q,1})] &= \operatorname{tr}[\boldsymbol{A}_{i,j}\operatorname{Cov}(\boldsymbol{x}_{r,q,1},\boldsymbol{x}_{r,q,1})]\\ &= \operatorname{tr}[\boldsymbol{A}_{i,j}p(r|q)(1-p(r|q))diag(\boldsymbol{\omega}_q)]\\ &= p(r|q)(1-p(r|q))\operatorname{tr}[\boldsymbol{A}_{i,j}diag(\boldsymbol{\omega}_q)]\\ &= (p(r|q)-p(r|q)p(r|q))\operatorname{tr}[\boldsymbol{A}_{i,j}diag(\boldsymbol{\omega}_q)];\end{aligned}$$

$$\begin{aligned}&\mathbb{E}[\boldsymbol{x}'_{u,p,1}\boldsymbol{A}\boldsymbol{x}_{r,q,1}]\\ =\ &\operatorname{tr}[\boldsymbol{A}_{i,j}\operatorname{Cov}(\boldsymbol{x}_{u,p,1},\boldsymbol{x}_{r,q,1})]+\mathbb{E}[\boldsymbol{x}'_{u,p,1}]\boldsymbol{A}_{i,j}\mathbb{E}[\boldsymbol{x}_{r,q,1}]\\ =\ &\operatorname{tr}[\boldsymbol{A}_{i,j}\operatorname{Cov}(\boldsymbol{x}_{r,q,1},\boldsymbol{x}_{r,q,1})]+\mathbb{E}[\boldsymbol{x}'_{r,q,1}]\boldsymbol{A}_{i,j}\mathbb{E}[\boldsymbol{x}_{r,q,1}]\\ =\ &(p(r|q)-p(r|q)^2)\operatorname{tr}[\boldsymbol{A}_{i,j}diag(\boldsymbol{\omega}_q)]+p(r|q)^2\boldsymbol{\omega}_q^{'}\boldsymbol{A}_{i,j}\boldsymbol{\omega}_q.\end{aligned}$$

Case 2: $u \neq r$,

$$\begin{cases}\mathbb{E}[\boldsymbol{x}_{u,q,1}] = p(u|q)\boldsymbol{\omega}_q\\ \mathbb{E}[\boldsymbol{x}_{r,q,1}] = p(r|q)\boldsymbol{\omega}_q\end{cases};$$

$$\operatorname{Cov}(\boldsymbol{x}_{u,p,1},\boldsymbol{x}_{r,q,1}) = \operatorname{Cov}(\boldsymbol{x}_{u,q,1},\boldsymbol{x}_{r,q,1}) = -p(u|q)p(r|q)diag(\boldsymbol{\omega}_q);$$

$$\begin{aligned}\operatorname{tr}[\boldsymbol{A}_{i,j}\operatorname{Cov}(\boldsymbol{x}_{u,p,1},\boldsymbol{x}_{r,q,1})] &= \operatorname{tr}[\boldsymbol{A}_{i,j}\operatorname{Cov}(\boldsymbol{x}_{u,q,1},\boldsymbol{x}_{r,q,1})]\\ &= \operatorname{tr}[-p(u|q)p(r|q)diag(\boldsymbol{\omega}_q)]\\ &= -p(u|q)p(r|q)\operatorname{tr}[\boldsymbol{A}_{i,j}diag(\boldsymbol{\omega}_q)]\\ &= -p(u|q)p(r|q)\operatorname{tr}[\boldsymbol{A}_{i,j}diag(\boldsymbol{\omega}_q)];\end{aligned}$$

$$\begin{aligned}&\mathbb{E}[\boldsymbol{x}'_{u,p,1}\boldsymbol{A}\boldsymbol{x}_{r,q,1}]\\ =\ &\operatorname{tr}[\boldsymbol{A}_{i,j}\operatorname{Cov}(\boldsymbol{x}_{u,p,1},\boldsymbol{x}_{r,q,1})]+\mathbb{E}[\boldsymbol{x}'_{u,p,1}]\boldsymbol{A}_{i,j}\mathbb{E}[\boldsymbol{x}_{r,q,1}]\\ =\ &\operatorname{tr}[\boldsymbol{A}_{i,j}\operatorname{Cov}(\boldsymbol{x}_{u,q,1},\boldsymbol{x}_{r,q,1})]+\mathbb{E}[\boldsymbol{x}'_{u,q,1}]\boldsymbol{A}_{i,j}\mathbb{E}[\boldsymbol{x}_{r,q,1}]\\ =\ &-p(u|q)p(r|q)\operatorname{tr}[\boldsymbol{A}_{i,j}diag(\boldsymbol{\omega}_q)]+p(r|q)p(u|q)\boldsymbol{\delta}'_q\boldsymbol{A}_{i,j}\boldsymbol{\omega}_q.\end{aligned}$$

We combine **Case 1** and **Case 2** to obtain the $R\times R$ partition matrices in the diagonal

of (3.10):

$$\begin{aligned}
&\mathbb{E}[\boldsymbol{X}'_{q,i}\boldsymbol{A}\boldsymbol{X}_{q,j}] \\
=\ &\mathrm{tr}[\boldsymbol{A}_{i,j}diag(\boldsymbol{\omega}_q)][diag(\boldsymbol{\pi}(q)) - \boldsymbol{\pi}(q)\boldsymbol{\pi}(q)'] + \boldsymbol{\pi}(q)\boldsymbol{\pi}(q)'\boldsymbol{\omega}_q^{'}\boldsymbol{A}_{i,j}\boldsymbol{\omega}_q,
\end{aligned}$$

where $\boldsymbol{\pi}(q)$ is an $R \times 1$ vector of $p(r|q)$, $r = 1, \ldots, R$ for $q = 1, \ldots, Q$, i.e., if q=1, R=2, we have $\boldsymbol{\pi}(1) = [p(1|1), p(2|1)]'$.

2. **For $p \neq q$:**

$$\begin{cases} \mathbb{E}[\boldsymbol{x}_{u,p,1}] = p(u|p)\boldsymbol{\omega}_p \\ \mathbb{E}[\boldsymbol{x}_{r,q,1}] = p(r|q)\boldsymbol{\omega}_q \end{cases};$$

$$\mathrm{Cov}(\boldsymbol{x}_{u,p,1}, \boldsymbol{x}_{r,q,1}) = 0;$$

$$\mathrm{tr}[\boldsymbol{A}_{i,j}\,\mathrm{Cov}(\boldsymbol{x}_{u,p,1}, \boldsymbol{x}_{r,q,1})] = 0;$$

$$\begin{aligned}
&\mathbb{E}[\boldsymbol{x}'_{u,p,1}\boldsymbol{A}\boldsymbol{x}_{r,q,1}] \\
=\ &\mathrm{tr}[\boldsymbol{A}_{i,j}\,\mathrm{Cov}(\boldsymbol{x}_{u,p,1}, \boldsymbol{x}_{r,q,1})] + \mathbb{E}(\boldsymbol{x}'_{u,p,1})\boldsymbol{A}_{i,j}\,\mathbb{E}(\boldsymbol{x}_{r,q,1}) \\
=\ &0 + p(u|p)\boldsymbol{\delta}'_p\boldsymbol{A}_{i,j}p(r|q)\boldsymbol{\omega}_q \\
=\ &p(u|p)p(r|q)\boldsymbol{\delta}'_p\boldsymbol{A}_{i,j}\boldsymbol{\omega}_q.
\end{aligned}$$

Hence, the $R \times R$ partition matrices for the off-diagonal of (3.10) where $p \neq q$ is

$$\mathbb{E}[\boldsymbol{X}'_{p,i}\boldsymbol{A}\boldsymbol{X}_{q,j}] = \boldsymbol{\pi}(q)\boldsymbol{\pi}(p)'\boldsymbol{\omega}_p^{'}\boldsymbol{A}_{i,j}\boldsymbol{\omega}_q.$$

We thus have the following formula, which can be easily built in a computer program to calculate the elements in $\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]$. In particular, the partitioned matrix $\mathbb{E}[\boldsymbol{X}'_{p,i}\boldsymbol{A}\boldsymbol{X}_{q,j}]$ in (3.10):

$$\begin{cases} T_q[diag(\boldsymbol{\pi}_q) - \boldsymbol{\pi}(q)\boldsymbol{\pi}(q)'] + \boldsymbol{\pi}(q)\boldsymbol{\pi}(q)'\boldsymbol{U}_{q,q}, \\ \boldsymbol{\pi}(q)\boldsymbol{\pi}(p)'\boldsymbol{U}_{p,q}, \quad p \neq q. \end{cases}$$

Where $\boldsymbol{U}_{p,q} = \boldsymbol{\omega}_p^{'}\boldsymbol{A}_{i,j}\boldsymbol{\omega}_q$, $T_q = trace\{\boldsymbol{A}_{i,j}diag(\boldsymbol{\omega}_q)\}$.

3.2.5.2 Expected Information Matrix for Detection

For detection, we have $\mathcal{I}^D(\boldsymbol{\zeta}) = \mathbb{E}(\boldsymbol{H}'\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}\boldsymbol{H}) = \boldsymbol{H}'\,\mathbb{E}(\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X})\boldsymbol{H}$, where $\boldsymbol{H}$ is as defined in Model 3.2. Note that $\mathbb{E}(\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X})$ can be obtained from the 'estimation problem'.

We make use of this analytical result to conduct some case studies in the next Chapter to demonstrate the usefulness of our proposed approach.

3.2.6 Properties of D-Optimal Designs

Through the case studies, we notice that for D-optimal criterion, the optimal designs do not depend on the conditional probability that the subject selects r^{th} answer for the q^{th}-type stimulus. We can further show that D-optimal design is invariant to the conditional probability $p(r|q)$.

Theorem 3.2.3 *The design optimizing $\phi_D\{\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]\}$ is invariant to the selection of conditional probability $p(r|q)$ whenever $p(r|q) > 0$ for all $r = 1, \ldots, R$ and $q = 1, \ldots, Q$.*

We now give the proof of Theorem 3.2.3 for estimation problem with $\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]$ when $\tau_{ISI} = \tau_{TR}$. Results and proofs for the other situations discussed in this work are similar and are thus omitted.

Suppose $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_Q]$, $\boldsymbol{X}_q = [\boldsymbol{X}_{q,1}, \ldots, \boldsymbol{X}_{q,K}]$, and $\boldsymbol{X}_{q,k} = [\boldsymbol{X}_{1,q,k}, \ldots, \boldsymbol{X}_{R,q,k}]$. The expected information matrix can be written as:

$$\begin{aligned}
\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}] &= \mathbb{E}[\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}|\boldsymbol{d}] \\
&= \mathbb{E}[(\boldsymbol{X}_1, ..., \boldsymbol{X}_Q)'\boldsymbol{A}(\boldsymbol{X}_1, ..., \boldsymbol{X}_Q)|\boldsymbol{d}] \\
&= \begin{pmatrix} \mathbb{E}[\boldsymbol{X}_1'\boldsymbol{A}\boldsymbol{X}_1|\boldsymbol{d}] & \cdots & \mathbb{E}[\boldsymbol{X}_1'\boldsymbol{A}\boldsymbol{X}_Q|\boldsymbol{d}] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[\boldsymbol{X}_Q'\boldsymbol{A}\boldsymbol{X}_1|\boldsymbol{d}] & \cdots & \mathbb{E}[\boldsymbol{X}_Q'\boldsymbol{A}\boldsymbol{X}_Q|\boldsymbol{d}] \end{pmatrix}. \qquad (3.14)
\end{aligned}$$

The partition matrix $\mathbb{E}[\boldsymbol{X}_1'\boldsymbol{A}\boldsymbol{X}_1|\boldsymbol{d}]$ in (3.14) when $\tau_{ISI} = \tau_{TR}$ is

$$\begin{cases} [diag(\boldsymbol{\pi}_q) - \boldsymbol{\pi}(q)\boldsymbol{\pi}(q)'] \otimes \boldsymbol{T}_q + \boldsymbol{\pi}(q)\boldsymbol{\pi}(q)' \otimes \boldsymbol{V}_{q,q}, \\ \boldsymbol{\pi}(q)\boldsymbol{\pi}(p)' \otimes \boldsymbol{V}_{p,q}, \quad p \neq q. \end{cases}$$

where the $(i,j)^{th}$ element of $\boldsymbol{T}_q$ is $((\boldsymbol{T}_q))_{ij} = trace\{(\boldsymbol{L}')^{i-1}\boldsymbol{A}\boldsymbol{L}^{j-1}diag(\boldsymbol{\delta}_q)\}$, and the $(i,j)^{th}$ element of $\boldsymbol{V}_{pq}$ is $((\boldsymbol{V}_{pq}))_{ij} = \boldsymbol{\delta}_p'(\boldsymbol{L}')^{i-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{L}^{j-1}\boldsymbol{\delta}_q$; $i,j = 1,...,K$, $p,q = 1,...,Q$; $\boldsymbol{L}$ and $\boldsymbol{\delta}_q$ are same as in Example 1; The remaining parts are same as in Theorem 3.2.1.

When finding D-optimal designs, we only need to focus on designs yielding a positive determinant $|\,\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]|$ with a non-singular $\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]$. As indicated in the next result, these designs will also make all the $\boldsymbol{T}_q$'s non-singular.

Lemma 3.2.4 *Suppose $p(r|q) > 0$ for all $r = 1,...,R$, and $q = 1,...,Q$, and $\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]$ is non-singular. Then, all the $\boldsymbol{T}_q$'s, $q = 1,...,Q$ are non-singular.*

To prove this result, we first note that $\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}] = \mathbb{E}[\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}|\boldsymbol{d}]$, and $\boldsymbol{A}$ can be written as $\boldsymbol{A} = \boldsymbol{C}'\boldsymbol{C}$ for some matrix $\boldsymbol{C}$; see, e.g., Theorem 2.6c of Rencher and Schaalje (2008). Thus, for any given vector $\boldsymbol{z}$ of an appropriate dimension, we have

$$\begin{aligned}\boldsymbol{z}'\,\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]\boldsymbol{z} &= \mathbb{E}[\boldsymbol{z}'\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}\boldsymbol{z}|\boldsymbol{d}] \\ &= \mathbb{E}[\boldsymbol{z}'\boldsymbol{X}'\boldsymbol{C}'\boldsymbol{C}\boldsymbol{X}\boldsymbol{z}|\boldsymbol{d}] = \mathbb{E}[||\boldsymbol{C}\boldsymbol{X}\boldsymbol{z}||^2|\boldsymbol{d}] \geq 0.\end{aligned}$$

Here, $||\boldsymbol{a}||^2 = \boldsymbol{a}'\boldsymbol{a}$ is the squared length of the vector $\boldsymbol{a}$, and is always non-negative. By definition, this indicates that $\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]$ is a non-negative definite matrix. With this observation and the assumption that $p(r|q) > 0$ for all r, q, we now prove Lemma 3.2.4 by showing that $\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]$ is not positive definite (and is thus singular) whenever $\boldsymbol{T}_q$ is a singular matrix for some q. To this end, we will construct a non-zero vector $\tilde{\boldsymbol{z}}$ that makes $\tilde{\boldsymbol{z}}'\,\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]\tilde{\boldsymbol{z}} = 0$. Without loss of generality, we suppose that $\boldsymbol{T}_1$ is a singular matrix. The $\tilde{\boldsymbol{z}}$ that we construct will then have the form of $\tilde{\boldsymbol{z}}' = (\tilde{\boldsymbol{z}}_1' \otimes \tilde{\boldsymbol{z}}_2', \boldsymbol{0}')$. Here, $\tilde{\boldsymbol{z}}_1'$ is any non-zero vector that is orthogonal to $\boldsymbol{\pi}_1$ (i.e., $\tilde{\boldsymbol{z}}_1'\boldsymbol{\pi}_1 = 0$), and $\tilde{\boldsymbol{z}}_2'$ is an eigenvector of $\boldsymbol{T}_1$ with $\tilde{\boldsymbol{z}}_2'\boldsymbol{T}_1\tilde{\boldsymbol{z}}_2 = 0$. Consequently, $\tilde{\boldsymbol{z}}$ is a non-zero vector with

$$\begin{aligned}\tilde{\boldsymbol{z}}'\,\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]\tilde{\boldsymbol{z}} &= (\tilde{\boldsymbol{z}}_1' \otimes \tilde{\boldsymbol{z}}_2')\,\mathbb{E}[\boldsymbol{X}_1'\boldsymbol{A}\boldsymbol{X}_1|\boldsymbol{d}](\tilde{\boldsymbol{z}}_1 \otimes \tilde{\boldsymbol{z}}_2) \\ &= \tilde{\boldsymbol{z}}_1'[diag(\boldsymbol{\pi}_1) - \boldsymbol{\pi}_1\boldsymbol{\pi}_1']\tilde{\boldsymbol{z}}_1 \otimes \tilde{\boldsymbol{z}}_2'\boldsymbol{T}_1\tilde{\boldsymbol{z}}_2 + \tilde{\boldsymbol{z}}_1'[\boldsymbol{\pi}_1\boldsymbol{\pi}_1']\tilde{\boldsymbol{z}}_1 \otimes \tilde{\boldsymbol{z}}_2'\boldsymbol{V}_{11}\tilde{\boldsymbol{z}}_2 = 0.\end{aligned}$$

The first equality follows from (3.14). This then proves our claim. With this lemma, we only need to consider cases where all the $\boldsymbol{T}_q$'s are non-singular. We now rewrite $\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]$ as

$$\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}] = \begin{pmatrix} diag(\boldsymbol{\pi}_1) \otimes \boldsymbol{T}_1 & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{O} & diag(\boldsymbol{\pi}_2) \otimes \boldsymbol{T}_2 & \vdots & \vdots \\ \vdots & \vdots & \ddots & \boldsymbol{O} \\ \boldsymbol{O} & \cdots & \boldsymbol{O} & diag(\boldsymbol{\pi}_Q) \otimes \boldsymbol{T}_Q \end{pmatrix} + \begin{pmatrix} \boldsymbol{\pi}_1\boldsymbol{\pi}_1' \otimes [\boldsymbol{V}_{11} - \boldsymbol{T}_1] & \boldsymbol{\pi}_1\boldsymbol{\pi}_2' \otimes \boldsymbol{V}_{12} & \cdots & \boldsymbol{\pi}_1\boldsymbol{\pi}_Q' \otimes \boldsymbol{V}_{1Q} \\ \boldsymbol{\pi}_2\boldsymbol{\pi}_1' \otimes \boldsymbol{V}_{21} & \boldsymbol{\pi}_2\boldsymbol{\pi}_2' \otimes [\boldsymbol{V}_{22} - \boldsymbol{T}_2] & \cdots & \boldsymbol{\pi}_2\boldsymbol{\pi}_Q' \otimes \boldsymbol{V}_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\pi}_Q\boldsymbol{\pi}_1' \otimes \boldsymbol{V}_{Q1} & \boldsymbol{\pi}_Q\boldsymbol{\pi}_2' \otimes \boldsymbol{V}_{Q2} & \cdots & \boldsymbol{\pi}_Q\boldsymbol{\pi}_Q' \otimes [\boldsymbol{V}_{QQ} - \boldsymbol{T}_Q] \end{pmatrix}.$$

We note that the first matrix on the right hand side is non-singular. With, e.g., a QR decomposition, and the fact that $(\boldsymbol{C}_1\boldsymbol{C}_2) \otimes (\boldsymbol{D}_1\boldsymbol{D}_2) = (\boldsymbol{C}_1 \otimes \boldsymbol{D}_1)(\boldsymbol{C}_2 \otimes \boldsymbol{D}_2)$, the second matrix can be expressed as

$$\begin{pmatrix} \boldsymbol{\pi}_1 \otimes \boldsymbol{A}_1 \\ \boldsymbol{\pi}_2 \otimes \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{\pi}_Q \otimes \boldsymbol{A}_Q \end{pmatrix} [\boldsymbol{\pi}_1' \otimes \boldsymbol{B}_1, \boldsymbol{\pi}_2' \otimes \boldsymbol{B}_2, \cdots, \boldsymbol{\pi}_Q' \otimes \boldsymbol{B}_Q]$$

for some $\boldsymbol{A}_1, ..., \boldsymbol{A}_Q$ and $\boldsymbol{B}_1, ..., \boldsymbol{B}_Q$. We then obtain $|\,\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]|$ by applying the following result which can be found in Theorem 18.1.1 of Harville (1997):

Lemma 3.2.5 *Let $\boldsymbol{R}$ be a non-singular matrix, $\boldsymbol{I}_m$ be the identity matrix of size m, $\boldsymbol{S}$ be an n-by-m matrix and $\boldsymbol{U}$ be an m-by-n matrix. We have $|\boldsymbol{R} + \boldsymbol{S}\boldsymbol{U}| = |\boldsymbol{R}||\boldsymbol{I}_m + \boldsymbol{U}\boldsymbol{R}^{-1}\boldsymbol{S}|$.*

With some algebra, $|\,\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]|$ is then

$$\prod_{q=1}^{Q} |diag(\boldsymbol{\pi}_q)|^K \times \prod_{q=1}^{Q} |\boldsymbol{T}_q|^{R_q} \times \left| \boldsymbol{I}_K + \sum_{q=1}^{Q} \boldsymbol{B}_q \boldsymbol{T}_q^{-1} \boldsymbol{A}_q \right|.$$

For given $\boldsymbol{\pi}_q$'s whose elements are all positive, $\prod_{q=1}^{Q} |diag(\boldsymbol{\pi}_q)|^K = c$ is a positive constant. The value of c is the same for all competing designs, and thus does not change the ranking

of the designs under the D-criterion. Consequently, a design that maximizes $|\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]|$ for a given c will remains to be D-optimal for another value of c (i.e. another specific set of $\boldsymbol{\pi}_q$'s) whenever $c > 0$. The D-optimal design is thus invariant to the selection of $p(r|q)$.

