UNIVERSITY OF SOUTHAMPTON

Faculty of Social Sciences
School of Mathematical Sciences

Design of experiments for models involving profile factors

by

Damianos Michaelides

A thesis for the degree of
Doctor of Philosophy

March 2023

University of Southampton

Abstract

Faculty of Social Sciences
School of Mathematical Sciences

Doctor of Philosophy

Design of experiments
for models involving profile factors

by Damianos Michaelides

In the traditional design of experiments, it is assumed that each run of the experiment involves the application of a treatment, consisting of static settings of the controllable factors. The objective of this work is to extend the usual optimal experimental design paradigm to modern experiments where the settings of factors are functions. Such factors are known as profile factors, or as dynamic factors. For these new experiments, the design problem is to identify optimal experimental conditions to vary the profile factors in each run of the experiment.

In general, functions are infinite dimensional objects. The latter produces challenges in estimation and design. To face the challenges, a new methodology using basis functions is developed. Primary focus is given on the B-spline basis system, due to its computational efficiency and useful properties. The methodology is applied to a functional linear model, and expanded to a functional generalised linear model, reducing the problem to an optimisation of basis coefficients. Special cases, including combinations of profile and scalar factors, interactions, and polynomial effects, are taken into consideration.

The methodology is demonstrated through multiple examples, aiming to find A- and D- optimal experimental designs. The sensitivity of optimal experimental conditions to changes in the settings of the profile factors and the functional parameters is extensively investigated. Bayesian optimal designs are identified through the addition of roughness penalties to penalise the complicated functions. The latter contributes in identifying the connection between the frequentist and the Bayesian approaches.

Contents

List of Figures

List of Tables

Declaration of Authorship

I, Damianos Michaelides, declare that the thesis entitled

"Design of Experiments for models involving profile factors",

and the work presented in the thesis are both my own, and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Part of this work has been added on arXiv:2110.09115 as a working paper.

Signed: Date: 05 November 2022

Acknowledgements

Along these lines, I would like to express my gratitude to the people who have been by my side on this journey. Each of them inspired and motivated me in their own way to the completion of this PhD work. If you are mentioned in the text below, I want you to know that without you, nothing would be the same.

First and foremost, I would like to thank my supervisors, Dr. Antony Overstall and Professor Dave Woods, for their ideas and advice throughout my PhD studies. A large part of my doctorate happened during the Covid-19 pandemic. Nevertheless, they both quickly adapted to the new circumstances and continued to support me even then. For this, I am grateful.

I am thankful to the Design Study Group members and especially to the people at the office, Theodora, Dasha, and Olga, for making the days at the office more enjoyable.

A big thanks goes to my Southampton flatmates. Simos, thank you for sharing two years of our PhD time, for the interesting discussions, and for the company during the lockdown. Andreas, thank you for the memories, for being a good influence, and for being a great flatmate.

I also thank my friends in Cyprus - you know who you are - for their friendship and for making my days at home worthwhile. A specific person I would like to thank is my friend Loizos, who not only during his time in Southampton, but also after his return to Cyprus, supported me and was in touch almost every day.

My greatest acknowledgement goes to my family. This doctorate and my education in general, would not have been possible without the efforts of my parents, Andreas and Ioanna. For this, and for countless other reasons, I will be forever grateful to them. I owe a huge thank you to my brother Michalis, for his continuous support and advice, but also for sharing his knowledge and experiences all these years. I could not but express my gratitude to my grandparents, Damianos and Despina, for everything they have given me in my life.

Last, but certainly not least, I would like to thank Ino, a very special person for many special reasons, who has been next to me and encouraged me with her positive energy, even at the most difficult of times.

Dedication

This thesis is dedicated to my beloved grandmother Despina, who passed away while I was completing the last few months of my doctoral studies. My grandmother gave me immense love and support throughout my life. I will never forget her, and she will always have a special place in my heart. Even though she did not get to see the completion of my PhD, I am sure she is very proud - wherever she is.

Her memory eternal.