3.3 Case Studies

In this section, we apply our proposed approach to optimize ϕ_1. For comparison, we follow the design selection criterion considered by Cordes *et al.* (2012) that is linked to ϕ_2. The goodness of $\boldsymbol{d}$'s using ϕ_2 is evaluated by conducting a Monte Carlo simulation to generate κ, say 100, realizations of $\phi(\boldsymbol{M})$, and then approximate ϕ_2 by a summary statistic such as the mean/median of the κ realizations of the ϕ-value. Here, we use the mean for all cases. Both criteria, ϕ_1 and ϕ_2, which are sometimes viewed as the pseudo-Bayesian versions of the optimality criteria, have been considered in the design literature; see Ch.18 of Atkinson *et al.* (2007). However, finding optimal designs with respect to the latter criterion is very time consuming as also indicated in Cordes *et al.* (2012). As we demonstrate below, with our analytical results presented in the previous section, obtaining an optimal design based on ϕ_1 is computationally much simpler than the use of ϕ_2; while ϕ_1 can itself serve as the design selection criterion, it also gives a very good surrogate criterion if the experimenter would like to find ϕ_2-optimal designs.

We now demonstrate the usefulness of our proposed approach where ϕ_1 is used as the optimality criterion. Both study objectives, namely the estimation of the HRFs and the detection of brain activity, are considered along with: (1) Model (3.1) with $\tau_{ISI} \neq \tau_{TR}$; (2) Model (3.2) with $\tau_{ISI} = \tau_{TR}$; (3) Model (3.2) with $\tau_{ISI} \neq \tau_{TR}$. Note that the case where Model (3.1) is used with $\tau_{ISI} = \tau_{TR}$ has been studied in Zhou (2014). Selected results will be presented in the next section.

We consider two (Q, N) combinations: $(1, 255)$, $(2, 242)$ with $R = 2$, also we set $\tau_{ISI}/\tau_{TR} = 2\,\text{s}/2\,\text{s}$ and $\tau_{ISI}/\tau_{TR} = 3\,\text{s}/1.5\,\text{s}$ as in Cordes *et al.* (2012), so that ΔT is equal to 2 and 1.5, respectively. The drift of the time series, $\boldsymbol{S\gamma}$ is assumed to be a second order Legendre polynomial. We also assume that the noise follows an stationary AR(1)

process with an autocorrelation coefficient of 0.3; other correlation coefficients can also be considered.

To compare with Cordes *et al.* (2012), we further consider a case follows their experimental settings, that is Model (3.2) with $\tau_{ISI} \neq \tau_{TR}$ for $(Q, N) = (3, 402)$ with specific contrasts of the effects of the stimuli, specifically, $\boldsymbol{\theta} = (\theta_{1|1}, \theta_{2,1}, \ldots, \theta_{1|3}, \theta_{2,3})'$; i.e. the unknown coefficients in Model (3.2), in Cordes *et al.* (2012), the contracts are defined by

1. recollection contract: $\left[(\theta_{1|1} + \theta_{2|2}) - (\theta_{1|2} + \theta_{2|1})\right]/2$;
2. familiarity contract: $(\theta_{1|2} + \theta_{2|1})/2 - \theta_{1|3}$.

The noise follows an AR(1) process, the autocorrelation coefficient is set to 0.2.

Table 3.1: The conditional probability matrices: we define $\boldsymbol{P}_Q^{(c)}$ as the probability matrix for Q stimulus types for case c, where c can be I, II or III.

Q	Probability matrices for different cases		
1	$\boldsymbol{P}_1^{(I)} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$	$\boldsymbol{P}_1^{(II)} = \begin{bmatrix} 0.2 \\ 0.8 \end{bmatrix}$	
2	$\boldsymbol{P}_2^{(I)} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$	$\boldsymbol{P}_2^{(II)} = \begin{bmatrix} 0.5 & 0.2 \\ 0.5 & 0.8 \end{bmatrix}$	$\boldsymbol{P}_2^{(III)} = \begin{bmatrix} 0.7 & 0.2 \\ 0.3 & 0.8 \end{bmatrix}$
3	$\boldsymbol{P}_3^{(I)} = \begin{bmatrix} 0.88 & 0.28 & 0.93 \\ 0.12 & 0.72 & 0.07 \end{bmatrix}$		

In addition, for calculating the expectations $\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}]$ and $\mathbb{E}[\phi(\boldsymbol{M})|\boldsymbol{d}]$, we define the matrix of conditional probabilities $\boldsymbol{P}_Q^{(c)}$ with $\boldsymbol{P}_Q^{(c)} = [\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_Q]$, where Q is the number of stimulus type, and the superscript c is the index for different cases that we considered. We consider six cases as listed in Table 3.1, including the case with $\boldsymbol{P}_3^I$ of Cordes *et al.* (2012).

Note that for $\mathbb{E}[\phi(\boldsymbol{M})|\boldsymbol{d}]$, the conditional probabilities are used to generate κ realizations of $\boldsymbol{M}$ for each given $\boldsymbol{d}$. These realizations are then used to calculate the approximation $\phi_2(\boldsymbol{d}; \kappa)$ of ϕ_2. In what follows, we will first adapt the genetic algorithm of Kao *et al.*

(2009) to obtain a design, $\boldsymbol{d}_{GA}$, that maximizes ϕ_1. With this optimality criterion, for case (1), (2) and (3), we compare our obtained designs with some traditional designs that are popular in practice (for different purposes). These traditional designs include random designs, m-sequences, and block designs:

Random designs. For random designs $\boldsymbol{d}_{rand}$, each element of a design is generated from a discrete uniform distribution over $\{0, 1, ..., Q\}$, forming 100 random designs, the mean and standard deviation of the ϕ_1-value over these 100 random designs are calculated.

M-sequences. These designs are to be denoted as $\boldsymbol{d}_{mseq}$. They are also known as maximum-length shift register sequences, and are introduced into fMRI by Buracas and Boynton (2002). These designs are known to perform well for estimating the HRF, and can be easily generated by the MATLAB program of Liu and Frank (2004).

Block designs. We consider block design $\boldsymbol{d}_{block}$ having a 16 s-on-16 s-off pattern. For example, when $Q = 1$, the first 16 seconds is the off-period, and no stimulus is shown to the subject. In the next 16 seconds, stimuli of the same type is shown to the subject every τ_{ISI} seconds. This is repeated for several cycles until the end of the experiment. In particular, a $\boldsymbol{d}_{block}$ may look like $\{000000001111111100000000 \cdots 0\}$ when $Q = 1$. They are known to be useful for the detection of activated brain voxels. But they do not perform well when the focus is on the estimation of the HRF, and may give rise to confounding psychological effects such as subject habituation or anticipation.

For all these traditional designs, we compare their ϕ_1-values to that of $\boldsymbol{d}_{GA}$. In addition, for case (1), (2) and (3), we use the genetic algorithm to obtain a design $\boldsymbol{d}_{r100}$ that maximizes $\phi_2(\boldsymbol{d}; 100)$. The resulting designs is compared with $\boldsymbol{d}_{GA}$ in terms of ϕ_1. To demonstrate that ϕ_1 provides a good surrogate for ϕ_2, we also compare the ϕ_2-values of $\boldsymbol{d}_{r100}$, and $\boldsymbol{d}_{GA}$ as well as the CPU times needed for generating these two types of designs. For this latter comparison, ϕ_2 is approximated by $\phi_2(\boldsymbol{d};\ \kappa = 1000)$ even though $\boldsymbol{d}_{r100}$ is obtained with $\phi_2(\boldsymbol{d};\ \kappa = 100)$. We note that $\phi_2(\boldsymbol{d}; 1000)$ is expected to have a higher precision than

$\phi_2(\boldsymbol{d}; 100)$ for approximating ϕ_2. However, the calculation of $\phi_2(\boldsymbol{d}; 1000)$ is computationally very expensive, and is thus difficult, if not infeasible, to be considered for obtaining $\boldsymbol{d}_{r100}$.

For the case similar to Cordes *et al.* (2012), we compare three methods for obtaining A-optimal designs. In addition to $\boldsymbol{d}_{GA}$, $\boldsymbol{d}_{r100}$ that maximizes ϕ_1 and $\phi_2(\boldsymbol{d}; \kappa = 100)$ respectively, we use GA of Cordes *et al.* (2012) for optimizing $\phi_2(\boldsymbol{d}; \kappa = 1000)$ in the search of an A-optimal design. The results are presented in Section 3.4.3.

All the computations are conducted on a desktop computer with a 3.4 GHz Intel Core i7-2600 quad-core processor.

3.4 Results

We have evaluated our proposed approach with all the cases described in the previous section. In this section, we present the simulation results under A-optimality and D-optimality in Section 3.4.1 and 3.4.2, respectively. Section 3.4.3 provides the results of the case similar to Cordes *et al.* (2012)'s study.

3.4.1 Evaluation Under the A-Optimality Criterion

3.4.1.1 Estimation Problem: Model (3.1) with $\tau_{ISI} \neq \tau_{TR}$

We evaluate the design approaches described in Section 3.3 with A-optimality criterion for Model (3.1) when $\tau_{ISI} = 3\,\text{s}$ and $\tau_{TR} = 1.5\,\text{s}$.

Design Comparisons in Terms of ϕ_1

Table 3.2 presents the results for cases with one and two stimulus types under the A-optimality criterion. The efficiency ratio $\phi_1(\boldsymbol{d})/\phi_1(\boldsymbol{d}_{GA})$ of these traditional fMRI designs and $\phi_1(\boldsymbol{d}_{r100})$ to our $\phi_1(\boldsymbol{d}_{GA})$ are shown in Figure 3.1. The good performance of $\boldsymbol{d}_{GA}$ is consistently demonstrated in Table 3.2 in all the cases that we studied. In addition, in Figure 3.1, it is clear that no design has a higher ϕ_1-value than $\boldsymbol{d}_{GA}$ among the designs we considered since the ratios $\phi_1(\boldsymbol{d})/\phi_1(\boldsymbol{d}_{GA})$ are all less than 1 for any other design $\boldsymbol{d}$.

Table 3.2: The ϕ_1-values for different designs with $Q = 1, 2$ evaluated by the A-optimality criterion for estimation when $\tau_{ISI} \neq \tau_{TR}$.

Case	$\boldsymbol{d}_{GA}$	$\boldsymbol{d}_{rand}$ (mean±std)	$\boldsymbol{d}_{block}$	$\boldsymbol{d}_{mseq}$	$\boldsymbol{d}_{r100}$
$\boldsymbol{P}_1^{(I)}$	40.6974	37.8775 ± 0.0025	0.5333	38.8668	40.6526
$\boldsymbol{P}_1^{(II)}$	30.8022	27.4689 ± 0.0031	0.5305	27.9125	30.6294
$\boldsymbol{P}_2^{(I)}$	24.8050	23.0556 ± 0.0018	0	24.6212	24.6802
$\boldsymbol{P}_2^{(II)}$	21.0975	19.4582 ± 0.0022	0	20.6775	20.8126
$\boldsymbol{P}_2^{(III)}$	19.9094	18.4382 ± 0.0022	0	19.5695	19.7783

It is noteworthy that the block designs perform poorly in terms of the ϕ_1 criterion for the estimation problem. These designs are not recommended when the study objective lies in the estimation of the HRF.

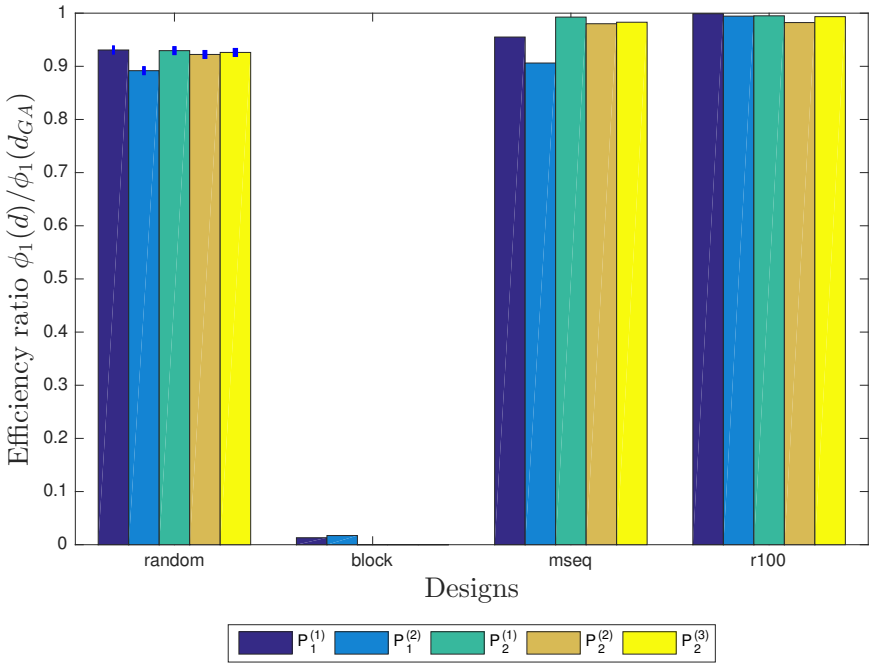

Figure 3.1: Relative design efficiencies with the A-optimality criterion for estimation when $\tau_{ISI} \neq \tau_{TR}$: this plot provides the relative efficiency $\phi_1(\boldsymbol{d})/\phi_1(\boldsymbol{d}_{GA})$ of different designs $\boldsymbol{d}$ with $Q = 1, 2$ for five different cases corresponding to $\boldsymbol{P}_Q^{(c)}$ with the A-optimality criterion.

Design Comparisons in Terms of $\phi_2(\boldsymbol{d}; 1000)$

In this subsection, we compare $\boldsymbol{d}_{GA}$, $\boldsymbol{d}_{r100}$ in terms of $\phi_2(\boldsymbol{d}; 1000)$ and the corresponding CPU time required for obtaining them. Specifically, for $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$, we generate $\kappa = 1000$ corresponding vectors $\boldsymbol{r}$ of the subject's answers for the calculation of $\phi_2(\boldsymbol{d}_{GA}; 1000)$, and $\phi_2(\boldsymbol{d}_{r100}; 1000)$.

The results in Table 3.3 for the A-optimality criterion suggest that $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ have similar performance with respect to $\phi_2(\boldsymbol{d}; 1000)$. It is even clear as presented in Figure 3.2, where the bars correspond to $\phi_2(\boldsymbol{d}_{GA}; 1000)/\phi_2(\boldsymbol{d}_{r100}; 1000)$. While $\boldsymbol{d}_{GA}$ is obtained by considering ϕ_1-value, it slightly outperforms $\boldsymbol{d}_{r100}$ in some cases, when $\phi_2(\boldsymbol{d}; 1000)$ is considered for design evaluations. Some ratios of $\phi_2(\boldsymbol{d}_{GA}; 1000)/\phi_2(\boldsymbol{d}_{r100}; 1000)$ are greater than 1 in Figure 3.2. This might be because that the two criteria lead to similar designs as well as the randomness of the search algorithm.

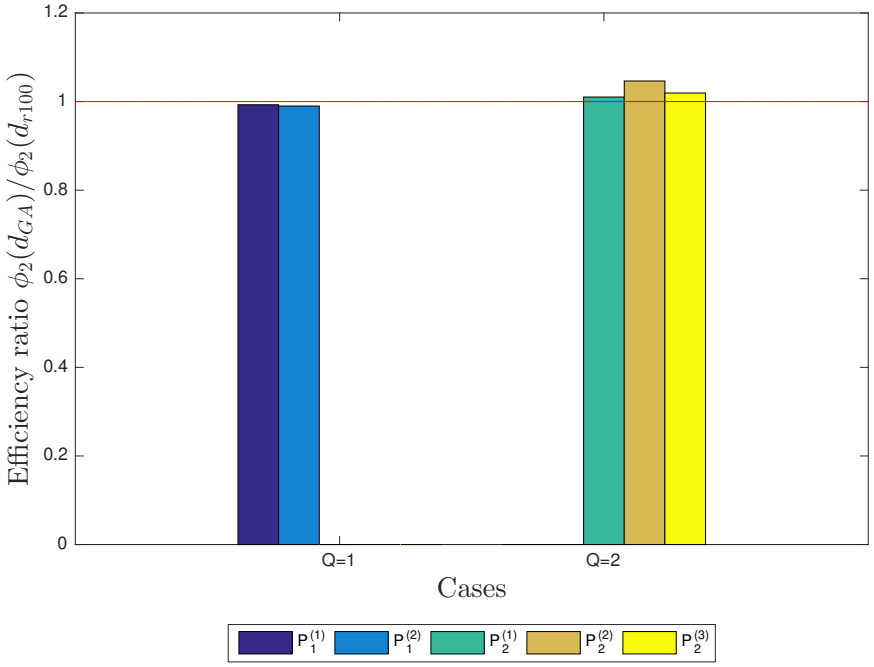

Figure 3.2: Relative design efficiencies with the A-optimality criterion for estimation when $\tau_{ISI} \neq \tau_{TR}$: this plot provides the relative efficiency $\phi_2(\boldsymbol{d}_{GA}; 1000)/\phi_2(\boldsymbol{d}_{r100}; 1000)$ with $Q = 1, 2$ for five different cases corresponding to $\boldsymbol{P}_Q^{(c)}$ with the A-optimality.

Table 3.3: The $\phi_2(\boldsymbol{d}; 1000)$-values of $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ with $Q = 1, 2$ under the A-optimality criterion for estimation ($\tau_{ISI} \neq \tau_{TR}$).

$\boldsymbol{P}_Q^{(c)}$	$\boldsymbol{P}_1^{(I)}$	$\boldsymbol{P}_1^{(II)}$	$\boldsymbol{P}_2^{(I)}$	$\boldsymbol{P}_2^{(II)}$	$\boldsymbol{P}_2^{(III)}$
$\boldsymbol{d}_{GA}$	36.1399	26.7289	17.8997	14.8367	14.0760
$\boldsymbol{d}_{r100}$	36.3982	26.9994	17.7174	14.1778	13.8087

Comparisons of CPU Time

The current results show that designs optimizing ϕ_1 can also perform well with respect to ϕ_2. One major advantage for considering the former criterion for obtaining designs is further evident in Table 3.4 that present the CPU times needed for obtaining $\boldsymbol{d}_{GA}$ under ϕ_1 and $\boldsymbol{d}_{r100}$ under $\phi_2(\cdot; 100)$, it takes much less CPU time to obtain a $\boldsymbol{d}_{GA}$ than $\boldsymbol{d}_{r100}$. The use of the former design is thus recommended.

Table 3.4: CPU times (hours) for obtaining A-optimal designs that optimize $\phi_1(\boldsymbol{d}_{GA})$ and $\phi_2(\boldsymbol{d}_{r100}; 100)$ respectively for $Q = 1, 2$ for estimation when $\tau_{ISI} \neq \tau_{TR}$.

$\boldsymbol{P}_Q^{(c)}$	$\boldsymbol{P}_1^{(I)}$	$\boldsymbol{P}_1^{(II)}$	$\boldsymbol{P}_2^{(I)}$	$\boldsymbol{P}_2^{(II)}$	$\boldsymbol{P}_2^{(III)}$
$T_{\phi_1(\boldsymbol{d}_{GA})}$	1.6167	1.5896	1.8103	3.1092	1.7774
$T_{\phi_2(\boldsymbol{d}_{r100};100)}$	287.4444	272.0000	310.6389	128.7833	192.1833

Robustness of Design Against Misspecification

The optimal designs are obtained based on the pre-specified conditional probabilities, usually obtained from a pilot study is not necessary the one to be occurred during the fMRI scanning. In terms of ϕ_1, we study the performance of our obtained designs when these probabilities are misspecified at the design stage, the optimal design obtained with a specified set of probabilities is evaluated by using ϕ_1 under the case when another set of probabilities is true. The ϕ_1-values are listed in Table 3.5 for $Q = 1$ and Table 3.6 for $Q = 2$. The results

show that the ϕ_1-values of misspecified designs are very close to the optimal design for each case. We obtained similar results for other cases we studied for the A-optimality designs, thus we will omit these results in the following subsections.

Table 3.5: Robustness of the A-optimal designs for estimation when $\tau_{ISI} \neq \tau_{TR}$: $\boldsymbol{d}^*_{GA}(\boldsymbol{P}^{(c)}_Q)$ is the optimal design obtained based on probability $\boldsymbol{P}^{(c)}_Q$ with $Q = 1$.

Optimal design	$\boldsymbol{P}^{(I)}_1$	$\boldsymbol{P}^{(II)}_1$	$\frac{\phi_1(\text{misspecified})}{\phi_1(\text{optimal})}$
$\boldsymbol{d}^*_{GA}(\boldsymbol{P}^{(I)}_1)$	40.6974	40.2521	0.989
$\boldsymbol{d}^*_{GA}(\boldsymbol{P}^{(II)}_1)$	30.4563	30.8022	0.989

Table 3.6: Robustness of the A-optimal designs for estimation when $\tau_{ISI} \neq \tau_{TR}$: $\boldsymbol{d}^*_{GA}(\boldsymbol{P}^{(c)}_Q)$ is the optimal design obtained based on probability $\boldsymbol{P}^{(c)}_Q$ with $Q = 2$.

Optimal design	$\boldsymbol{P}^{(I)}_2$	$\boldsymbol{P}^{(II)}_2$	$\boldsymbol{P}^{(III)}_2$	$\frac{\phi_1(\text{misspecified})}{\phi_1(\text{optimal})}$
$\boldsymbol{d}^*_{GA}(\boldsymbol{P}^{(I)}_2)$	24.8050	24.5688	24.6094	0.990-0.992
$\boldsymbol{d}^*_{GA}(\boldsymbol{P}^{(II)}_2)$	20.8226	21.0975	21.0238	0.987-0.997
$\boldsymbol{d}^*_{GA}(\boldsymbol{P}^{(III)}_2)$	19.7573	19.9174	19.9094	0.992-1.000

The Obtained Optimal Designs

Table 3.7 presents the stimulus frequencies of the obtained A-optimal designs for estimation problem when $\tau_{ISI} \neq \tau_{TR}$ based on ϕ_1 and ϕ_2. Among all cases, the frequency of the occurrences for the q^{th} stimulus type are similar to each other, and it tends to depend on the probability $P^{(c)}_Q$. For example, in Table 3.7, the stimulus frequency for equal probability $P^{(I)}_1$ is smaller than that of unequal probability $P^{(II)}_1$, similar trend is observed in $Q = 2$.

Table 3.7: Stimulus frequencies of A-optimal designs for estimation when $\tau_{ISI} \neq \tau_{TR}$: the designs are obtained by optimizing $\phi_1(\boldsymbol{d}_{GA})$ and $\phi_2(\boldsymbol{d}_{r100}; 100)$ with $\boldsymbol{P}_Q^{(c)}$ for $Q = 1, 2$.

	ϕ_1		ϕ_2	
Stimulus type (q)	1	2	1	2
$Q = 1, P_1^{(I)}$	59%	-	59%	-
$Q = 1, P_1^{(II)}$	64%	-	65%	-
$Q = 2, P_2^{(I)}$	34%	33%	35%	34%
$Q = 2, P_2^{(II)}$	31%	39%	33%	37%
$Q = 2, P_2^{(III)}$	34%	38%	35%	36%

3.4.1.2 Detection Problem: Model 3.2 with $\tau_{ISI} = \tau_{TR}$

In this section, we work with Model (3.2) for detecting brain activations under the A-optimality criterion, τ_{ISI} and τ_{TR} are both set to 2 s. Same design approaches described in Section 3.3 are compared here.

Design Comparisons in Terms of ϕ_1

Table 3.8 and Figure 3.3 provide the results for $Q = 1, 2$. They all demonstrate the good performance of the designs, $\boldsymbol{d}_{GA}$, obtained by our approach. More clearly, from Figure 3.3, we can see that $\boldsymbol{d}_{r100}$ performs similarly to $\boldsymbol{d}_{GA}$. The block designs $\boldsymbol{d}_{block}$ also perform relatively well, which are as stated in the literature that block design are powerful term for signal detection Wager and Nichols (2003). Moreover, random designs and m-sequence do not perform well in terms of the ϕ_1 criterion. Hence, these designs are not recommended for signal detection study when $\tau_{ISI} = \tau_{TR}$.

Table 3.8: The ϕ_1-values for different designs with $Q = 1, 2$ evaluated by the A-optimality criterion for detection when $\tau_{ISI} = \tau_{TR}$.

Case	$\boldsymbol{d}_{GA}$	$\boldsymbol{d}_{rand}$ (mean±std)	$\boldsymbol{d}_{block}$	$\boldsymbol{d}_{mseq}$	$\boldsymbol{d}_{r100}$
$\boldsymbol{P}_1^{(I)}$	50.2272	26.9807 ± 0.0037	45.3967	26.9543	49.8770
$\boldsymbol{P}_1^{(II)}$	33.0788	19.2655 ± 0.0030	27.3789	19.3592	31.4719
$\boldsymbol{P}_2^{(I)}$	29.2910	16.7620 ± 0.0032	28.4515	17.1499	28.9776
$\boldsymbol{P}_2^{(II)}$	22.9827	14.0640 ± 0.0031	21.4885	14.3970	22.4312
$\boldsymbol{P}_2^{(III)}$	21.1488	13.3584 ± 0.0026	19.8565	13.6304	20.7085

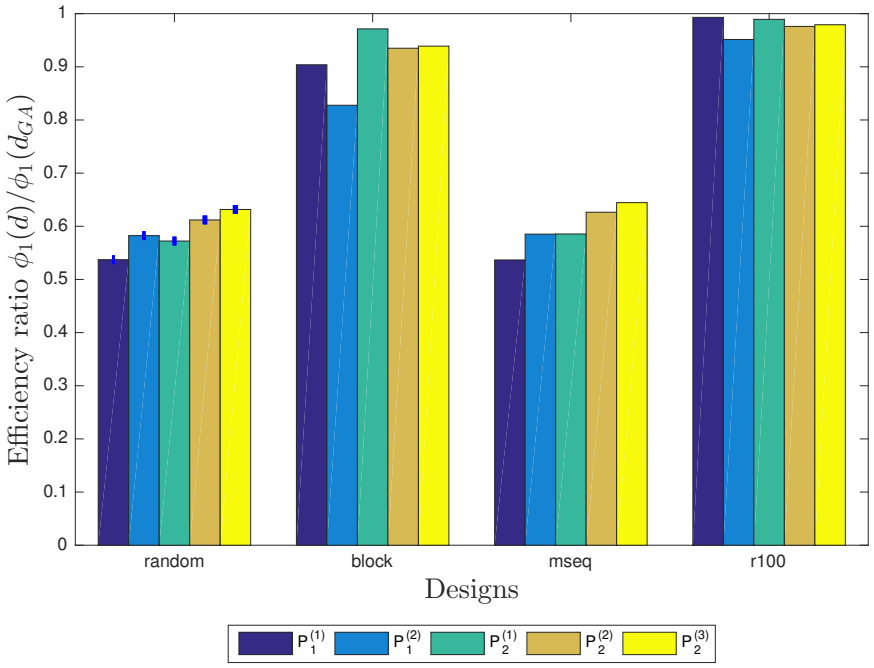

Figure 3.3: Relative design efficiencies with the A-optimality criterion for detection when $\tau_{ISI} = \tau_{TR}$: this plot provides the relative efficiency $\phi_1(\boldsymbol{d})/\phi_1(\boldsymbol{d}_{GA})$ of different designs $\boldsymbol{d}$ with $Q = 1$ and $Q = 2$ for five different cases corresponding to $p(r|q)$ with the A-optimality criterion.

Design Comparisons in Terms of $\phi_2(\boldsymbol{d}; 1000)$

Now, we compare $\boldsymbol{d}_{GA}$, $\boldsymbol{d}_{r100}$ in terms of $\phi_2(\boldsymbol{d}; 1000)$. We adopt the same method in Section 3.4.1.1. The results in Table 3.9 suggest that under the A-Optimality, both methods have similar performance in terms of $\phi_2(\boldsymbol{d}; 1000)$. Based on the efficiency ratio in Figure 3.4, we conclude that $\boldsymbol{d}_{GA}$ obtained from ϕ_1 still perform reasonably well evaluated by $\phi_2(\boldsymbol{d}; 1000)$.

Table 3.9: The $\phi_2(\boldsymbol{d}; 1000)$-values of $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ with $Q = 1, 2$ under the A-optimality criterion for detection When $\tau_{ISI} = \tau_{TR}$.

$\boldsymbol{P}_Q^{(c)}$	$\boldsymbol{P}_1^{(I)}$	$\boldsymbol{P}_1^{(II)}$	$\boldsymbol{P}_2^{(I)}$	$\boldsymbol{P}_2^{(II)}$	$\boldsymbol{P}_2^{(III)}$
$\boldsymbol{d}_{GA}$	13.7983	10.4764	4.7104	4.8410	5.6410
$\boldsymbol{d}_{r100}$	14.7420	11.2888	5.3076	5.8229	5.2825

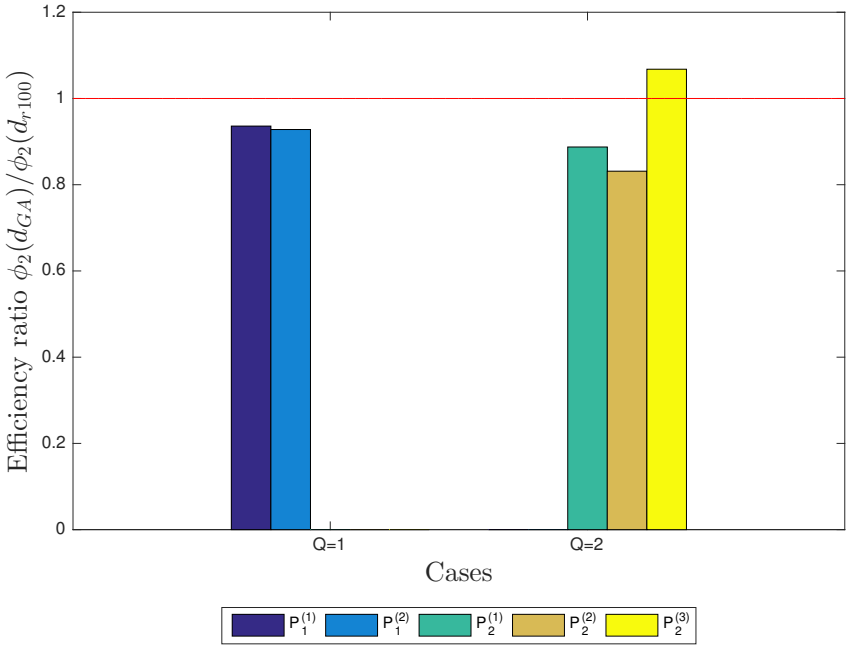

Figure 3.4: Relative design efficiencies with the A-optimality criterion for detection when $\tau_{ISI} = \tau_{TR}$: this plot provides the relative efficiency $\phi_2(\boldsymbol{d}_{GA}; 1000)/\phi_2(\boldsymbol{d}_{r100}; 1000)$ with $Q = 1, 2$ for five different cases corresponding to $\boldsymbol{P}_Q^{(c)}$ with the A-optimality.

Comparisons of CPU Time

Results in Table 3.10 show that the CPU time needed for obtaining $\boldsymbol{d}_{100}$ under $\phi_2(\cdot;100)$ is much longer than obtaining $\boldsymbol{d}_{GA}$ under ϕ_1. The results in previous subsection show that optimal designs obtained under ϕ_1 can also have good performance with respect to ϕ_2. Thus, our method for finding optimal design is recommended.

Table 3.10: CPU times (hours) for obtaining A-optimal designs that optimize $\phi_1(\boldsymbol{d}_{GA})$ and $\phi_2(\boldsymbol{d}_{r100};100)$ respectively for $Q=1,2$ for detection when $\tau_{ISI}=\tau_{TR}$.

$\boldsymbol{P}_Q^{(c)}$	$\boldsymbol{P}_1^{(I)}$	$\boldsymbol{P}_1^{(II)}$	$\boldsymbol{P}_2^{(I)}$	$\boldsymbol{P}_2^{(II)}$	$\boldsymbol{P}_2^{(III)}$
$T_{\phi_1(\boldsymbol{d}_{GA})}$	0.3557	0.3522	0.6212	0.5114	0.6172
$T_{\phi_2(\boldsymbol{d}_{r100};100)}$	30.97	27.68	14.88	26.60	29.16

The Obtained Optimal Designs

The stimulus frequencies of A-optimal designs that optimize ϕ_1 and ϕ_2 for detection problem when $\tau_{ISI}=\tau_{TR}$ are presented in Table 3.11.

Table 3.11: Stimulus frequencies of A-optimal designs for detection when $\tau_{ISI}=\tau_{TR}$: the designs are obtained by optimizing $\phi_1(\boldsymbol{d}_{GA})$ and $\phi_2(\boldsymbol{d}_{r100};100)$ with $\boldsymbol{P}_Q^{(c)}$ for $Q=1,2$.

	ϕ_1		ϕ_2	
Stimulus type (q)	1	2	1	2
$Q=1, P_1^{(I)}$	65%	-	62%	-
$Q=1, P_1^{(II)}$	71%	-	66%	-
$Q=2, P_2^{(I)}$	36%	37%	35%	35%
$Q=2, P_2^{(II)}$	32%	43%	33%	41%
$Q=2, P_2^{(III)}$	35%	41%	35%	38%

From the obtained optimal designs of both approaches, the frequency of the occurrences for the q^{th} stimulus type are different but quite close to each other. The results are similar to estimation problem when $\tau_{ISI} \neq \tau_{TR}$.

3.4.1.3 Detection Problem: Model 3.2 with $\tau_{ISI} \neq \tau_{TR}$

Lastly, we investigate the design performance with Model (3.2) under the A-optimality criterion when $\tau_{ISI} = 3\,\text{s}$ and $\tau_{TR} = 1.5\,\text{s}$. Again, we compare the same design approaches as previously demonstrated.

Design Comparisons in Terms of ϕ_1

Table 3.12 present the ϕ_1-values of optimal design obtained from our proposed approach $\boldsymbol{d}_{GA}$, Cordes *et al.* (2012)'s approach $\boldsymbol{d}_{r100}$ and some previously mentioned tradition designs. We omit the cases for $Q = 2$ using Cordes *et al.* (2012)'s approach due to time constraints. All the available results in Table 3.12 show that $\boldsymbol{d}_{GA}$ perform well among all the studied cases. Looking at Figure 3.5, we would have a similar conclusion as in Section 3.4.1.1. That is, under the ϕ_1 criterion, $\boldsymbol{d}_{GA}$ perform slightly better than $\boldsymbol{d}_{r100}$, block designs is showed to be a comparatively good design for detecting brain activity, but the poorly performed random designs and m-sequence are not recommended for the detection of brain activity.

Table 3.12: The ϕ_1-values for different designs with $Q = 1, 2$ evaluated by the A-optimality criterion for detection when $\tau_{ISI} \neq \tau_{TR}$.

Case	$\boldsymbol{d}_{GA}$	$\boldsymbol{d}_{rand}$ (mean±std)	$\boldsymbol{d}_{block}$	$\boldsymbol{d}_{mseq}$	$\boldsymbol{d}_{r100}$
$\boldsymbol{P}_1^{(I)}$	66.1678	42.6625 ± 0.0030	61.9083	42.4686	65.8431
$\boldsymbol{P}_1^{(II)}$	45.5662	30.6979 ± 0.0028	39.6666	30.7261	44.5149
$\boldsymbol{P}_2^{(I)}$	39.3002	26.5765 ± 0.0027	38.8411	26.9238	NA
$\boldsymbol{P}_2^{(II)}$	31.6220	22.2833 ± 0.0027	30.4432	22.6544	NA
$\boldsymbol{P}_2^{(III)}$	29.3871	21.2107 ± 0.0023	28.3712	21.4715	NA

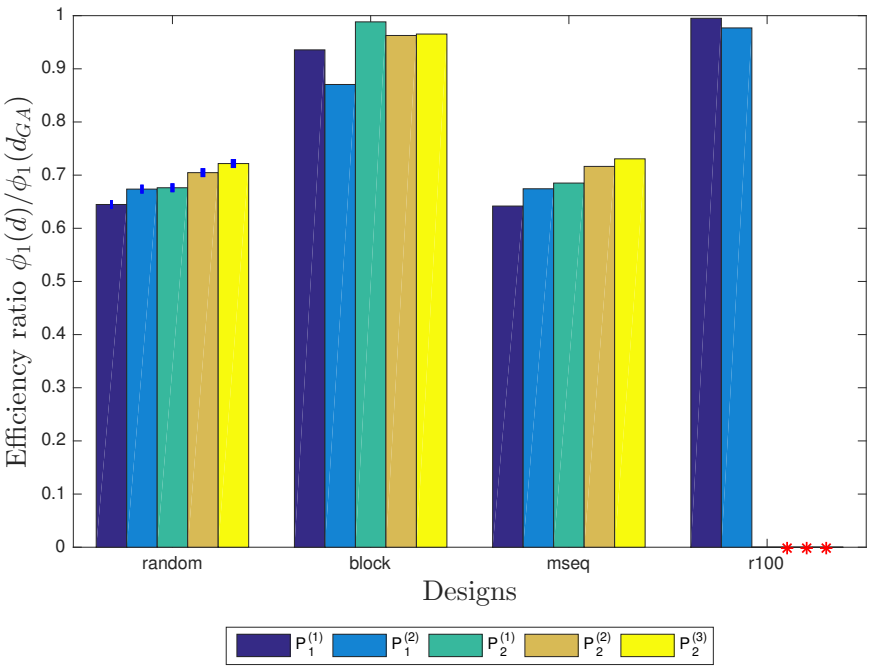

Figure 3.5: Relative design efficiencies with the A-optimality criterion for detection when $\tau_{ISI} \neq \tau_{TR}$: this plot provides the relative efficiency $\phi_1(\boldsymbol{d})/\phi_1(\boldsymbol{d}_{GA})$ of different designs $\boldsymbol{d}$ with $Q = 1$ and $Q = 2$ for five different cases corresponding to $p(r|q)$ with the A-optimality criterion (note: '*' indicates 'no result').

Design Comparisons in Terms of ϕ_2

The comparison of $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ in terms of $\phi_2(\boldsymbol{d}; 1000)$ under the A-optimality is presented in Table 3.13. It shows that $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ perform similarly under $\phi_2(\boldsymbol{d}; 1000)$. Again, we omit cases with $Q = 2$ due to time constraints.

Table 3.13: The $\phi_2(\boldsymbol{d}; 1000)$-values of $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ with $Q = 1$ under the A-optimality criterion for detection when $\tau_{ISI} \neq \tau_{TR}$.

Case	$\boldsymbol{P}_1^{(I)}$	$\boldsymbol{P}_1^{(II)}$
$\boldsymbol{d}_{GA}$	65.7330	45.1494
$\boldsymbol{d}_{r100}$	65.3080	43.8917

Comparisons of CPU Time

Table 3.14 provides the CPU time needed for obtaining $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ when $Q = 1$. The results consistently show that the CPU time needed for obtaining $\boldsymbol{d}_{GA}$ under ϕ_1 is much less than obtaining $\boldsymbol{d}_{r100}$.

Table 3.14: CPU times (hours) for obtaining A-optimal designs that optimize $\phi_1(\boldsymbol{d}_{GA})$ and $\phi_2(\boldsymbol{d}_{r100}; 100)$ respectively for $Q = 1$ for detection when $\tau_{ISI} \neq \tau_{TR}$.

Case	$\boldsymbol{P}_1^{(I)}$	$\boldsymbol{P}_1^{(II)}$
$T_{\phi_1(\boldsymbol{d}_{GA})}$	1.1097	2.1089
$T_{\phi_2(\boldsymbol{d}_{r100};100)}$	350.7778	287.8889

The Obtained Optimal Designs

The frequency of the occurrences of the optimal designs for the q^{th} stimulus type are shown in Table 3.15, for different cases when $Q = 1$ or $Q = 2$. The frequencies are similar to each other, and they tend to depend on the probability $P_Q^{(c)}$.

Table 3.15: Stimulus frequencies of A-optimal designs for detection when $\tau_{ISI} \neq \tau_{TR}$: the designs are obtained by optimizing $\phi_1(\boldsymbol{d}_{GA})$ for $Q = 1, 2$ and $\phi_2(\boldsymbol{d}_{r100}; 100)$ for $Q = 1$ with $\boldsymbol{P}_Q^{(c)}$.

	ϕ_1		ϕ_2
Stimulus type (q)	1	2	1
$Q = 1, P_1^{(I)}$	62%	-	61%
$Q = 1, P_1^{(II)}$	67%	-	64%
$Q = 2, P_2^{(I)}$	36%	35%	-
$Q = 2, P_2^{(II)}$	32%	41%	-
$Q = 2, P_2^{(III)}$	34%	40%	-

3.4.2 Evaluation Under the D-Optimality Criterion

3.4.2.1 Estimation Problem: Model 3.1 with $\tau_{ISI} \neq \tau_{TR}$

The design approaches described in Section 3.3 are studied here with Model 3.1 when $\tau_{ISI} = 3\,\text{s}$ and $\tau_{TR} = 1.5\,\text{s}$ under the D-optimality criterion.

Design Comparisons in Terms of ϕ_1

Table 3.16 and Figure 3.6 provide the performance of different designs that we studied for estimation problem when $\tau_{ISI} \neq \tau_{TR}$. These designs are evaluated by ϕ_1 under the D-optimality criterion. We omit the designs of d_{r100} when $Q = 2$ since the running time for cases with $Q = 1$ was already very long, and the CPU time for $Q = 2$ is expected to be much longer.

Table 3.16: The ϕ_1-values for different designs with $Q = 1, 2$ evaluated by the D-optimality criterion for estimation when $\tau_{ISI} \neq \tau_{TR}$.

Case	$\boldsymbol{d}_{GA}$	$\boldsymbol{d}_{rand}$ (mean±std)	$\boldsymbol{d}_{block}$	$\boldsymbol{d}_{mseq}$	$\boldsymbol{d}_{r100}$
$\boldsymbol{P}_1^{(I)}$	53.0559	46.1990 ± 0.0031	14.7006	46.7666	52.9894
$\boldsymbol{P}_1^{(II)}$	42.4447	37.1987 ± 0.0031	11.7605	37.4133	42.2007
$\boldsymbol{P}_2^{(I)}$	33.7808	30.4724 ± 0.0027	0	31.1633	NA
$\boldsymbol{P}_2^{(II)}$	30.2144	27.2447 ± 0.0027	0	27.8733	NA
$\boldsymbol{P}_2^{(III)}$	28.9257	26.0840 ± 0.0027	0	26.6845	NA

Designs obtained from our approach $\boldsymbol{d}_{GA}$ perform relatively well among all the designs that we considered as shown in Figure 3.6. Again, block designs are not recommended for estimating HRF.