List of Symbols

n	Number of runs
$\Psi(\cdot)$	Objective function
$\Psi_A(\cdot)$	A-optimality objective function
$\Psi_D(\cdot)$	D-optimality objective function
$\boldsymbol{y}$	Vector of responses
$\boldsymbol{X}$	Design matrix in the basic linear model
$\boldsymbol{X}^*$	An optimal design in the basic linear model
$\boldsymbol{F}$	Model matrix of the basic linear model
β	Vector of unknown scalar parameters
ϵ	Vector of error terms
$u(\cdot)$	Utility function under the Bayesian framework
$\Psi_{nsel}(\cdot)$	Expected utility of the NSEL utility function
$\Psi_{sig}(\cdot)$	Expected utility of the SIG utility function
$\pi(\cdot)$	Probability density
$\boldsymbol{V}$	Symmetric matrix of known hyperparameters
$\pi(\cdot\|\cdot)$	Conditional probability density
δ	Tolerance value
t	Time
$\mathcal{T}$	Upper bound of time
K	Total number of interior knots
$\boldsymbol{\lambda}$	Vector of interior knots
$\boldsymbol{\lambda}^*$	The extended knot vector
s_h	The degree of continuity of the h^{th} knot
d	The degree of smoothness of a polynomial
N	Total number of basis functions
Υ	Set of truncated power series basis functions
$B_{\kappa,d}(t)$	B-spline of degree d, at the κ^{th} knot span at time t
$D^{(m)}(\cdot)$	m^{th} derivative
$\boldsymbol{x}(t)$	Function of profile factors
$\boldsymbol{x}_{\cdot j}(t)$	Column vector of all the runs of the function of the j^{th} profile factor
$\boldsymbol{\beta}(t)$	Vector of unknown functional parameters
J	Number of profile factors

Q	Number of terms in a model
n_x	Number of basis functions for the profile factors
n_β	Number of basis functions for the functional parameters
$\boldsymbol{X}(t)$	Matrix of all profile factors in every run of the experiment
$\boldsymbol{c}(t)$	Known basis functions for the profile factor
$\boldsymbol{b}(t)$	Known basis functions for the functional parameters
$\boldsymbol{\Gamma}$	Coefficient matrix from the basis expansion for the profile factor / design matrix in a functional m
$\boldsymbol{\Gamma}^*$	An optimal design in a functional model
$\boldsymbol{\theta}$	Coefficient matrix from the basis expansion for the functional parameters / vector of functional m
$\boldsymbol{Z}$	Model matrix of the extended functional model
$\boldsymbol{Z}_{\bullet q}$	The q^{th} partition column block of $\boldsymbol{Z}$
$\mathcal{X}$	Design space
$\boldsymbol{\Theta}$	Parameter space
$\boldsymbol{R}_p$	Block diagonal matrix of the roughness penalties
Λ	Smoothing parameter
$g(\cdot)$	Link function of a GLM
η	Linear predictor of a GLM
$U(\cdot)$	Score functions
$\mathcal{I}(\cdot,\cdot)$	Fisher information matrix
$\psi(\boldsymbol{\theta},\boldsymbol{\Gamma})$	General objective function depending on the model parameters
$\boldsymbol{y}(t)$	Vector of responses/outputs of time-varying models
$\dot{\boldsymbol{y}}(t)$	First derivative of the response/output of time-varying models
$\boldsymbol{y}_0$	Initial conditions in time-varying models
$\boldsymbol{\Psi}_R$	An $(n \times 1) \times (n \times 1)$ correlation matrix
$\boldsymbol{R}$	An $n \times n$ correlation matrix
$\boldsymbol{r}$	Vector of correlations
$\zeta(\cdot,\cdot)$	Correlation function used in the Gaussian Process
$\boldsymbol{\theta}_r$	Vector of unknown parameters in the correlation function
$\boldsymbol{e}$	Function for the Euclidean distance
ν	Smoothness parameter in the Matérn correlation function
K_ν	Modified Bessel function
$\boldsymbol{\Sigma}$	Positive definite covariance matrix
$\boldsymbol{B}$	Vector of unknown model parameters in the Multivariate Gaussian Process