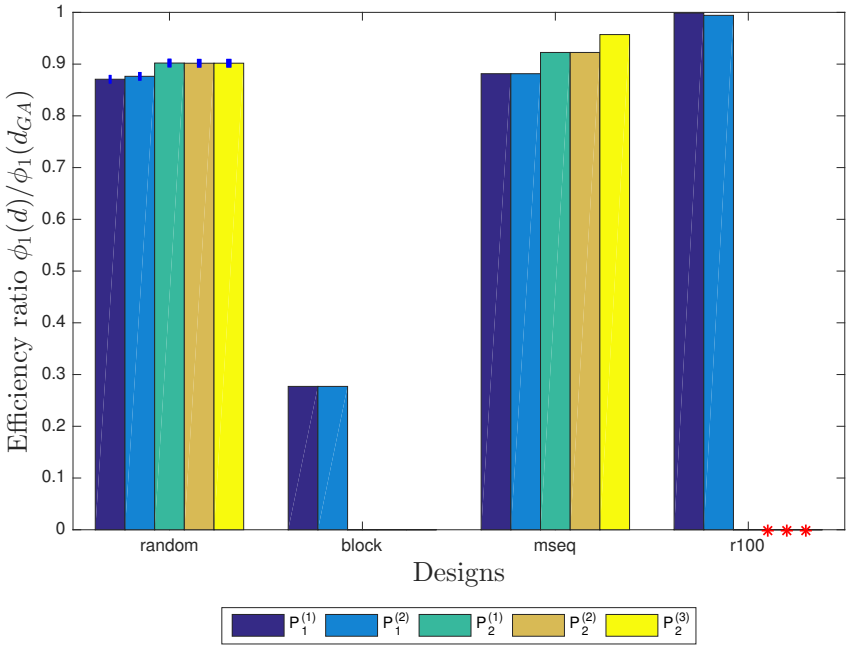

Figure 3.6: Relative design efficiencies with the D-optimality criterion for setimation when $\tau_{ISI} \neq \tau_{TR}$: this plot provides the relative efficiency $\phi_1(\boldsymbol{d})/\phi_1(\boldsymbol{d}_{GA})$ of different designs $\boldsymbol{d}$ with $Q = 1, 2$ for five different cases corresponding to $p(r|q)$ with the D-optimality (note: '*' indicates 'no result').

Design Comparisons in Terms of ϕ_2

The comparisons of $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ in terms of $\phi_2(\boldsymbol{d}; 1000)$ under the D-optimality criterion are presented in Table 3.17. It shows that $\boldsymbol{d}_{GA}$ performs slightly better than $\boldsymbol{d}_{r100}$ in terms of $\phi_2(\boldsymbol{d}; 1000)$ for $Q = 1$. We omit the simulations for obtaining optimal designs $\boldsymbol{d}_{r100}$ for $Q = 2$ since the computing time will be too long.

Table 3.17: The $\phi_2(\boldsymbol{d}; 1000)$-values of $\boldsymbol{d}_{GA}$ with $Q = 1, 2$ and $\boldsymbol{d}_{r100}$ with $Q = 1$ under the D-optimality criterion for estimation when $\tau_{ISI} \neq \tau_{TR}$.

$\boldsymbol{P}_Q^{(c)}$	$\boldsymbol{P}_1^{(I)}$	$\boldsymbol{P}_1^{(II)}$	$\boldsymbol{P}_2^{(I)}$	$\boldsymbol{P}_2^{(II)}$	$\boldsymbol{P}_2^{(III)}$
$\boldsymbol{d}_{GA}$	47.4467	37.9723	27.4013	24.4574	23.4402
$\boldsymbol{d}_{r100}$	47.0518	37.6103	NA	NA	NA

Comparisons of CPU Time

Table 3.18 provides the CPU times for the two design approaches. For $Q = 1$, the CPU time needed for obtaining $\boldsymbol{d}_{GA}$ under ϕ_1 is significantly less than obtaining $\boldsymbol{d}_{r100}$ under $\phi_2(\cdot; 100)$. For $Q = 2$ cases, our approach only need a little bit longer time comparing with the $Q = 1$ cases.

Table 3.18: CPU times (hours) for obtaining D-optimal designs that optimize $\phi_1(\boldsymbol{d}_{GA})$ with $Q = 1, 2$ and $\phi_2(\boldsymbol{d}_{r100}; 100)$ with $Q = 1$ for estimation when $\tau_{ISI} \neq \tau_{TR}$.

Case	$\boldsymbol{P}_1^{(I)}$	$\boldsymbol{P}_1^{(II)}$	$\boldsymbol{P}_2^{(I)}$	$\boldsymbol{P}_2^{(II)}$	$\boldsymbol{P}_2^{(III)}$
$T_{\phi_1(\boldsymbol{d}_{GA})}$	2.0790	1.6150	3.1581	2.4001	2.3906
$T_{\phi_2(\boldsymbol{d}_{r100};100)}$	350.7778	287.8889	NA	NA	NA

The Obtained Optimal Designs

Table 3.19 gives the stimulus frequencies of the obtained D-optimal designs, we observe that the frequency of the occurrences for the q^{th} type stimulus does not depend on the conditional probability $P_Q^{(c)}$. For example, in Table 3.19, the frequencies for $Q = 2$ are the same for all three cases, where the occurrence of stimulus type 1 is 40%, and 41% for stimulus type 2.

3.4.2.2 Detection Problem: Model 3.2 with $\tau_{ISI} = \tau_{TR}$

In this subsection, we work with Model (3.2) where the study objective is on the detection of brain activations. We set both τ_{ISI} and τ_{TR} to 2 s. We consider same design approaches as in Section 3.3 under the D-optimality criterion.

Design Comparisons in Terms of ϕ_1

Table 3.20 and Figure 3.7 present the comparison results in terms of ϕ_1. All the results in Table 3.20 suggest a good performance of the designs ($\boldsymbol{d}_{GA}$) obtained by our proposed

Table 3.19: Stimulus frequencies of D-optimal designs for estimation when $\tau_{ISI} \neq \tau_{TR}$: the designs are obtained by optimizing $\phi_1(\boldsymbol{d}_{GA})$ for $Q = 1, 2$ and $\phi_2(\boldsymbol{d}_{r100}; 100)$ for $Q = 1$ with $\boldsymbol{P}_Q^{(c)}$.

	ϕ_1		ϕ_2
Stimulus type (q)	1	2	1
$Q = 1, P_1^{(I)}$	68%	-	68%
$Q = 1, P_1^{(II)}$	68%	-	66%
$Q = 2, P_2^{(I)}$	40%	41%	-
$Q = 2, P_2^{(II)}$	40%	41%	-
$Q = 2, P_2^{(III)}$	40%	41%	-

approach, compared with tradition designs and $\boldsymbol{d}_{r100}$. We notice that the block designs also perform well for detection problem under the D-optimality criterion. However, we do not recommend to use random designs or m-sequence for detecting brain activations due to their poor performance.

Table 3.20: The ϕ_1-values for different designs with $Q = 1, 2$ evaluated by the D-optimality criterion for detection when $\tau_{ISI} = \tau_{TR}$.

Case	$\boldsymbol{d}_{GA}$	$\boldsymbol{d}_{rand}$ (mean±std)	$\boldsymbol{d}_{block}$	$\boldsymbol{d}_{mseq}$	$\boldsymbol{d}_{r100}$
$\boldsymbol{P}_1^{(I)}$	1.0658	$1.0307 \pm 3.1996 \times 10^{-4}$	1.0604	1.0304	1.0653
$\boldsymbol{P}_1^{(II)}$	1.0519	$1.0166 \pm 2.9705 \times 10^{-4}$	1.0466	1.0169	1.0514
$\boldsymbol{P}_2^{(I)}$	1.0394	$1.0064 \pm 2.6804 \times 10^{-4}$	1.0358	1.0071	1.0382
$\boldsymbol{P}_2^{(II)}$	1.0326	$1.0000 \pm 2.7801 \times 10^{-4}$	1.0290	1.0005	1.0318
$\boldsymbol{P}_2^{(III)}$	1.0300	$0.9975 \pm 2.2762 \times 10^{-4}$	1.0264	0.9980	1.0287

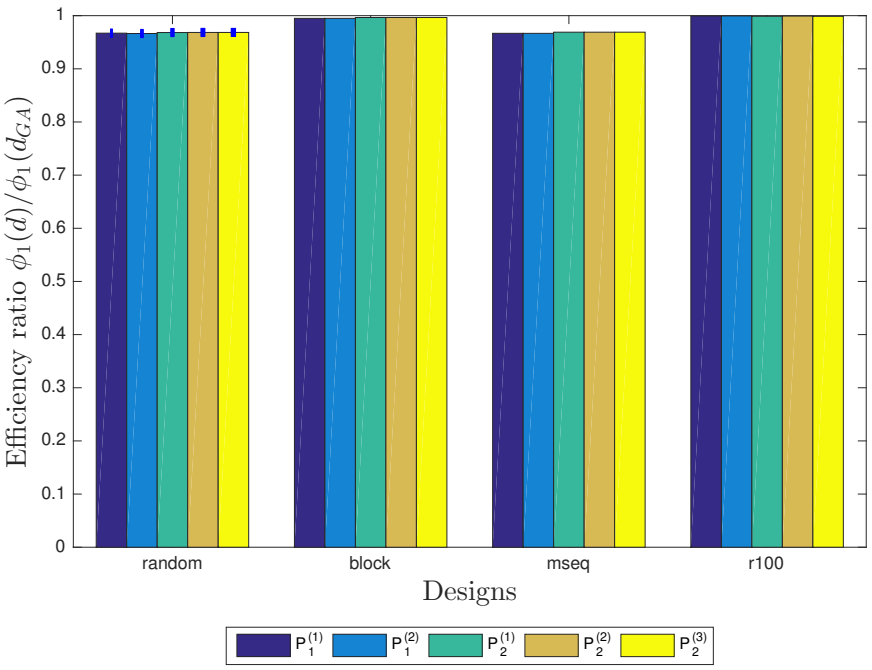

Figure 3.7: Relative design efficiencies with the D-optimality criterion for detection when $\tau_{ISI} = \tau_{TR}$: this plot provides the relative efficiency $\phi_1(\boldsymbol{d})/\phi_1(\boldsymbol{d}_{GA})$ of different designs $\boldsymbol{d}$ with $Q = 1$ and $Q = 2$ for five different cases corresponding to $p(r|q)$ with the D-optimality criterion.

Design Comparisons in Terms of $\phi_2(\boldsymbol{d}; 1000)$

Table 3.21 gives the results for comparisons between $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ in terms of $\phi_2(\boldsymbol{d}; 1000)$. All the results show that both approaches have similar performance. From Figure 3.8, we can clearly see that the optimal designs $\boldsymbol{d}_{GA}$ obtained by considering ϕ_1 slightly outperform $\boldsymbol{d}_{r100}$ when evaluated by $\phi_2(\boldsymbol{d}; 1000)$ for some cases.

Comparisons of CPU Time

Results in Table 3.22 show that the CPU time spent for obtaining $\boldsymbol{d}_{GA}$ under ϕ_1 is much less than that for obtaining $\boldsymbol{d}_{100}$ under $\phi_2(\cdot; 100)$, while the previous results showed that the optimal designs obtained under ϕ_1 can have good performance with respect to ϕ_2. Thus, our method for finding optimal design is recommended.

Table 3.21: The $\phi_2(\boldsymbol{d}; 1000)$-values of $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ with $Q = 1, 2$ under the D-optimality criterion for detection when $\tau_{ISI} = \tau_{TR}$.

$\boldsymbol{P}_Q^{(c)}$	$\boldsymbol{P}_1^{(I)}$	$\boldsymbol{P}_1^{(II)}$	$\boldsymbol{P}_2^{(I)}$	$\boldsymbol{P}_2^{(II)}$	$\boldsymbol{P}_2^{(III)}$
$\boldsymbol{d}_{GA}$	27.4802	21.9573	14.8979	13.2862	12.7324
$\boldsymbol{d}_{r100}$	27.4964	21.2327	14.9453	12.7286	12.6729

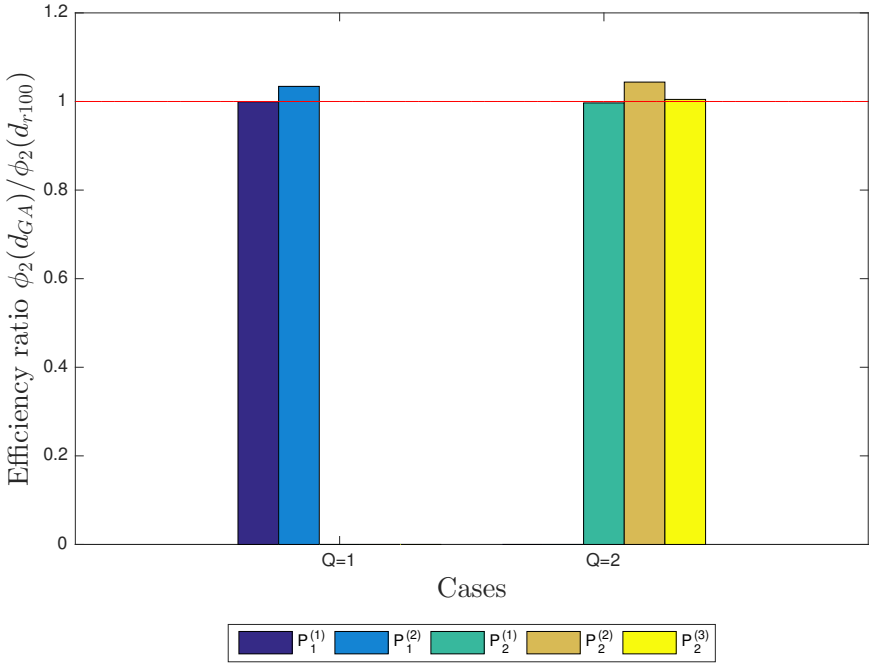

Figure 3.8: Relative design efficiencies with the D-optimality criterion for detection when $\tau_{ISI} = \tau_{TR}$: this plot provides the relative efficiency $\phi_2(\boldsymbol{d}_{GA}; 1000)/\phi_2(\boldsymbol{d}_{r100}; 1000)$ with $Q = 1, 2$ for five different cases corresponding to $\boldsymbol{P}_Q^{(c)}$ with the A-optimality.

Table 3.22: CPU times (hours) for obtaining D-optimal designs that optimize $\phi_1(\boldsymbol{d}_{GA})$ and $\phi_2(\boldsymbol{d}_{r100}; 100)$ respectively for $Q = 1, 2$ for detection when $\tau_{ISI} = \tau_{TR}$.

Case	$\boldsymbol{P}_1^{(I)}$	$\boldsymbol{P}_1^{(II)}$	$\boldsymbol{P}_2^{(I)}$	$\boldsymbol{P}_2^{(II)}$	$\boldsymbol{P}_2^{(III)}$
$T_{\phi_1(\boldsymbol{d}_{GA})}$	0.2654	0.2642	1.2479	0.9897	1.0489
$T_{\phi_2(\boldsymbol{d}_{r100};100)}$	29.6694	45.8917	48.4972	15.8506	18.9706

The Obtained Optimal Designs

The frequency of the occurrences of the q^{th} stimulus type for the obtained optimal designs are given in Table 3.23. The results indicate that the frequency of the occurrence was not affected by the conditional probability $P_Q^{(c)}$. The frequencies of optimal designs $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ are slightly different from each other.

Table 3.23: Stimulus frequencies of D-optimal designs for detection when $\tau_{ISI} = \tau_{TR}$: the designs are obtained by optimizing $\phi_1(\boldsymbol{d}_{GA})$ and $\phi_2(\boldsymbol{d}_{r100}; 100)$ with $\boldsymbol{P}_Q^{(c)}$ for $Q = 1, 2$.

	ϕ_1		ϕ_2	
Stimulus type (q)	1	2	1	2
$Q = 1, P_1^{(I)}$	66%	-	61%	-
$Q = 1, P_1^{(II)}$	66%	-	61%	-
$Q = 2, P_2^{(I)}$	39%	39%	37%	38%
$Q = 2, P_2^{(II)}$	39%	39%	38%	37%
$Q = 2, P_2^{(III)}$	39%	39%	39%	38%

3.4.2.3 Detection Problem: Model 3.2 with $\tau_{ISI} \neq \tau_{TR}$

We evaluate the design performance with Model (3.2) for detection problem when $\tau_{ISI} = 3\,\text{s}$ and $\tau_{TR} = 1.5\,\text{s}$. Similarly to the previous sections, we study performance of different design approaches.

Design Comparisons in Terms of ϕ_1

Results in Table 3.24 show that $\boldsymbol{d}_{GA}$'s perform well in all the studied cases. From Figure 3.9, we get similar results as those in Section 3.4.2.2. In terms of the ϕ_1 criterion, $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ perform similarly. For detecting brain activations, block designs are shown to be a relatively good designs, but the random designs and m-sequence do not perform as well as the others.

Table 3.24: The ϕ_1-values for different designs with $Q = 1, 2$ evaluated by D-optimality criterion for detection when $\tau_{ISI} \neq \tau_{TR}$.

Case	$\boldsymbol{d}_{GA}$	$\boldsymbol{d}_{rand}$ (mean±std)	$\boldsymbol{d}_{Block}$	$\boldsymbol{d}_{mseq}$	$\boldsymbol{d}_{r100}$
$\boldsymbol{P}_1^{(I)}$	1.0519	$1.0327 \pm 2.2129 \times 10^{-4}$	1.0482	1.0330	1.0515
$\boldsymbol{P}_1^{(II)}$	1.0413	$1.0220 \pm 2.0949 \times 10^{-4}$	1.0376	1.0226	1.0411
$\boldsymbol{P}_2^{(I)}$	1.0321	$1.0146 \pm 1.6161 \times 10^{-4}$	1.0295	1.0148	NA
$\boldsymbol{P}_2^{(II)}$	1.0269	$1.0095 \pm 1.3856 \times 10^{-4}$	1.0243	1.0097	NA
$\boldsymbol{P}_2^{(III)}$	1.0249	$1.0073 \pm 1.4866 \times 10^{-4}$	1.0223	1.0077	NA

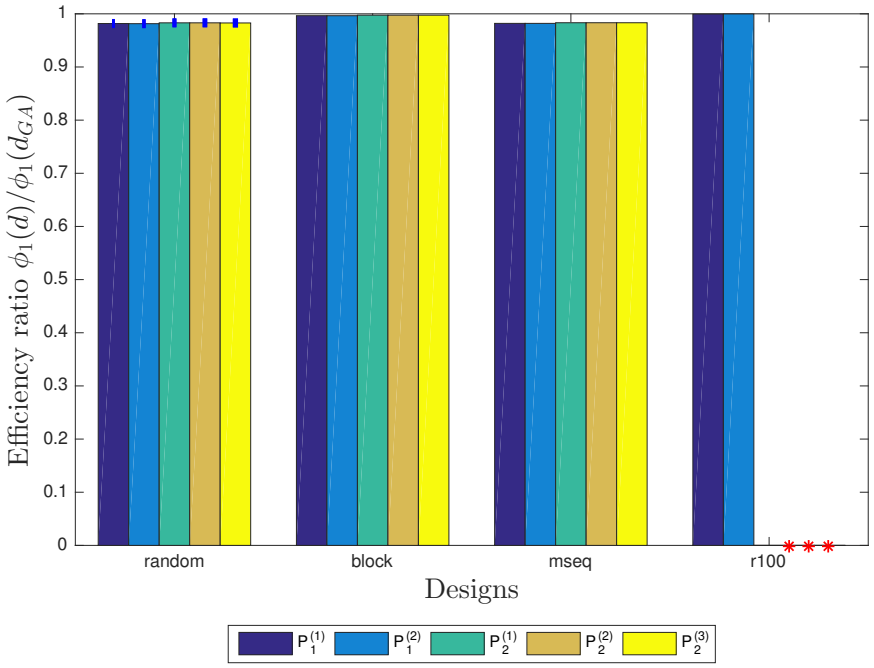

Figure 3.9: Relative design efficiencies with the D-optimality criterion for detection when $\tau_{ISI} \neq \tau_{TR}$: this plot provides the relative efficiency $\phi_1(\boldsymbol{d})/\phi_1(\boldsymbol{d}_{GA})$ of different designs $\boldsymbol{d}$ with $Q = 1$ and $Q = 2$ for five different cases corresponding to $p(r|q)$ with the D-optimality criterion (note: '*' indicates no result).

Design Comparisons in Terms of $\phi_2(\boldsymbol{d}; 1000)$

For detection problem when $\tau_{ISI} \neq \tau_{TR}$, the comparisons of $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ in terms of $\phi_2(\boldsymbol{d}; 1000)$ under the D-optimality criterion are presented in Table 3.25. It shows that $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ have similar performance in terms of $\phi_2(\boldsymbol{d}; 1000)$. Again, simulations for designs with $Q = 2$ are omitted due to the time constraints.

Table 3.25: The $\phi_2(\boldsymbol{d}; 1000)$-values of $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ with $Q = 1$ under the D-optimality criterion for detection when $\tau_{ISI} \neq \tau_{TR}$.

Case	$\boldsymbol{P}_1^{(I)}$	$\boldsymbol{P}_1^{(II)}$
$\boldsymbol{d}_{GA}$	30.7438	24.5210
$\boldsymbol{d}_{r100}$	31.8376	23.1844

Comparisons of CPU Time

Table 3.26 provides CPU times for obtaining $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ when $Q = 1$. Our approach uses significantly less CPU time. Moreover, for obtaining $\boldsymbol{d}_{r100}$ under $\phi_2(\cdot; 100)$, the time needed for $\boldsymbol{P}_1^{(II)}$ is almost doubled comparing to the time needed for $\boldsymbol{P}_1^{(I)}$, on the other hand, our approach spent similar amount of time for both cases.

Table 3.26: CPU times (hours) for obtaining D-optimal designs that optimize $\phi_1(\boldsymbol{d}_{GA})$ and $\phi_2(\boldsymbol{d}_{r100}; 100)$ respectively for $Q = 1$ for detection when $\tau_{ISI} \neq \tau_{TR}$.

Case	$\boldsymbol{P}_1^{(I)}$	$\boldsymbol{P}_1^{(II)}$
$T_{\phi_1(\boldsymbol{d}_{GA})}$	2.1381	1.7047
$T_{\phi_2(\boldsymbol{d}_{r100};100)}$	148.7556	290.5277

The Obtained Optimal Designs

Again, the frequencies of the occurrences of the q^{th} stimulus type are similar for $\boldsymbol{d}_{GA}$ and $\boldsymbol{d}_{r100}$ as shown in Table 3.27. The available results for $Q = 2$ indicate that the frequency of the occurrence does not depend on the conditional probability $P_Q^{(c)}$ under the D-optimality.

Table 3.27: Stimulus frequencies of D-optimal designs for detection when $\tau_{ISI} \neq \tau_{TR}$: the designs are obtained by optimizing $\phi_1(\boldsymbol{d}_{GA})$ and $\phi_2(\boldsymbol{d}_{r100}; 100)$ with $\boldsymbol{P}_Q^{(c)}$ for $Q = 1, 2$.

	ϕ_1		ϕ_2
Stimulus type (q)	1	2	1
$Q = 1, P_1^{(I)}$	64%	-	63%
$Q = 1, P_1^{(II)}$	64%	-	67%
$Q = 2, P_2^{(I)}$	40%	40%	-
$Q = 2, P_2^{(II)}$	40%	40%	-
$Q = 2, P_2^{(III)}$	40%	40%	-

3.4.3 An Example Similar to Cordes et al. *(2012)'s Study*

In this subsection, we present the results for the case similar to Cordes *et al.* (2012)'s Study. We provide the comparison results for in Table 3.28. The results suggest a high performance of the propose approach that optimizes $\phi_1(\boldsymbol{d})$ among the three methods. Our method requires less time comparing with the other two methods. Moreover, Cordes *et al.* (2012)'s approach does not lead to a much better design for a longer running time. For example, the designs obtained by $\phi_2(\boldsymbol{d}; 100)$ and $\phi_2(\boldsymbol{d}; 1000)$ has very close $\phi_2(\boldsymbol{d}; 1000)$-values, however, the design obtained by $\phi_2(\boldsymbol{d}; 1000)$ spent much longer computing time than that of $\phi_2(\boldsymbol{d}; 100)$; see Table 3.28.

Table 3.28: The performance of the obtained designs for the $Q = 3$ case, evaluated by $\phi_2(\cdot; \kappa = 10,000)$ and the ϕ_1, and the computing time (minutes) for obtaining the designs.

Optimality Criterion	$\phi_2(\boldsymbol{d}; 10000)$	ϕ_1	Time
$\phi_1(\boldsymbol{d}_{GA})$	42.64	44.88	106.54
$\phi_2(\boldsymbol{d}; 100)$	39.82	41.72	237.73
$\phi_2(\boldsymbol{d}; 1000)$	39.88	41.77	2041.64

3.5 Conclusion and Discussion

We propose an efficient approach to obtain robust designs for fMRI experiments when the design matrices depend not only on the selected designs, but also on the subject's probabilistic behavior during the experiments. The main idea is by considering the optimality criterion $\phi_1 \equiv \phi(\mathbb{E}[\boldsymbol{M}|\boldsymbol{d}])$. Given a design $\boldsymbol{d}$ and the conditional probability $p(r|q)$, we can view the expected information matrix as a measure of the expected amount of information about the parameters of interest in the data to be collected. A computer algorithm such as the genetic algorithm techniques can then be considered to find a design $\boldsymbol{d}$ that optimizes ϕ_1.

We demonstrate our method with both estimation and detection problems for $\tau_{ISI} = \tau_{TR}$ and $\tau_{ISI} \neq \tau_{TR}$ under the A- and D-optimality criteria, and an example similar to Cordes *et al.* (2012) through case studies. We find that our obtained designs outperform some traditional fMRI designs in all cases. We also observe that ϕ_1 provides a very good surrogate for $\phi_2 \equiv \mathbb{E}[\phi(\boldsymbol{M}|\boldsymbol{d})]$, which is also not uncommon in practice. The value of ϕ_2 is normally unavailable and needs to be approximated. One possible way is to conduct a Monte Carlo simulation to generate κ realizations of $\boldsymbol{M}$ for each $\boldsymbol{d}$, and calculate the mean of the resulting κ realizations of $\phi(\boldsymbol{M})$ as an approximation of ϕ_2. Such an approach has recently been considered by Cordes *et al.* (2012) for tackling the same design issues. We show that, with a much less CPU time than this latter approach, our method can obtain designs that perform

very well in terms of the ϕ_2-value. Our observation of cases with one or two stimulus types and two possible responses consistently show the usefulness of our proposed method. Thus, we infer further that our method will provide optimal designs for experiments with more complicated settings, such as more stimulus types and possible responses.

For the optimal designs we obtained, we observe that the frequency of the occurrences for the q^{th}-type stimulus for the A-optimal designs tend to depend on the conditional probability $p(r|q)$, that is, the frequency tends to increase when the conditional probability $p(r|q)$ moves away from the balanced case with $p(r|q) = 1/R$, where R is the number of stimulus types; on the other hand, We observe the same frequencies for cases with different $p(r|q)$ for the D-optimal designs. We further prove that the D-optimal designs remain the same regardless the value of $p(r|q)$.

Chapter 4

WAVELET-BASED LINEAR REGRESSION MODEL WITH LONG MEMORY

4.1 fMRI Time Series in Wavelet Domain

Research in fMRI is a fast growing field, new models are introduced and applied to fMRI data analysis constantly. Zarahn *et al.* (1997) suggested using long memory processes in the analysis of fMRI time series to deal with temporal autocorrelation in the error terms. Wavelet-based modeling is an advanced statistical method that has been used in the analysis of fMRI data (Ruttimann *et al.*, 1998; Brammer, 1998; Bullmore *et al.*, 2004; Costafreda *et al.*, 2009). Bullmore *et al.* (2001) suggested that resampling fMRI data in a wavelet domain is a valid alternative to autoregressive models in time domain. They observed the superiority of wavelet denoising over autoregressive (AR(1) and AR(3)) models in whitening the fMRI noise when models with the form of (2.4) are considered. Recently, Jeong *et al.* (2013) proposed a wavelet-based approach to linear regression model. They showed that discrete wavelet transforms (DWTs) can serve as whitening filters by simplifying the dense covariance matrix into a sparse one, and their approach is shown to be suitable for applications to fMRI data. Although wavelet-based models are now widely used in fMRI data analysis, there is apparently no research article on optimal design studies based on these models. Without much knowledge in choosing an optimal design for the statistical model, researchers may not be able to obtain informative data, which may render an fMRI experiment useless. The purpose of this work is to provide some guidance on optimal designs for wavelet-based models. Specifically, develop a computer algorithm to obtain optimal designs with wavelet-based linear regression models for traditional fMRI studies (without considering the subject's probabilistic behavior during the experiment). In particular, our algorithm incorporates simulated annealing and co-ordinate exchange algorithms. The main purpose of this work is to develop an algorithm that is suitable for the fMRI design problem based

on wavelet-based models. We would like to investigate the effect of several parameters in the new algorithm, and provide guidance on the selection of these parameters for a good performance.

One of the advantages of wavelet transform for fMRI data analysis is the decorrelation of the time series, which can help improve the linear model parameter estimation. Thus, understanding how wavelet transforms work is helpful in solving fMRI experimental design problems. We apply wavelet transform to epoch data set ('attention to visual motion' fMRI dataset in NIfTI file format), which is available at `http://www.fil.ion.ucl.ac.uk/spm/data/attention/`. The dataset was collected by Büchel and Friston (1997). The archive contains the smoothed, spatially normalized, realigned, slice-time corrected images. We give a brief description of the experiment as follows.

4.1.1 Data Description and Experimental Design

4.1.1.1 Image Acquisition

The experiment was performed on a 2 Tesla Magnetom VISION (Siemens, Erlangen) whole-body MRI system equipped with a head volume coil. T_2^* – weighted fMRI images ($TE = 40\,\text{ms}, TR = 3.22\,\text{s}$, and 64×64 pixels [$19.2 \times 19.2\,\text{cm}$]) were obtained with echo-planar imaging (EPI) using an axial slice orientation, with 32 contiguous slices covering the brain (slice thickness 3 mm, giving 9.6 cm vertical field of view).

Subjects were scanned during four runs of scanning periods, each lasting 5 min and 22 s. A total of 100 image volumes were acquired during each run. In each run, the first 10 scans were discarded in order to eliminate magnetic saturation effects. Thus, the length of the dataset is 360.

4.1.1.2 Attention to Visual Motion Experiment

The experiment was performed with 4 conditions, 'F': fixation, 'A': attention, 'N': no attention, 'S': stationary. A central fixation point was shown to the experimental subject as

the 'fixation' condition. For the 'attention' and 'no attention' conditions, 250 white dots emerged from central fixation point and moved towards the edge of the screen. The subjects were asked to 'detect changes' and 'just look' for the 'attention' and 'no attention' conditions, respectively. For the 'stationary' condition, the fixation point and 250 stationary dots were shown to the subject. The experimental design $\boldsymbol{d}$ for each run is $[FAFNAFNS]$, same design was repeated for 4 times to complete the experiment. More details can be found in Büchel and Friston (1997).

4.1.2 Data Analysis

We import the 'attention to visual motion' dataset using MATLAB function from Statistical Parametric Mapping (SPM), the dimensions of the data is $53 \times 63 \times 46$, the first dimension is the MRI x-direction: ear-to-ear; the second dimension is the MRI y-direction: back-to-front, the third dimension is the MRI z-direction: bottom-to-top. That is, for one single slice of the brain, the image is 53×63, and we have 46 slices. The length of the data is 360. For simplicity and for the demonstration purposes only, we consider the first 256 time points in our analysis below.

We fit a linear model (Fadili and Bullmore, 2002) to a single voxel,

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(0, \Sigma), \tag{4.1}$$

where $\boldsymbol{X}$ is the design matrix, which is given by the design $\boldsymbol{d}$ previously described. We assume $\boldsymbol{\epsilon}$ to be fractional Gaussian noise characterized by the Hurst parameter H as described in section 2.5. Thus, the covariance matrix Σ will not be a diagonal matrix.

To analyze the data, we randomly pick one slice of the brain from the dataset and estimate the Hurst parameters of every signal voxel. Figures 4.1 and 4.2 plot the Hurst estimators (H) of each voxel on slice = 18 and slice = 35, the Hurst parameters were estimated by the aggregated variance method as described in Section 2.5.3.2 from the original fMRI time series.

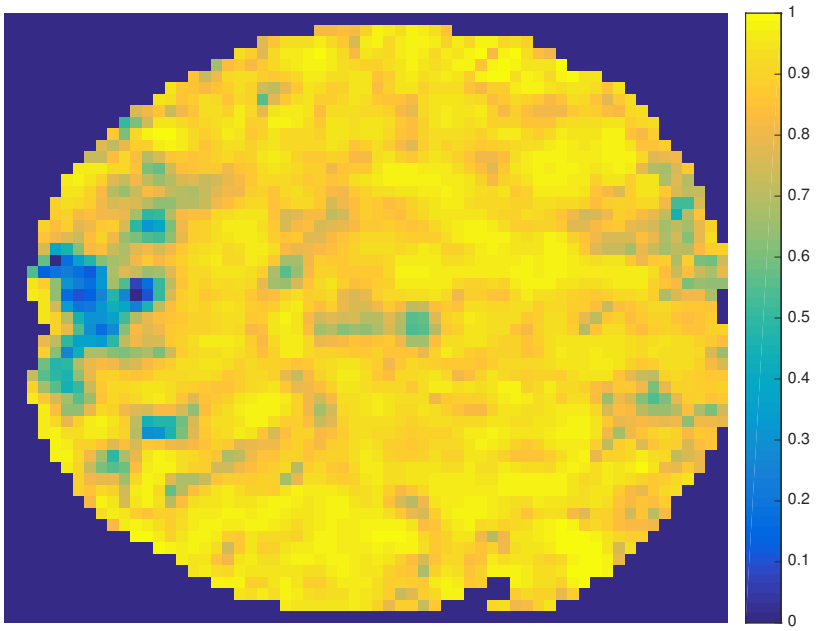

Figure 4.1: Original fMRI time series, estimation of Hurst parameters with aggregated variance method (slice 18).

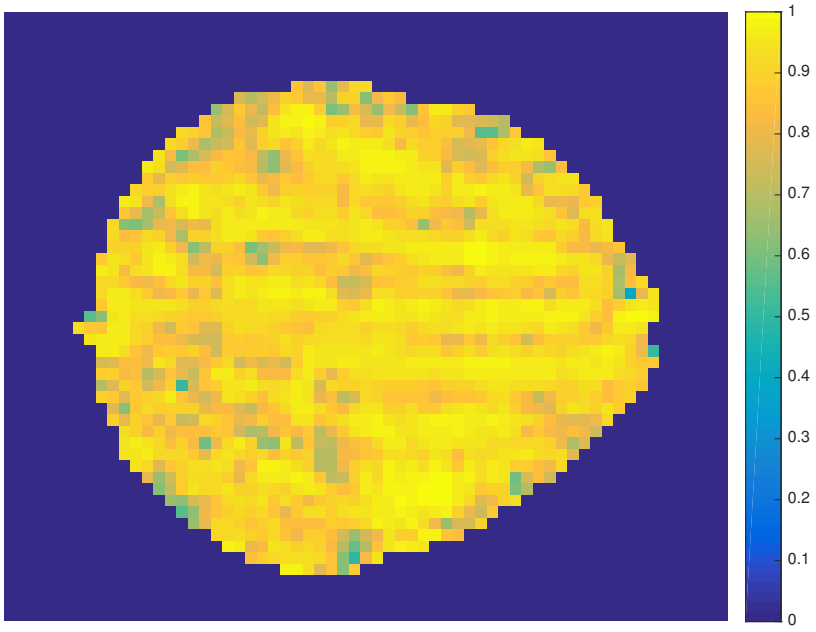

Figure 4.2: Original fMRI time series, estimation of Hurst parameters with aggregated variance method (slice 35).

We follow the tutorial from Raizada lab for detecting the edges of the brain image. Specifically, we set a threshold value to detect the brain edge. Note that this method might not be the best way for detecting the edge to our data set, but this is not our main focus in this work. From the plot, we can see that most of the values of H are between 0.5 and 1, which indicate long memory behavior.

Additionally, we fit the same fMRI time series to Model 4.1 and estimate the Hurst parameters of the error terms using aggregated variance method. The results of slice 18 and 35 of the brain are given in Figures 4.3 and 4.4. Similar to the original data, the values of Hurst parameters in Figures 4.3 and 4.4 mostly fall between 0.5 and 1.

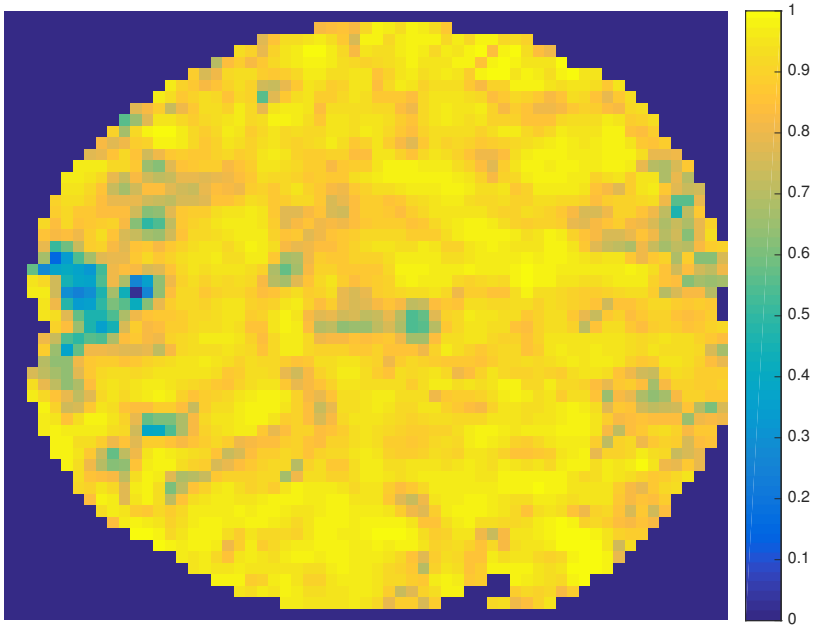

Figure 4.3: Fit the data to Model 4.1, estimation of Hurst parameters of the errors with aggregated variance method (slice 18).

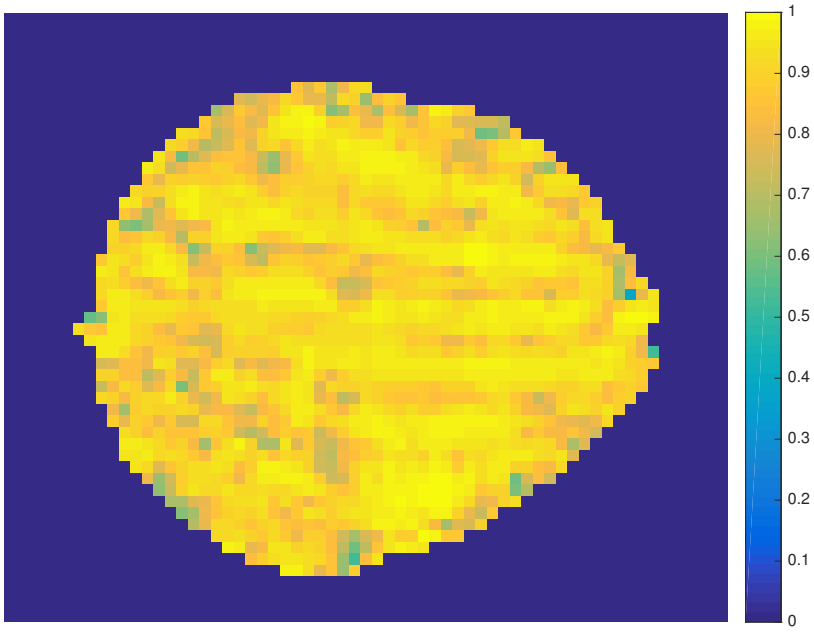

Figure 4.4: Fit the data to Model 4.1, estimation of Hurst parameters of the errors with aggregated variance method (slice 35).

4.1.3 Data Analysis in Wavelet Domain

Looking at the Hurst parameters estimated from the above fMRI data 'attention to visual motion', we indeed observed long memory, and the data are highly correlated, as a result, if we fit the data to a linear model, we will have a dense variance-covariance matrix. Now, we would like to check the variance-covariance structure of data contain long memory in wavelet domain. Because of the limited data in the experiment described above, we consider simulated random sets with specified Hurst parameter.

4.1.3.1 Simulated fMRI Data

We simulate ER-fMRI data as in Section 2.5.2. We randomly generate 100 time series, each time series has length $N = 32$, and the calculate the variance-covariance matrix of the data as shown in Figure 4.5. Next, we apply DWTs with both Haar wavelet and Daubechies 4 (db4) wavelet. Again, we calculate the variance-covariance matrix of the transformed data,

also called scaling and wavelet coefficients. Note that we omit the scaling coefficient. The results are presented in Figures 4.6 and 4.7.

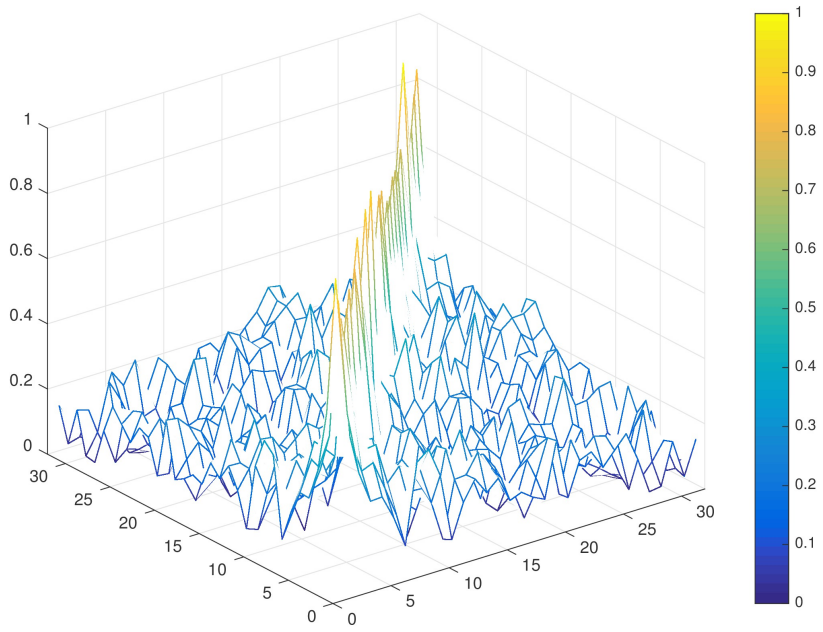

Figure 4.5: Variance-covariance matrix of simulated fMRI data by model 4.1 with long memory error (dfGn ($H = 0.75$)).

The obtained variance-covariance matrix is divided by the largest element of the matrix. From these simulated fMRI data, we can see that wavelet transforms serve as a whitening matrix, where the dense variance-covariance matrix is simplified into a sparse form. The off-diagonal elements of both variance-covariance matrices of wavelet transformed data on Figure 4.6 and 4.7 are approximately zeros. The variance-covariance matrices of wavelet coefficients are similar to the results in McCoy and Walden (1996).