List of Abbreviations

OLS	**O**rdinary **L**east **S**quares
RSS	**R**esidual **S**um of **S**quares
IG	**I**nverse **G**amma
NIG	**N**ormal **I**nverse **G**amma
TPS	**T**runcated **P**ower **S**eries
TPF	**T**runcated **P**ower **F**unction
BS	**B**-**S**pline
RSM	**R**esponse **S**urface **M**ethodology
FLM	**F**unctional **L**inear **M**odel
GLM	**G**eneralised **L**inear **M**odel
FGLM	**F**unctional **G**eneralised **L**inear **M**odel
NSEL	**N**egative **S**quared **E**rror **L**oss
SIG	**S**hannon **I**nformation **G**ain
MC	**M**onte **C**arlo
ODE	**O**rdinary **D**ifferential **E**quations
GP	**G**aussian **P**rocess
MLE	**M**aximum **L**ikelihood **E**stimator
PCA	**P**rincipal **C**omponent **A**nalysis
PLS	**P**artial **L**easts **S**quares

Chapter 1

Introduction

The aim of this thesis is to discuss, review, and enrich the Statistics literature on design of experiments, for models that involve factors whose settings can vary as a function of an indexing variable. Even though statistical modelling with this type of factors is well established, the design of experiments for such models has received much less attention. The motivation is to investigate such models, and develop a new approach for identifying optimal designs.

1.1 Design of Experiments

Experimentation is the process of gathering data under controlled conditions to answer scientific questions of interest. Different experiments are used in many areas of applied science, including medicine, biology, chemistry, and engineering. Common examples include: experiments by pharmaceutical companies to develop new drugs; experiments by agricultural companies to optimise yield from their production processes; and clinical trials to compare drug treatments with current standard treatments. For all applications of experiments, it is essential to collect accurate and relevant data so as to generate valid, objective, and defensible statistical and non-statistical conclusions.

The branch in Statistics that is concerned with the choice of the experimental conditions to maximise the effectiveness in a given experiment, is called design of experiments; see Atkinson et al. (2007), Antony (2014), Dean et al. (2017), and Montgomery (2017). Design of experiments is a well-established research topic in the Statistics literature for more than 100 years. Its purpose is to use all the known features of a particular application to design an experiment, and answer questions of interest. Aims of experiments, for which design of experiments is a useful tool, include:

- Treatment comparison: A treatment is defined as a combination of particular values of the controllable factors. An experimental aim is to compare treatments and choose the most effective.
- Factor screening: A model may include a large number of controllable factors. An experimental aim is to compare the factors and decide which factors have a substantive impact.
- Response surface exploration: Description of the relationship between the response and the important factors.
- Optimisation: The experimental aim, can be represented by a function, called objective function. For example, the objective function may represent cost or gain. An experimental aim is to identify the settings of the controllable factors that minimise or maximise an objective function.

The focus in the thesis is to consider design of experiments to estimate the parameters in a statistical model, and identify efficient designs by optimising the combinations of the settings of factors to be run in the experiment. The settings of factors are represented in a design matrix that belongs to the design space $\mathcal{X}$. Specifically, the objective is to estimate the parameters with respect to the choice of objective criteria, for instance, minimise the average variance of the parameter estimator. A detailed description on optimal experimental designs, objective functions, and optimality criteria is given in Chapter 2.

Prior to the design of an experiment it is vital to know the aim of the experiment, as well as the controllable factors and their capabilities of variation. In the traditional design of experiments, for a single run of the experiment the set up involves the application of a treatment consisting of fixed or static settings of the controllable factors. In every run a simultaneous variation of the different factors takes place with the aim of maximising the effectiveness of the given experiment.