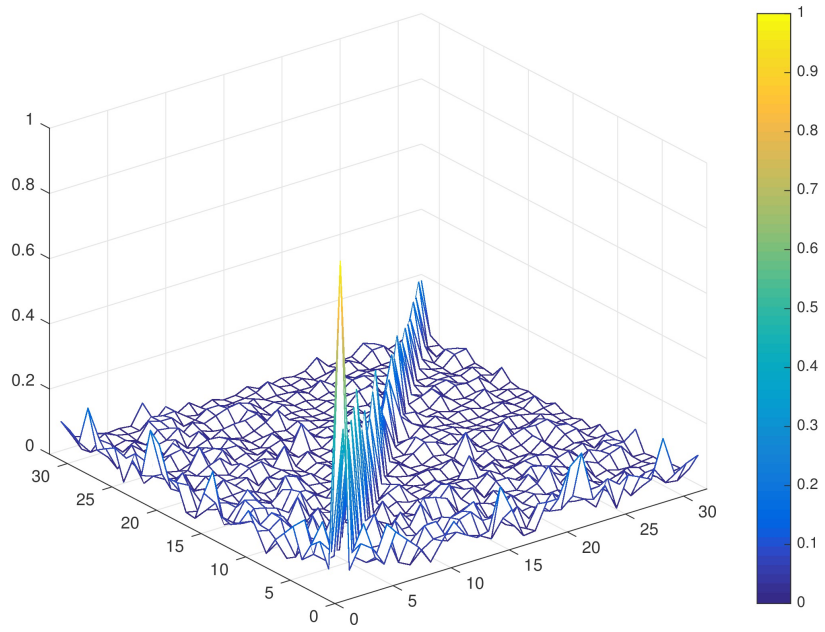

Figure 4.6: Variance-covariance matrix of wavelet coefficients from the simulated fMRI data with DWT (Haar).

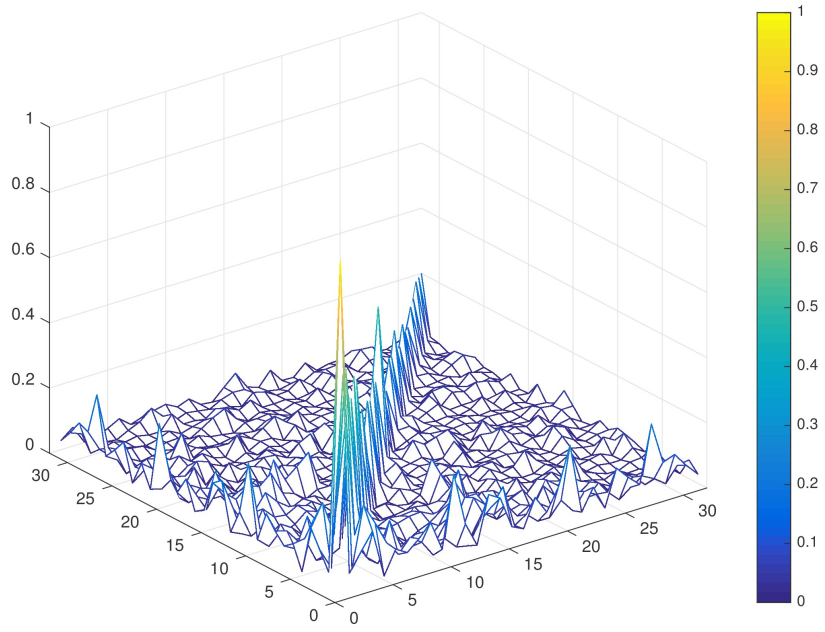

Figure 4.7: Variance-covariance matrix of wavelet coefficients from the simulated fMRI data with DWT (db4).

4.2 Methodology

The strongly correlated noise in fMRI time series can be approximately uncorrelated in wavelet domain. A closed form of the variance-covariance matrix of DWT transform data can be found in Section 2.6.2 (2.19). We now present the regression models in Section 2.5.2 in wavelet domain.

4.2.1 Wavelet-Based Linear Models

4.2.1.1 A Linear Model for Estimation

For estimating the HRF, first, the model in (2.4) can be written in centered form,

$$\boldsymbol{y} = \boldsymbol{j}\mu + \boldsymbol{X}\boldsymbol{\zeta} + \boldsymbol{\epsilon}, \tag{4.2}$$

where $\boldsymbol{j}$ is a vector of 1's, μ is the average of the observations, and the rest are as given in (2.4).

Next, the linear model in wavelet domain is obtained by taking the DWT of Model (4.2):

$$\boldsymbol{y}_w = \boldsymbol{j}_w\mu + \boldsymbol{X}_w\boldsymbol{\zeta} + \boldsymbol{\epsilon}_w, \quad \boldsymbol{\epsilon}_w \sim N(0, \Sigma_{\boldsymbol{\epsilon}_w}), \tag{4.3}$$

where $T \times 1$ vector $\boldsymbol{y}_w = W\boldsymbol{y}$, $\boldsymbol{y}$ represents the fMRI measurements obtained from a voxel every τ_{TR}. For convenience, we set $T = 2^J$; $\boldsymbol{j}_w = W\boldsymbol{j}$; $\boldsymbol{\zeta}$ is the unknown parameter vector. $\boldsymbol{X}_w = W\boldsymbol{X}$, the $T \times QK$ matrix $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_Q]$, $\boldsymbol{X}_q = [\boldsymbol{x}_{q,1}, \ldots, \boldsymbol{x}_{q,K}]$ is the $0-1$ design matrix; $\boldsymbol{\epsilon}_w = W\boldsymbol{\epsilon}$, $\boldsymbol{\epsilon}$ consists of T highly correlated errors terms that are assumed to follow a long memory process; W is the specified wavelet matrix that applies the column-wise DWT to the data. The approximated covariance matrix $\Sigma_{\boldsymbol{\epsilon}_w}$ is as described in (2.19). Define the whitening matrix

$$\boldsymbol{\Omega} = \Sigma_{\boldsymbol{\epsilon}_w}^{-1/2} = (W\Sigma_{\boldsymbol{\epsilon}}W')^{-1/2} \tag{4.4}$$

so that the covariance matrix of $\Omega\boldsymbol{\epsilon}_w$ is a diagonal matrix, $\mathrm{Cov}(\boldsymbol{\Omega}\boldsymbol{\epsilon}_w) = \sigma^2 I$.

4.2.1.2 A Linear Model for Detection

For the detection problem, we consider

$$\boldsymbol{y} = \boldsymbol{j}\mu + \boldsymbol{X}\boldsymbol{H}\boldsymbol{\theta} + \boldsymbol{\eta}, \tag{4.5}$$

where $\boldsymbol{H} = \boldsymbol{I}_Q \otimes \boldsymbol{h}^*$, $\boldsymbol{h}^*$ is defined by function (2.1). The corresponding coefficient $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_Q]$, where θ_q is defined in Model (2.3). The vector $\boldsymbol{\eta}$ is the correlated error terms that are assumed to be long memory. All the remaining terms of (4.5) are as in (4.3). After DWT, we can write (4.5) in wavelet domain, that is

$$\boldsymbol{y}_w = \boldsymbol{j}_w\mu + \boldsymbol{X}_w\boldsymbol{H}\boldsymbol{\theta} + \boldsymbol{\eta}_w, \quad \boldsymbol{\eta}_w \sim N(0, \Sigma_{\boldsymbol{\eta}_w}), \tag{4.6}$$

where $\boldsymbol{H}$ is as in Model (4.5), $\boldsymbol{\eta}_w = W\boldsymbol{\eta}$, and the variance-covariance matrix $\Sigma_{\boldsymbol{\eta}_w}$ is approximately a diagonal matrix as in (2.19). All remaining terms are as described in (4.3).

4.2.2 Information Matrix

Again, we would like to find a design that gives the most precise least square estimates of the parametric functions of interest. The goodness of designs is evaluated by a real-valued function of the information matrix $\boldsymbol{M}$. Now, for Model (4.3), we approximately have $\mathrm{Var}((y_w)_t) = C2^{md}$ with a constant C, m and d are given in (2.20), and $\mathrm{Cov}((y_w)_i, (y_w)_j) \approx 0$ for $i \neq j$, where $(y_w)_t$ is the t^{th} wavelet coefficient of the observation of the fMRI measurement. The approximate covariance matrix for $\hat{\boldsymbol{\zeta}}$ is proportional to:

$$\begin{aligned} \mathrm{Cov}(\hat{\boldsymbol{\zeta}})/\sigma^2 &= (\boldsymbol{X}'_w\boldsymbol{\Omega}'(\boldsymbol{I} - P_{\boldsymbol{\Omega}\boldsymbol{j}_w})\boldsymbol{\Omega}\boldsymbol{X}_w)^{-1} \\ &= (\boldsymbol{X}'W'\boldsymbol{\Omega}'(\boldsymbol{I} - P_{\boldsymbol{\Omega}W\boldsymbol{j}})\boldsymbol{\Omega}W\boldsymbol{X})^{-1}. \end{aligned}$$

where projection matrix $P_{\boldsymbol{\Omega}W\boldsymbol{j}} = \boldsymbol{\Omega}W\boldsymbol{j}(\boldsymbol{j}'W'\boldsymbol{\Omega}'\boldsymbol{\Omega}W\boldsymbol{j})^{-1}\boldsymbol{j}'W'\boldsymbol{\Omega}'$, $\boldsymbol{\Omega}$ is whitening matrix defined in (4.4).

Thus, we have the information matrix,

$$\boldsymbol{M}_w = \boldsymbol{X}'W'\boldsymbol{\Omega}'(\boldsymbol{I} - P_{\boldsymbol{\Omega}W\boldsymbol{j}})\boldsymbol{\Omega}W\boldsymbol{X}.$$

Let $\boldsymbol{A}_w = W'\boldsymbol{\Omega}'(\boldsymbol{I} - P_{\boldsymbol{\Omega}W\boldsymbol{j}})\boldsymbol{\Omega}W$, then we have $\boldsymbol{M}_w = \boldsymbol{X}'\boldsymbol{A}_w\boldsymbol{X}$. We would like to find a $\boldsymbol{d}$ that optimizes $\phi(\boldsymbol{M})$ for some ϕ as described in Chapter 3.

For the detection problem, the information matrix for Model (4.6) is

$$\boldsymbol{M}_w = \boldsymbol{H}'\boldsymbol{X}'W'\boldsymbol{\Omega}'(\boldsymbol{I} - P_{\boldsymbol{\Omega}W\boldsymbol{j}})\boldsymbol{\Omega}W\boldsymbol{X}\boldsymbol{H},$$

where $\boldsymbol{H}$ is as defined in (4.5).

4.2.3 Design Selection Criterion

To compare with Saleh *et al.* (2017), we apply the same criterion, D-optimality criteria ϕ:

$$\phi(\boldsymbol{M}) = det(\boldsymbol{M})^{1/\mathcal{R}}, \tag{4.7}$$

where $\boldsymbol{M}$ is nonsingular. We set $\phi(\boldsymbol{M}) = 0$ when $\boldsymbol{M}$ is singular, $\mathcal{R}$ is set to QK for Model (4.3), and Q for Model (4.5) which corresponds to the number of the parameters of interest.

4.3 Searching Algorithm

Saleh *et al.* (2017) proposed an algorithm for finding D-optimal design for fMRI experiment by combining hill climbing algorithm and exchange algorithm. Compared to genetic algorithm, their algorithms are superior in terms of computing time and design efficiency. Inspired by their work, we consider an algorithm that incorporates concepts of simulated annealing and the exchange algorithms for wavelet-based linear models.

To find a D-optimal design, the algorithm of Saleh *et al.* (2017) starts with a random design $\boldsymbol{d} = \{d_1, d_2, \ldots, d_N\}$. It then sequentially perturbs one element at a time of the design, which is d_n, where $n = 1, \ldots, N$. Specifically, with Q stimulus types, d_n is replaced by an integer from $\{0, 1, \ldots, Q\} - \{d_n\}$. At each step, there are Q candidate designs, and the integer that yields the highest design efficiency is used to update the design. The pseudocode for exchange algorithm is shown in Algorithm 3. Hill climbing algorithm is then

applied to obtain a optimal design, that is, the updated design is accepted if the change improves the value of ϕ.

The process is repeated until no further improvement can be achieved. To find the optimal design, this algorithm requires to have certain number of initial designs. This is sometimes viewed as a disadvantage of such a hill climbing algorithm, in which a candidate design is only accepted if it is better than the current design. It may thus be easily trapped in a local optimum solution, and the quality of the obtained design will depend on the initial design. One of the methods to avoid trapping in local optimum is to repeat algorithm many times with different initials. On the other hand, simulated annealing algorithms, which allow to accept a worse design with certain probability during the search, can escape the local optimum and reach a global optimum.

Algorithm 3 Exchange algorithm (Saleh *et al.*, 2017)

```
1 d ← random design of length N
2 for n = 1 to N do
3     q = d(n)                                   ▷ n^th element of design d
4     for j = 0 to Q except for q do
5         d_candidate ← d : d(n) = j             ▷ change the n^th element of d into j
6         Calculate φ(d_candidate)
7     end for
8     Find max or min of φ-values and obtain the updated design d_try
9 end for
```

We now present some details about our proposed algorithm for wavelet-based linear model. To maximize the optimality criterion over the design space, the algorithm begins with an initial design $\boldsymbol{d}$, which can be any design such as a sequence of zeros, or a randomly generated sequence. Based on $\boldsymbol{d}$, we calculate its design matrix $\boldsymbol{X}$ and the value of the D-optimality criterion ϕ. We adapt Algorithm 3 to define our candidate designs, and denote the 'better' candidate (i.e. the one with the larger ϕ) by $\boldsymbol{d}_{try}$. This design is accepted when it improves the value of ϕ. When $\boldsymbol{d}_{try}$ is worse than the current design $\boldsymbol{d}$, we may still accept it with probability

$$p = exp(\Delta\phi/T), \tag{4.8}$$

where $\Delta\phi = \phi(\boldsymbol{d}_{try}) - \phi(\boldsymbol{d})$, and T is a parameter of the algorithm called 'temperature'. When T is large, the probability of accepting a worse design is high (close to 1). Along the search, we gradually decrease the value of T, the probability of accepting a worse design is thus lower and lower. The algorithm keeps track of the best design during the search, it is denoted by $\boldsymbol{d}_{best}$. After a fixed number of iterations, if design $\boldsymbol{d}$ is updated, we lower the temperature by multiplying a constant reduction factor R and continue the search, otherwise, the algorithm is terminated to give the current best design $\boldsymbol{d}_{best}$. An pseudocode can be found in Algorithm 4. Next, we run several simulations to demonstrate the performance of the proposed algorithm and provide some guideline on the selection of parameter values.

4.4 Stimulation Studies

For comparison purposes, we also adapt the algorithm of Saleh *et al.* (2017) to search for optimal D-optimal designs with wavelet-based linear models for both estimation and detection problems. We consider cases with $Q = 1, 2, 3$. The length of designs are all set to $2^8 = 256$. The inter-stimulus-interval and the time-to-repetition (τ_{ISI}, τ_{TR}) are set to $(2\,\text{s}, 2\,\text{s})$ for $\tau_{ISI} = \tau_{TR}$, and $(3\,\text{s}, 1.5\,\text{s})$ for $\tau_{ISI} \neq \tau_{TR}$. We assume that the noise follows a long memory process; specifically, we consider a discrete fractional Gaussian noise with Hurst parameter $H = 0.75$. For all the cases, both the previously described algorithm and that of Saleh *et al.* (2017) are employed for obtaining optimal designs. Originally, Saleh *et al.* (2017)'s approach is designed for another model that involves an AR(1) noise. We modify the corresponding part in their code to suit our problem. To model the variance-covariance matrix, we choose Daubechies wavelet with 4 vanish moments (db4) (Daubechies, 1992) as wavelet matrix which is considered in Jeong *et al.* (2013).

In addition, we compare our obtained designs with two traditional designs, they are 16 s-on-16 s-off block design $\boldsymbol{d}_{block}$, as described in Chapter 3; and m-sequence based design $\boldsymbol{d}_{mseq}$. With $N = 256$, we generate m-sequence based designs as follows,

Algorithm 4 Our proposed algorithm

Initialization:
T_0: starting temperature ($T_0 = 1$);
R: reduction factor for temperature T;
$\boldsymbol{d}_{ini}$: a random design of length N;
$T = T_0$; $\boldsymbol{d} = \boldsymbol{d}_{ini}$;
$\boldsymbol{d}_{best} = \boldsymbol{d}_{ini}$; $\phi_{best} = \phi(\boldsymbol{d}_{best})$;
while STOP = False **do**
    STOP = True
    for $i = 1$ to N **do**
        calculate $\phi(\boldsymbol{d})$
        adapt Algorithm 3, find $\boldsymbol{d}_{try}$ such that $\phi_{max} = arg\,max(\phi(\boldsymbol{d}_{candidate}))$
        $\Delta\phi = \phi_{max} - \phi(\boldsymbol{d}) = \phi(\boldsymbol{d}_{try}) - \phi(\boldsymbol{d})$
        if $\phi_{max} \neq 0$ **then**
            if $\Delta\phi \geq 0$ **then**
                $\boldsymbol{d} = \boldsymbol{d}_{try}$ ▷ accept the improvement
                STOP = False
            else
                $p = exp(\frac{\Delta\phi}{T})$
                if $p > rand(0, 1)$ **then**
                    $\boldsymbol{d} = \boldsymbol{d}_{try}$ ▷ accept the worsening
                    STOP = False
                end if
            end if
        else
            $p = exp(-\frac{1}{T})$
            if $p > rand(0, 1)$ **then**
                $\boldsymbol{d} = \boldsymbol{d}_{try}$ ▷ accept the singular
                STOP = False
            end if
        end if
        if $\phi_{max} \neq 0$ and $\phi_{max} > \phi_{best}$ **then**
            $\phi_{best} = \phi_{max}$
            $\boldsymbol{d}_{best} = \boldsymbol{d}_{try}$ ▷ keep track of best design $\boldsymbol{d}_{best}$
        end if
    end for
    if STOP == False **then**
        $T = T \times R$
    end if
end while

Case 1: For $Q = 1, 3$, first, we generate m-sequence with length $2^8 - 1$ for $Q = 1$, and length $4^4 - 1$ with $Q = 3$, then we find the longest run of zeros, and add one more zero to obtain a m-sequence design with $N = 2^8$.

Case 2: For $Q = 2$, first, we generate m-sequence with length $3^5 - 1$, then we repeat the same sequence once to form a sequence with length 484, we select first 256 elements as a m-sequence based design.

As mentioned before, to avoid being caught at a local optimum, Saleh *et al.* (2017)'s algorithm needs to have a sufficient number of initial designs. With more initial designs, their algorithm will have a higher chance to give a better design. As for the newly proposed algorithm, it is crucial to pick an appropriate value for reduction factor R. To investigate the trade offs between computing time and design efficiency of both algorithms, we run the simulations with different number of initial designs for Saleh *et al.* (2017)'s algorithm; and for the new algorithm, we consider various values of reduction factor R.

Estimating the Hurst parameter of a noisy fMRI data is quite challenging. Sometimes, we might use a wrong Hurst parameter in the model for obtaining optimal designs. We are interested in the performance of the optimal designs obtained from wrong model assumption. We run the simulation with different value of Hurst parameter, and evaluate the optimal designs with 'true' Hurst parameter to investigate the robustness of the optimal designs to the Hurst parameter.

4.5 Results

We demonstrate the performance of our proposed algorithm with the following simulations.

4.5.1 Evaluation Under the D-Optimality Criterion

We evaluate Saleh *et al.* (2017)'s algorithm and the newly proposed algorithm with the D-optimality criterion for Models 4.3 and 4.6 when $\tau_{ISI} = \tau_{TR}$ and $\tau_{ISI} \neq \tau_{TR}$. We then

compare the optimal designs obtained from each algorithm together with some tradition designs including m-sequence based design and block design.

4.5.1.1 Estimation Problem with $\tau_{ISI} = \tau_{TR}$

Figure 4.8 presents some features of our algorithm. In this example, we set the reduction factor $R = 0.96$ and the Hurst parameter $H = 0.75$ to search the D-optimal design with 2 stimulus types for estimation problem when $\tau_{ISI} = \tau_{TR}$.

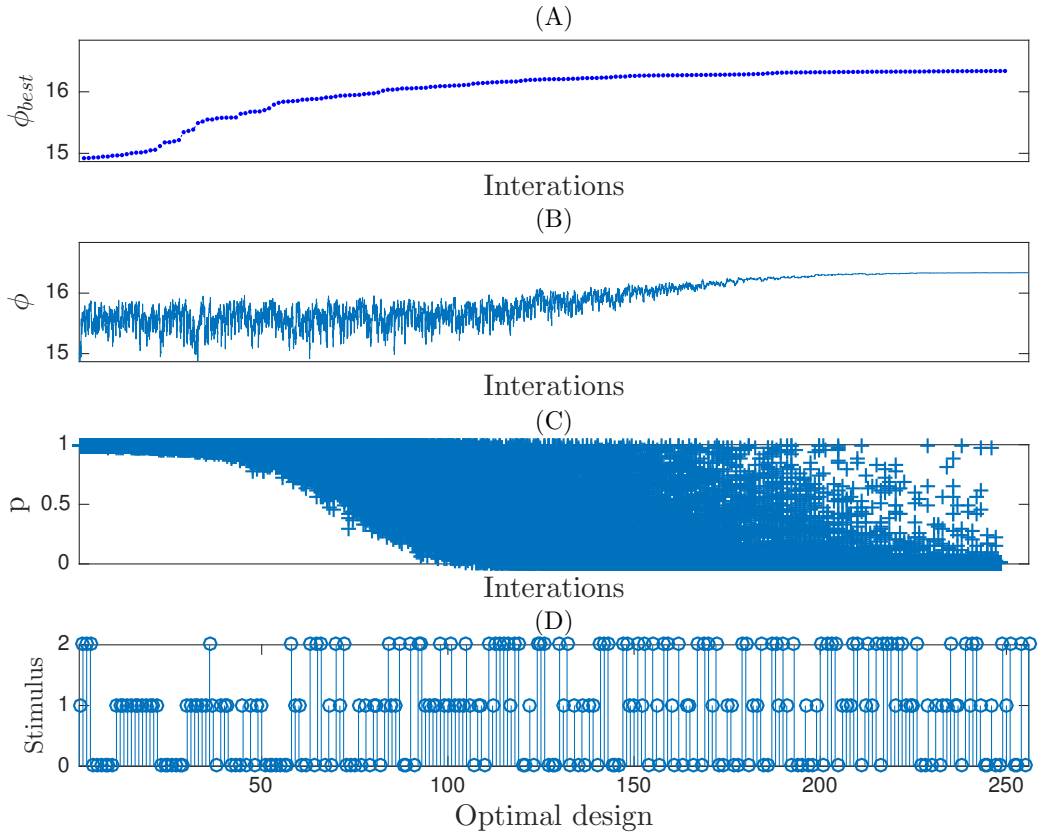

Figure 4.8: Simulation process of the newly proposed algorithm for estimation problem when $\tau_{ISI} = \tau_{TR}$ with $Q = 2$, the reduction factor $R = 0.96$. Plots from top to bottom: (A) The record of best ϕ-values during the entire search for the optimal design; (B) ϕ-values for each iteration during the searching process; (C) Probability for accepting a worsen design; (D) The optimal design obtained.

Plot (A) in Figure 4.8 gives the best ϕ-values obtained achieved over the iterations of our algorithm. As excepted, at the beginning of the search, the ϕ values may go up or down as shown in the plot (B) of Figure 4.8. At this stage, the algorithm tends to explore the entire design space and there is a high probability to accept a worsen design; see also plot (C) of Figure 4.8, where the probability of accepting a worse design over iterations is presented. We can see a significant improvement at the early stage of the search, and we then slowly approach the optimal solution. Plot (D) shows the optimal design we obtained from our new algorithm. We observe similar performance of the new algorithm for estimation problem with $\tau_{ISI} \neq \tau_{TR}$, thus, we omit the results.

Tables 4.1, 4.2 and 4.3 present the D-optimality criterion values of the optimal designs (ϕ_{best}) and computing time of Saleh *et al.* (2017)'s algorithm and the new algorithm for $Q = 1, 2, 3$ for estimating the HRF. With similar CPU time, the optimal designs obtained from the new algorithm tend to perform slightly better than those obtained by Saleh *et al.* (2017)'s algorithm. Additionally, comparing to Saleh *et al.* (2017)'s algorithm, for achieving similar design efficiencies, the new algorithm tends to require a less CPU time than the algorithm of Saleh *et al.* (2017).

To compare the performance of the algorithms, we run the simulations with 10, 25, 50, 75, 100, 250, 500 and 1000 initial designs for Saleh *et al.* (2017)'s algorithm and the new algorithm has a reduction factor of $R = 0.7, 0.75, 0.8, 0.85, 0.9, 0.96, 0.97$ and 0.98. For the former algorithm, we did not observe a significant improvement by using a large number of initials designs. As for the new algorithm, better designs are obtained when we have a bigger reduction factor R; see also Figure 4.9, which presents the achieved D-optimal criterion values over the iterations of the proposed algorithm for obtaining optimal designs with $Q = 1$. The reduction factor is set to $R = [0.7, 0.75, 0.8, 0.9, 0.95]$. Despite the 'badness' of the randomly generated initial design, our proposed algorithm reaches to a good design after certain number of iterations. We also observe that the designs obtained from both algorithms perform better than m-sequence based design and block design.

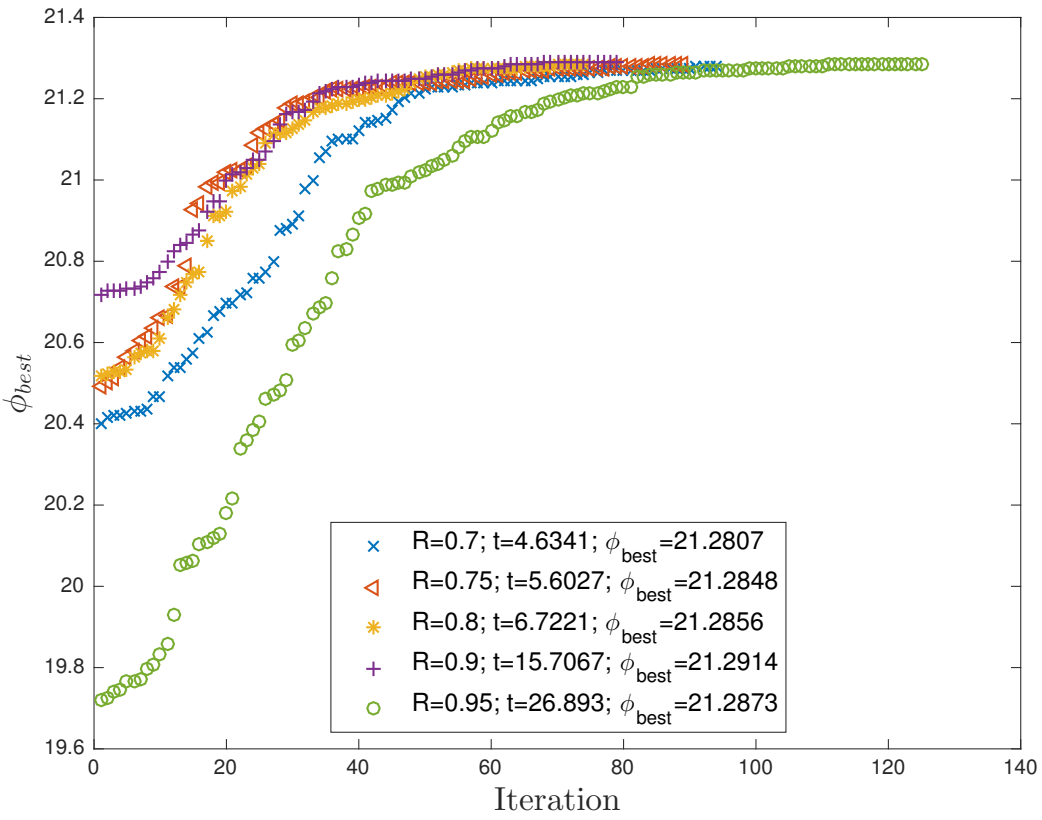

Figure 4.9: The achieved D-optimal criterion values (ϕ_{best}) over iterations of proposed algorithm for reduction factor $R = [0.7, 0.75, 0.8, 0.9, 0.95]$ and $Q = 1$.

To investigate the effect of the value of the Hurst parameter on the obtained optimal designs, we run the simulations with Hurst parameter $H = [0.6, 0.75, 0.9]$. With the optimal design obtained from each simulation, we then evaluate the designs with a 'true' criterion $\phi(H)$, where $H = [0.6, 0.75, 0.9]$, the relative efficiencies is given in Table 4.4. From the results, the new algorithm obtained optimal designs with similar performance over the different values of the Hurst parameter. This suggests that the optimal designs obtained from the new algorithm is robust to a misspecified Hurst parameters in the model. Having slightly inaccurate information about the value of the Hurst parameter does not seem to have a serious consequence on the performance of the obtained optimal designs.

Table 4.1: The performance of the optimal designs for estimation obtained from Saleh *et al.* (2017)'s algorithm (old algorithm) with I initial designs, our proposed algorithm (new algorithm) with R reduction factor, block design and m-sequence based design, and the their computing time (seconds): $Q = 1$ and $\tau_{ISI} = \tau_{TR}$.

Old algorithm			New algorithm		
I	Time (s)	ϕ_{best}	R	Time (s)	ϕ_{best}
10	7.1860	21.2731	0.70	4.6341	21.2807
25	13.9154	21.2861	0.75	5.6027	21.2848
50	30.9679	21.2902	0.80	6.7221	21.2856
75	45.1091	21.2905	0.85	9.1236	21.2853
100	66.1904	21.2818	0.90	15.7067	21.2914
250	169.3016	21.2933	0.96	35.0616	21.2944
500	310.0266	21.2921	0.97	45.5945	21.2971
1000	573.3169	21.2962	0.98	65.1150	21.3031
Tradition designs					
$\phi_{best}(\boldsymbol{d}_{block})$		0.46838			
$\phi_{best}(\boldsymbol{d}_{mseq})$		21.0837			

4.5.1.2 Estimation Problem with $\tau_{ISI} \neq \tau_{TR}$

Tables 4.5, 4.6 and 4.7 provide the ϕ values with the D-optimality of the optimal designs that were found by Saleh *et al.* (2017)'s algorithm and our algorithm. The computing time needed for obtaining these designs are also provided there. For comparison purposes, Saleh *et al.* (2017)'s algorithm is tested several times with different number of initial designs, $I = 10, 25, 50, 75, 100, 250, 500, 1000$. We repeated our proposed algorithm with different values for the reduction factor $R = 0.7, 0.75, 0.8, 0.85, 0.9, 0.96, 0.97, 0.98$. Similarly to the cases with $\tau_{ISI} = \tau_{TR}$, we see that, with similar computing time, our algorithm can obtain

Table 4.2: The performance of the optimal designs for estimation obtained from Saleh *et al.* (2017)'s algorithm (old algorithm) with I initial designs, our proposed algorithm (new algorithm) with R reduction factor, block design and m-sequence based design, and the their computing time (seconds): $Q = 2$, $\tau_{ISI} = \tau_{TR}$.

Old algorithm			New algorithm		
Initials	Time (s)	ϕ_{best}	R	Time (s)	ϕ_{best}
10	24.1112	16.3079	0.70	15.2565	16.3096
25	58.9485	16.3033	0.75	19.9904	16.3233
50	118.3059	16.3025	0.80	24.7501	16.3171
75	184.2289	16.3204	0.85	30.8919	16.3298
100	243.0310	16.3256	0.90	47.4077	16.3369
250	609.2890	16.3229	0.96	113.9497	16.3380
500	1.2271×10^3	16.3187	0.97	154.3477	16.3419
1000	2.5437×10^3	16.3289	0.98	227.8445	16.3391
Tradition designs					
$\phi_{best}(\boldsymbol{d}_{block})$		0			
$\phi_{best}(\boldsymbol{d}_{mseq})$		16.0519			

statistically more efficient designs than designs achieved by Saleh *et al.* (2017)'s algorithm. In addition, our algorithm can obtain statistically equivalent design with a much less computing time. By increasing number of initial designs, the computing time of Saleh *et al.* (2017)'s algorithm increases linearly, however, the achieved design efficiency is not significantly improved. On the other hand, with a higher reduction factor for our algorithm, although the computing time increases, it also gives a better optimal design. Designs obtained from both algorithms perform better than m-sequence based design and block design.

Table 4.3: The performance of the optimal designs for estimation obtained from Saleh *et al.* (2017)'s algorithm (old algorithm) with I initial designs, our proposed algorithm (new algorithm) with R reduction factor, block design and m-sequence based design, and the their computing time (seconds): $Q = 3$, $\tau_{ISI} = \tau_{TR}$.

Old algorithm			New algorithm		
Initials	Time (s)	ϕ_{best}	R	Time (s)	ϕ_{best}
10	66.2604	13.1973	0.70	32.9317	13.2439
25	158.9035	13.2384	0.75	46.3659	13.2594
50	310.0428	13.2187	0.80	55.0119	13.2491
75	459.6314	13.2322	0.85	70.0541	13.2824
100	619.7323	13.2218	0.90	101.1490	13.2885
250	1.5532×10^3	13.2205	0.96	264.6488	13.2960
500	3.1586×10^3	13.2333	0.97	342.4683	13.2931
1000	6.6084×10^3	13.2497	0.98	517.1954	13.2913
Tradition designs					
$\phi_{best}(\boldsymbol{d}_{block})$		0			
$\phi_{best}(\boldsymbol{d}_{mseq})$		12.9442			

Table 4.4: Robustness of the designs obtained by the proposed algorithm with misspecified Hurst parameter for estimation When $\tau_{ISI} = \tau_{TR}$.

Q	1			2			3		
$\boldsymbol{d}^*(H = \cdot)$	0.9	0.75	0.6	0.9	0.75	0.6	0.9	0.75	0.6
$\phi(0.9)$	1	0.998	0.991	1	0.996	0.989	1	0.997	0.988
$\phi(0.75)$	0.996	1	0.999	0.995	1	0.997	0.992	1	0.999
$\phi(0.6)$	0.991	0.998	1	0.987	0.996	1	0.982	0.996	1

Table 4.5: The performance of the optimal designs for estimation obtained from Saleh *et al.* (2017)'s algorithm (old algorithm) with I initial designs, our proposed algorithm (new algorithm) with R reduction factor, block design and m-sequence based design, and the their computing time (seconds): $Q = 1$, $\tau_{ISI} \neq \tau_{TR}$.

Old algorithm			New algorithm		
I	Time (s)	ϕ_{best}	R	Time (s)	ϕ_{best}
10	24.7688	24.0313	0.70	13.3208	24.1598
25	65.4386	24.0844	0.75	16.7258	24.1633
50	123.8135	24.0663	0.80	20.3398	24.1556
75	175.6222	24.0352	0.85	27.6570	24.1591
100	230.4629	24.0481	0.90	45.1774	24.1646
250	563.8479	24.0232	0.96	108.9030	24.1587
500	1.1369×10^3	24.1333	0.97	151.8808	24.1596
1000	2.2452×10^3	24.0978	0.98	219.3428	24.1791
Tradition designs					
$\phi_{best}(\boldsymbol{d}_{block})$		0.5740			
$\phi_{best}(\boldsymbol{d}_{mseq})$		23.9369			

We also study the robustness of optimal design obtained by our algorithm under misspecified Hurst parameter. The results for $Q = 1, 2, 3$ with $H = 0.6, 0.75$, and 0.9 are presented in Table 4.8. The relative efficiencies are all very close to 1, the performance of the obtained design is better if the specified value of the Hurst parameter is closer to the 'true' one. For example, when $Q = 1$ and the 'true' Hurst parameter H is 0.1, the optimal design obtained under $H = 0.75$ is better than the optimal design obtained under $H = 0.6$ when these designs are evaluated with $H = 0.9$. But, again, all the relative efficiencies are close to 1. Thus, we say that our designs are robust under misspecified Hurst parameter.

Table 4.6: The performance of the optimal designs for estimation obtained from Saleh *et al.* (2017)'s algorithm (old algorithm) with I initial designs, our proposed algorithm (new algorithm) with R reduction factor, block design and m-sequence based design, and the their computing time (seconds): $Q = 2$, $\tau_{ISI} \neq \tau_{TR}$).

Old algorithm			New algorithm		
I	Time (s)	ϕ_{best}	R	Time (s)	ϕ_{best}
10	77.5988	17.5335	0.70	57.5101	17.6993
25	182.0149	17.6058	0.75	63.3740	17.7015
50	366.8843	17.6044	0.80	82.8099	17.7030
75	518.8944	17.6201	0.85	108.0507	17.6999
100	735.1227	17.6136	0.90	161.4309	17.7017
250	1780.9	17.6528	0.96	418.0875	17.7061
500	3.5397×10^3	17.5964	0.97	536.0883	17.7047
1000	7.1772×10^3	17.5234	0.98	817.4659	17.7083
Tradition designs					
$\phi_{best}(\boldsymbol{d}_{block})$		0			
$\phi_{best}(\boldsymbol{d}_{mseq})$		17.4636			

4.5.1.3 Detection Problem with $\tau_{ISI} = \tau_{TR}$

In this section, we study the detection problem, the aim is at obtaining a design that helps to provide precise inference on detecting the activated regions of the brain in response to stimuli.

Figure 4.10 presents the process of our algorithm in search of optimal design for detection problem when $\tau_{ISI} = \tau_{TR}$, we set the reduction factor $R = 0.96$ and the Hurst parameter $H = 0.75$ to search the D-optimal design with 2 stimulus types. We observe the similar features of the algorithm for detection problem when $\tau_{ISI} \neq \tau_{TR}$, and thus omit the results

Table 4.7: The performance of the optimal designs for estimation obtained from Saleh *et al.* (2017)'s algorithm (old algorithm) with I initial designs, our proposed algorithm (new algorithm) with R reduction factor, block design and m-sequence based design, and the their computing time (seconds): $Q = 3$, $\tau_{ISI} \neq \tau_{TR}$).

Old algorithm			New algorithm		
I	Time (s)	ϕ_{best}	R	Time (s)	ϕ_{best}
10	170.8543	13.9787	0.70	106.3306	14.1563
25	370.5744	13.9351	0.75	140.5539	14.5960
50	8.0015×10^2	13.9950	0.80	186.6411	14.1571
75	1.1979×10^3	13.9605	0.85	280.9567	14.1734
100	1.6435×10^3	14.0497	0.90	352.533	14.1751
250	4.0718×10^3	14.0014	0.96	889.4479	14.1786
500	7.9138×10^3	13.9763	0.97	1.2083×10^3	14.1796
1000	1.6030×10^4	14.0148	0.98	1.7527×10^3	14.1839
Tradition designs					
$\phi_{best}(\boldsymbol{d}_{block})$		0			
$\phi_{best}(\boldsymbol{d}_{mseq})$		13.9031			

Table 4.8: Robustness of the designs obtained by the proposed algorithm with misspecified Hurst parameter for estimation When $\tau_{ISI} \neq \tau_{TR}$.

Q	1			2			3		
$\boldsymbol{d}^*(H = \cdot)$	0.9	0.75	0.6	0.9	0.75	0.6	0.9	0.75	0.6
$\phi(0.9)$	1	0.999	0.987	1	0.998	0.991	1	0.998	0.993
$\phi(0.75)$	0.998	1	0.999	0.995	1	0.999	0.996	1	1
$\phi(0.6)$	0.993	0.999	1	0.989	0.998	1	0.989	0.995	1

in the next subsection. The best ϕ-values achieved over the iterations of our algorithm is presented in plot (A). Plot (B) gives the search path, as excepted, at the beginning of the search, the ϕ-value goes up and down and there is a high probability to accept a worsen design; see also plot (C) of Figure 4.10. Same as estimation problem, the algorithm has a significant improvement of ϕ-value at the early stage of the search, it then slowly approach the optimal solution. Plot (D) presents the D-optimal design we obtained. The pattern of the design is very close to a block design.

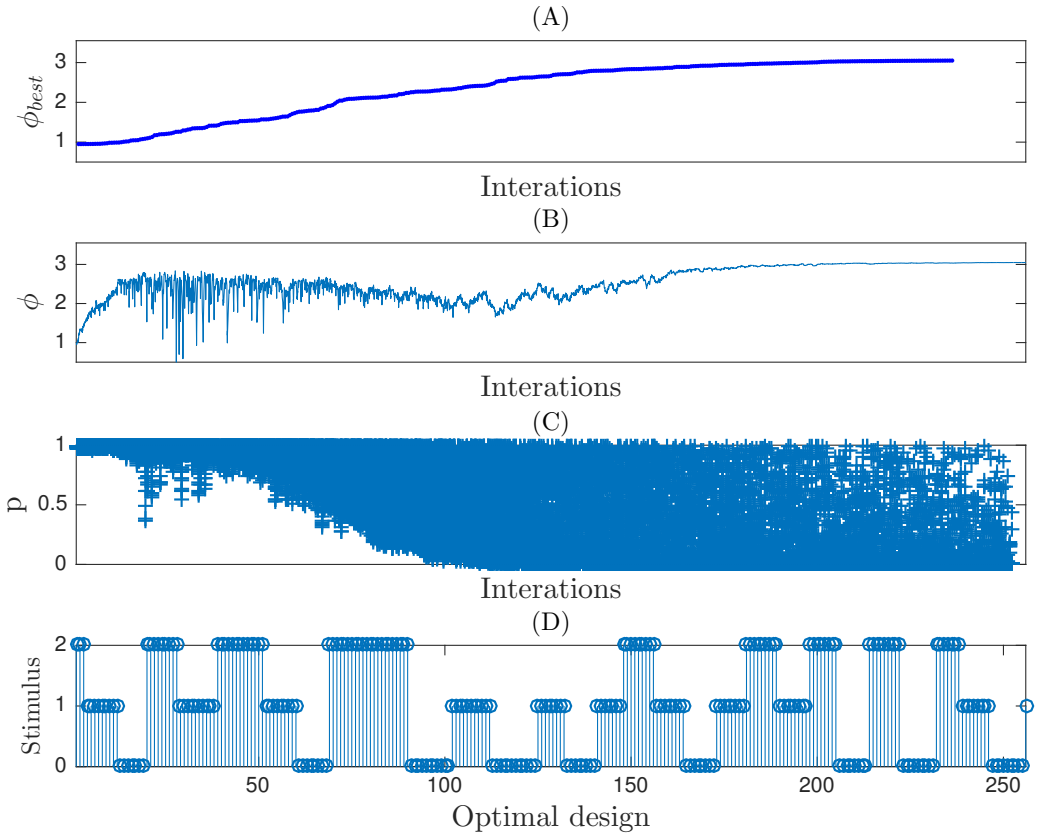

Figure 4.10: Simulation process of the newly proposed algorithm for detection problem when $\tau_{ISI} = \tau_{TR}$ with $Q = 2$, the reduction factor $R = 0.96$. Plots from top to bottom: (A) The record of best ϕ-values during the entire search for the optimal design; (B) ϕ-values for each iteration during the searching process; (C) Probability for accepting a worsen design; (D) The optimal design obtained.

Tables 4.9, 4.10 and 4.11 show the best ϕ-values of D-optimality designs and the computing times of Saleh *et al.* (2017)'s algorithm and our proposed algorithm with $Q = 1, 2$ and 3, for $\tau_{ISI} = \tau_{TR}$. From these results, we observe that our new algorithm can obtain optimal designs that perform similarly to the designs obtained by Saleh *et al.* (2017)'s algorithm with less CPU time. However, we do not have a significantly improved result by increasing the number of initial designs for the algorithm of Saleh *et al.* (2017). Similar observation is made for our algorithm when we elevate the value of the reduction factor.

Table 4.9: The performance of the optimal designs for detection obtained from Saleh *et al.* (2017)'s algorithm (old algorithm) with I initial designs, our proposed algorithm (new algorithm) with R reduction factor, block design and m-sequence based design, and the their computing time (seconds): $Q = 1$, $\tau_{ISI} = \tau_{TR}$.

Old algorithm			New algorithm		
Initials	Time (s)	ϕ_{best}	R	Time (s)	ϕ_{best}
10	3.1893	4.0436	0.70	1.7270	4.0201
25	9.3451	4.0452	0.75	2.5745	4.0353
50	18.4501	4.0533	0.80	2.8857	4.0324
75	27.7966	4.0529	0.85	4.0925	4.0503
100	36.2295	4.0526	0.90	6.1673	4.0457
250	87.8387	4.0539	0.96	14.4039	4.0372
500	176.3849	4.0559	0.97	18.7579	4.0462
1000	363.1153	4.0543	0.98	28.8654	4.0425
Tradition designs					
$\phi_{best}(\boldsymbol{d}_{block})$		3.9884			
$\phi_{best}(\boldsymbol{d}_{mseq})$		1.3678			

Table 4.10: The performance of the optimal designs for detection obtained from Saleh *et al.* (2017)'s algorithm (old algorithm) with I initial designs, our proposed algorithm (new algorithm) with R reduction factor, block design and m-sequence based design, and the their computing time (seconds): $Q = 2$, $\tau_{ISI} = \tau_{TR}$.

Old algorithm			New algorithm		
Initials	Time (s)	ϕ_{best}	R	Time (s)	ϕ_{best}
10	9.8523	3.0507	0.70	5.4796	3.0440
25	22.5540	3.0597	0.75	6.7517	3.0500
50	45.5222	3.0571	0.80	8.4988	3.0484
75	67.7046	3.0596	0.85	11.4983	3.0517
100	91.4980	3.0592	0.90	17.8382	3.0590
250	221.8722	3.0594	0.96	38.5067	3.0500
500	455.0383	3.0637	0.97	51.7084	3.0599
1000	884.0667	3.0622	0.98	77.7042	3.0554
Tradition designs					
$\phi_{best}(\boldsymbol{d}_{block})$		3.0460			
$\phi_{best}(\boldsymbol{d}_{mseq})$		0.9914			

Table 4.12 presents the results for the robustness of the obtained optimal deigns from our proposed algorithm with respect to misspecified Hurst parameter where H is set to $0.4, 0.75, 0.9$ of a long memory process, for $Q = 1, 2, 3$. We conclude that the optimal designs obtained from our proposed algorithm are robust with misspecified Hurst parameter. With an inaccurate Hurst parameter in model assumption, we still be able to get a relatively good design evaluated by a given optimal criterion with the 'true' Hurst parameter.

Table 4.11: The performance of the optimal designs for detection obtained from Saleh *et al.* (2017)'s algorithm (old algorithm) with I initial designs, our proposed algorithm (new algorithm) with R reduction factor, block design and m-sequence based design, and the their computing time (seconds): $Q = 3$, $\tau_{ISI} = \tau_{TR}$.

Old algorithm			New algorithm		
Initials	Time (s)	ϕ_{best}	R	Time (s)	ϕ_{best}
10	12.4685	2.4842	0.70	10.7526	2.4747
25	29.9098	2.4834	0.75	14.2351	2.4773
50	60.1788	2.4871	0.80	16.8970	2.4774
75	94.6489	2.4849	0.85	22.5703	2.4845
100	119.6936	2.4855	0.90	34.5422	2.4840
250	296.1110	2.4873	0.96	80.8378	2.4847
500	591.9253	2.4872	0.97	108.1506	2.4830
1000	1.1861 $\times 10^3$	2.4881	0.98	162.0152	2.4846
Tradition designs					
$\phi_{best}(\boldsymbol{d}_{block})$		2.4763			
$\phi_{best}(\boldsymbol{d}_{mseq})$		0.8297			

Table 4.12: Robustness of the designs obtained by the proposed algorithm with misspecified Hurst parameter for detection When $\tau_{ISI} = \tau_{TR}$.

Q	1			2			3		
$\boldsymbol{d}^*(H = \cdot)$	0.9	0.75	0.6	0.9	0.75	0.6	0.9	0.75	0.6
$\phi(0.9)$	1	0.999	0.987	1	0.999	0.991	1	0.996	0.985
$\phi(0.75)$	0.998	1	0.996	0.997	1	0.997	0.999	1	0.997
$\phi(0.6)$	1	1.003	1	0.997	1.002	1	0.996	1.001	1

4.5.1.4 Detection Problem with $\tau_{ISI} \neq \tau_{TR}$

We now present selected the results for detection problem where $\tau_{ISI} \neq \tau_{TR}$. Tables 4.13, 4.14 and 4.15 convey very similar information as their counterparts for the case where $\tau_{ISI} = \tau_{TR}$. The ϕ-values of the optimal designs obtained from both algorithms are very close, however, our proposed algorithm requires much less computing time. We can thus conclude that our proposed algorithm give good quality designs without much computational effort.

Table 4.13: The performance of the optimal designs for detection obtained from Saleh *et al.* (2017)'s algorithm (old algorithm) with I initial designs, our proposed algorithm (new algorithm) with R reduction factor, block design and m-sequence based design, and the their computing time (seconds): $Q = 1$, $\tau_{ISI} \neq \tau_{TR}$.

Old algorithm			New algorithm		
Initials	Time (s)	ϕ_{best}	R	Time (s)	ϕ_{best}
10	4.7702	3.6318	0.70	4.4399	3.6123
25	12.1525	3.6358	0.75	5.2267	3.6184
50	22.6666	3.6449	0.80	12.9373	3.6443
75	34.8532	3.6408	0.85	8.5356	3.6438
100	46.5148	3.6473	0.90	12.9373	3.6443
250	110.8309	3.6441	0.96	20.6292	3.6458
500	222.2530	3.6441	0.97	28.3089	3.6500
1000	446.6612	3.6481	0.98	41.8434	3.6480
Tradition designs					
$\phi_{best}(\boldsymbol{d}_{block})$		3.6130			
$\phi_{best}(\boldsymbol{d}_{mseq})$		1.6652			

Table 4.14: The performance of the optimal designs for detection obtained from Saleh *et al.* (2017)'s algorithm (old algorithm) with I initial designs, our proposed algorithm (new algorithm) with R reduction factor, block design and m-sequence based design, and the their computing time (seconds): $Q = 2$, $\tau_{ISI} \neq \tau_{TR}$.

Old algorithm			New algorithm		
Initials	Time (s)	ϕ_{best}	R	Time (s)	ϕ_{best}
10	12.0889	2.7295	0.70	16.5506	2.7345
25	27.8303	2.7420	0.75	20.3560	2.7406
50	53.4159	2.7376	0.80	29.3155	2.7459
75	78.5595	2.7385	0.85	35.0732	2.7480
100	109.1063	2.7429	0.90	52.5037	2.7457
250	264.6267	2.7418	0.96	79.4459	2.7484
500	537.9187	2.7474	0.97	105.5870	2.7419
1000	1.0637×10^3	2.7454	0.98	158.9135	2.7417
Tradition designs					
$\phi_{best}(\boldsymbol{d}_{block})$		2.7596			
$\phi_{best}(\boldsymbol{d}_{mseq})$		1.2628			

4.6 Conclusion and Discussion

In this chapter, we develop an efficient computer algorithm to obtain optimal fMRI designs for estimating the HRFs and detecting brain activation using the wavelet-based models. The proposed algorithm was inspired by Saleh *et al.* (2017), to avoid being trapped into local optimum, we incorporate a simulated annealing algorithm and the widely used coordinate-exchange algorithm. We then demonstrate the usefulness of the new algorithm through several case studies. The performance of the new algorithm is compared with Saleh *et al.* (2017)'s algorithm along with some tradition designs under the D-optimality

Table 4.15: The performance of the optimal designs for detection obtained from Saleh *et al.* (2017)'s algorithm (old algorithm) with I initial designs, our proposed algorithm (new algorithm) with R reduction factor, block design and m-sequence based design, and the their computing time (seconds): $Q = 3$, $\tau_{ISI} \neq \tau_{TR}$.

Old algorithm			New algorithm		
Initials	Time (s)	ϕ_{best}	R	Time (s)	ϕ_{best}
10	16.8382	2.2145	0.70	38.9274	2.2202
25	35.0378	2.2128	0.75	45.4260	2.2269
50	70.8276	2.2165	0.80	66.0379	2.2213
75	118.5528	2.2129	0.85	82.8047	2.2225
100	148.3543	2.2182	0.90	127.5102	2.2293
250	368.6457	2.2249	0.96	191.0071	2.2259
500	730.7741	2.2233	0.97	262.7654	2.2234
1000	1.4634×10^3	2.2244	0.98	398.1672	2.2263
Tradition designs					
$\phi_{best}(\boldsymbol{d}_{block})$		2.2360			
$\phi_{best}(\boldsymbol{d}_{mseq})$		1.1248			

Table 4.16: Robustness of the designs obtained by the proposed algorithm with misspecified Hurst parameter for detection When $\tau_{ISI} \neq \tau_{TR}$.

Q	1			2			3		
$\boldsymbol{d}^*(H = \cdot)$	0.9	0.75	0.6	0.9	0.75	0.6	0.9	0.75	0.6
$\phi(0.9)$	1	0.996	0.989	1	1	0.973	1	0.997	0.981
$\phi(0.75)$	1.002	1	0.999	0.999	1	0.992	0.998	1	0.992
$\phi(0.6)$	1	1	1	0.998	1.008	1	1.002	1.004	1

criterion. The results show that our algorithm requires significantly less time to obtain optimal designs with similar design efficiencies, compared with Saleh *et al.* (2017)'s algorithm. The optimal designs obtained by the new algorithm always perform better than tradition designs. Furthermore, our algorithm is superior to Saleh *et al.* (2017)'s algorithm as the design matrix becomes larger in terms of computing time. For comparison purposes, we choose the D-optimality criterion as in Saleh *et al.* (2017), however, our algorithm can be further extended to other optimality criteria, such as A-optimality criterion. In addition, to check the robustness to the Hurst parameter of our algorithm, we run the simulations with different values of the Hurst parameter, with the obtained optimal designs, we then evaluate them under 'true' Hurst parameters. We observe that our algorithm is robust to the Hurst parameter. That is, if the experimenter use a wrong Hurst parameter for error term in the model to obtain an optimal design, it will still perform well with a 'true' Hurst parameter.

REFERENCES

Atkinson, A. C., A. N. Donev and R. D. Tobias, *Optimum experimental designs, with SAS*, vol. 34 (Oxford University Press Oxford, 2007).

Baillie, R. T., "Long memory processes and fractional integration in econometrics", Journal of Econometrics **73**, 1, 5 – 59 (1996).

Barbee, K., W. Van Moer and G. Nagels, "Fractional-order time series models for extracting the haemodynamic response from functional magnetic resonance imaging data", IEEE Trans Biomed Eng **59**, 8, 2264–72 (2012).

Beran, J., *Statistics for long-memory processes*, vol. 61 (CRC Press, 1994).

Brammer, M. J., "Multidimensional wavelet analysis of functional magnetic resonance images", Hum Brain Mapp **6**, 5-6, 378–82 (1998).

Büchel, C. and K. Friston, "Modulation of connectivity in visual pathways by attention: cortical interactions evaluated with structural equation modelling and fmri.", Cerebral cortex **7**, 8, 768–778 (1997).

Bullmore, E., M. Brammer, S. C. Williams, S. Rabe-Hesketh, N. Janot, A. David, J. Mellers, R. Howard and P. Sham, "Statistical methods of estimation and inference for functional MR image analysis", Magnetic Resonance in Medicine **35**, 2, 261–277 (1996).

Bullmore, E., J. Fadili, V. Maxim, L. Şendur, B. Whitcher, J. Suckling, M. Brammer and M. Breakspear, "Wavelets and functional magnetic resonance imaging of the human brain", Neuroimage **23**, S234–S249 (2004).

Bullmore, E., C. Long, J. Suckling, J. Fadili, G. Calvert, F. Zelaya, T. A. Carpenter and M. Brammer, "Colored noise and computational inference in neurophysiological (fMRI) time series analysis: resampling methods in time and wavelet domains", Hum Brain Mapp **12**, 2, 61–78 (2001).

Buracas, G. T. and G. M. Boynton, "Efficient design of event-related fMRI experiments using M-sequences", Neuroimage **16**, 3 Pt 1, 801–13 (2002).

Caporin, M., J. Pres *et al.*, "Forecasting temperature indices with time-varying long-memory models", Available in SSRN (2009).

Černỳ, V., "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm", Journal of optimization theory and applications **45**, 1, 41–51 (1985).

Clegg, R. G., "A practical guide to measuring the hurst parameter", arXiv preprint math/0610756 (2006).

Cordes, D., G. Herzmann, R. Nandy and T. Curran, "Optimization of contrast detection power with probabilistic behavioral information", Neuroimage **60**, 3, 1788–99 (2012).

Costafreda, S. G., G. J. Barker and M. J. Brammer, "Bayesian wavelet-based analysis of functional magnetic resonance time series", Magn Reson Imaging **27**, 4, 460–9 (2009).

Dale, A. M., "Optimal experimental design for event-related fMRI", Hum Brain Mapp **8**, 2-3, 109–14 (1999).

Daubechies, I., *Ten lectures on wavelets* (SIAM, 1992).

Delzell, D. A., R. F. Gunst, W. R. Schucany, P. S. Carmack, Q. Lin, J. S. Spence and R. W. Haley, "Key properties of d-optimal designs for event-related functional mri experiments with application to nonlinear models", Statistics in medicine **31**, 29, 3907–3920 (2012).

Dietrich, C. R. and G. N. Newsam, "Fast and exact simulation of stationary gaussian processes through circulant embedding of the covariance matrix", SIAM Journal on Scientific Computing **18**, 4, 1088–1107 (1997).

et al., J. J. M., "Functional magnetic resonance imaging: The current science of clinical application", Neuropsychology Review **17** (2007).

Fadili, M. J. and E. T. Bullmore, "Wavelet-generalized least squares: a new BLU estimator of linear regression models with 1/f errors", Neuroimage **15**, 1, 217–32 (2002).

Friston, K. J., A. P. Holmes, J. B. Poline, P. J. Grasby, S. C. Williams, R. S. Frackowiak and R. Turner, "Analysis of fMRI time-series revisited", Neuroimage **2**, 1, 45–53 (1995).

Friston, K. J., P. Jezzard and R. Turner, "Analysis of functional MRI time-series", Human brain mapping **1**, 2, 153–171 (1994).

Glover, G. H., "Deconvolution of impulse response in event-related BOLD fMRI", Neuroimage **9**, 4, 416–29 (1999).

Gotwalt, C. M., B. A. Jones and D. M. Steinberg, "Fast computation of designs robust to parameter uncertainty for nonlinear settings", Technometrics **51**, 1, 88–95 (2009).

Harville, D. A., *Matrix algebra from a statistician's perspective*, vol. 1 (Springer, 1997).

Hosking, J. R., "Fractional differencing", Biometrika **68**, 1, 165–176 (1981).

Hurst, H. E., "Long-term storage capacity of reservoirs", Trans. Amer. Soc. Civil Eng. **116**, 770–808 (1951).

Jean-Francois, C., "Simulation and identification of the franctional brownian motion: A bibliographical and comparative study", (2007).

Jeong, J., M. Vannucci and K. Ko, "A wavelet-based Bayesian approach to regression models with long memory errors and its application to fMRI data", Biometrics **69**, 1, 184–196 (2013).

Jones, B. and P. Goos, "A candidate-set-free algorithm for generating d-optimal split-plot designs", Journal of the Royal Statistical Society: Series C (Applied Statistics) **56**, 3, 347–364 (2007).

Kao, M.-H., "Multi-objective optimal experimental designs for ER-fMRI using MATLAB", Journal of Statistical Software **30**, 11, 1–13 (2009).

Kao, M.-H., A. Mandal, N. Lazar and J. Stufken, "Multi-objective optimal experimental designs for event-related fMRI studies", Neuroimage **44**, 3, 849–56 (2009).

Kao, M.-H. and H. D. Mittelmann, "A fast algorithm for constructing efficient event-related fMRI designs", Computational Statistics and Data Analysis (2011).

Kirkpatrick, S., C. D. Gelatt, M. P. Vecchi *et al.*, "Optimization by simulated annealing", science **220**, 4598, 671–680 (1983).

Kwong, K. and D. Chesler, "Functional MRI", Medical Devices and Systems pp. 22–30 (1995).

Lazar, N., *The statistical analysis of functional MRI data* (Springer Science & Business Media, 2008).

Lindquist, M. A. *et al.*, "The statistical analysis of fMRI data", Statistical Science **23**, 4, 439–464 (2008).

Liu, T. T. and L. R. Frank, "Efficiency, power, and entropy in event-related fMRI with multiple trial types. part i: theory", Neuroimage **21**, 1, 387–400 (2004).

Mandelbrot, B. B. and J. W. V. Ness, "Fractional brownian motions, fractional noises and applications", SIAM Review **10**, 4, 422–437 (1968).

Mandelbrot, B. B. and M. S. Taqqu, *Robust R/S analysis of long run serial correlation* (IBM Thomas J. Watson Research Division, 1979).

Mandelbrot, B. B. and J. R. Wallis, "Computer experiments with fractional gaussian noises: Part 2, rescaled ranges and spectra", Water resources research **5**, 1, 242–259 (1969).

Maus, B., G. J. P. van Breukelen, R. Goebel and M. P. F. Berger, "Robustness of optimal design of fMRI experiments with application of a genetic algorithm", Neuroimage **49**, 3, 2433–43 (2010).

McCoy, E. and A. Walden, "Wavelet analysis and synthesis of stationary long-memory processes", Journal of Computational and Graphical Statistics **5**, 1, 26–56 (1996).

Meyer, R. K. and C. J. Nachtsheim, "The coordinate-exchange algorithm for constructing exact optimal experimental designs", Technometrics **37**, 1, 60–69 (1995).

Pritchard, C., A. Mayers and D. Baldwin, "Changing patterns of neurological mortality in the 10 major developed countries–1979-2010", Public Health **127**, 4, 357–68 (2013).

Purdon, P. L., V. Solo, R. M. Weisskoff and E. N. Brown, "Locally regularized spatiotemporal modeling and model comparison for functional {MRI}", NeuroImage **14**, 4, 912 – 923 (2001).

Purdon, P. L. and R. M. Weisskoff, "Effect of temporal autocorrelation due to physiological noise and stimulus paradigm on voxel-level false-positive rates in fmri", Human brain mapping **6**, 4, 239–249 (1998).

Rencher, A. C. and G. B. Schaalje, *Linear models in statistics* (John Wiley & Sons, 2008).

Ruttimann, U. E., M. Unser, R. Rawlings, D. Rio, N. Ramsey, V. Mattay, D. Hommer, J. Frank and D. Weinberger, "Statistical analysis of functional MRI data in the wavelet domain", Medical Imaging, IEEE Transactions on **17**, 2, 142–154 (1998).

Saleh, M., M.-H. Kao and R. Pan, "Design D-optimal event-related functional magnetic resonance imaging experiments", Journal of the Royal Statistical Society: Series C (Applied Statistics) **66**, 1, 73–91 (2017).

Tan, C. M. and N. Raghavan, *Simulated Annealing for mixture distribution analysis and its applications to reliability testing* (INTECH Open Access Publisher, 2008).

Taqqu, M. S., V. Teverovsky and W. Willinger, "Estimators for long-range dependence: an empirical study", Fractals **3**, 04, 785–798 (1995).

Wager, T. D. and T. E. Nichols, "Optimization of experimental design in fMRI: a general framework using a genetic algorithm", Neuroimage **18**, 2, 293–309 (2003).

Weisskoff, R., J. Baker, J. Belliveau, T. Davis, K. Kwong, M. Cohen and B. Rosen, "Power spectrum analysis of functionally-weighted mr data: what?s in the noise", in "Proc. Soc. Magn. Reson. Med", vol. 1 (1993).

Weyl, H., "Bemerkungen zum begriff des differentialquotienten gebrochener ordnung", Zürich. Naturf. Ges **62**, 296–302 (1917).

Wood, A. T. and G. Chan, "Simulation of stationary gaussian processes in [0, 1] d", Journal of computational and graphical statistics **3**, 4, 409–432 (1994).

Wornell, G. W. and A. V. Oppenheim, "Estimation of fractal signals from noisy measurements using wavelets", Signal Processing, IEEE Transactions on **40**, 3, 611–623 (1992).

Worsley, K. J. and K. J. Friston, "Analysis of fMRI time-series revisited–again", Neuroimage **2**, 3, 173–81 (1995).

Worsley, K. J., C. Liao, J. Aston, V. Petre, G. Duncan, F. Morales and A. Evans, "A general statistical analysis for fmri data", Neuroimage **15**, 1, 1–15 (2002).

Zarahn, E., G. K. Aguirre and M. D'Esposito, "Empirical analyses of BOLD fMRI statistics. i. spatially unsmoothed data collected under null-hypothesis conditions", Neuroimage **5**, 3, 179–97 (1997).

Zhou, L., *Robust experimental designs for fMRI with an uncertain design matrix*, Master's thesis, Arizona State University (2014).