Suppose an experiment of n runs, J controllable factors, and responses y_i are collected, for $i = 1, 2, \dots, n$ with i representing the i^{th} run of the experiment. One way of modelling the relationship between the scalar responses and the J controllable factors is a linear model of the form:

$$y_i = \boldsymbol{f}^T(\boldsymbol{x}_i)\boldsymbol{\beta} + \epsilon_i, \quad i = 1, 2, \dots, n. \tag{1.1}$$

The vector of $\boldsymbol{x}_i$ represents the values of the controllable factors for the i^{th} run of the experiment, and it is defined as:

$$\boldsymbol{x}_i^T = \begin{pmatrix} x_{i1} & x_{i2} & \cdots & x_{iJ} \end{pmatrix}, \quad i = 1, 2, \dots, n, \tag{1.2}$$

with each $x_{ij}, i = 1, 2, \ldots, n, j = 1, 2, \ldots, J$, the value of the j^{th} controllable factor at the i^{th} run of the experiment. The values of the controllable factors are usually in an interval $[u, v]$, i.e., $x_{ij} \in [u, v]$, with u and v scalar values. The function $\boldsymbol{f}(\boldsymbol{x}_i)$ is a $Q \times 1$ vector, that contains the effects of the explanatory variables at the i^{th} run of the experiment such that,

$$\boldsymbol{f}^T(\boldsymbol{x}_i) = \begin{pmatrix} f_1(\boldsymbol{x}_i) & f_2(\boldsymbol{x}_i) & \cdots & f_Q(\boldsymbol{x}_i) \end{pmatrix}, \quad i = 1, 2, \ldots, n. \tag{1.3}$$

This means that the number of terms in the model depends on the specification of the function $\boldsymbol{f}^T(\boldsymbol{x}_i)$. The intercept is incorporated into the model through the function $f(\boldsymbol{x}_i)$. Meaning that if the intercept is included in the model, then the first entry of $\boldsymbol{f}(\boldsymbol{x}_i)$ is always equal to one, i.e., $f_1(\boldsymbol{x}_i) = 1$. For example, if the model includes the intercept and the main effects of the J controllable factors, then $\boldsymbol{f}^T(\boldsymbol{x}_i) = (1 \; \boldsymbol{x}_i^T)$ and $Q = J + 1$. The vector $\boldsymbol{\beta}$ is the vector of the Q unknown scalar parameters, and finally ϵ_i the independent and identically distributed errors with mean zero and variance σ^2.

In matrix form, the linear model in (1.1) is given by,

$$\boldsymbol{y} = \boldsymbol{F}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{1.4}$$

where $\boldsymbol{y}$ is the $n \times 1$ vector of responses, that is $\boldsymbol{y}^T = (y_1, y_2, \ldots, y_n)$, and $\boldsymbol{\beta}$ is a vector for the unknown parameters as before. The matrix $\boldsymbol{F}$ is the model matrix of dimension $n \times Q$, with the i^{th} row of $\boldsymbol{F}$ being $\boldsymbol{f}^T(\boldsymbol{x}_i)$. Finally, $\boldsymbol{\epsilon}$ is a vector of length n of the error terms. It is assumed that the responses are observed independently of each other. Thus, the errors are independent with,

$$\text{Var}(\boldsymbol{\epsilon}) = \boldsymbol{I}_n\sigma^2,$$

with $\boldsymbol{I}_n$ the $n \times n$ identity matrix.

The design problem is to find the settings of the factors $x_{i1}, x_{i2}, \ldots, x_{iJ}$ for each of the n runs, that minimise or maximise an objective function. To identify the connection between the design problem and model matrix, the model matrix can be alternatively defined as,

$$\boldsymbol{F} = \boldsymbol{f}^T(\boldsymbol{X}),$$

where $\boldsymbol{X}$ is the design matrix, carrying the settings of the controllable factors such that,

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_n^T \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_{\cdot 1} & \boldsymbol{x}_{\cdot 2} & \cdots & \boldsymbol{x}_{\cdot J} \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1J} \\ x_{21} & x_{22} & \cdots & x_{2J} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nJ} \end{pmatrix},$$

with the vector $\boldsymbol{x}_{\cdot j}, j = 1, 2, \ldots, J$ representing the j^{th} controllable factor in every run of the experiment. Hence, optimisation is required to find optimal conditions for the controllable factors, i.e., an optimal design matrix $\boldsymbol{X}$.

1.2 Motivation of the thesis

The vast majority of experiments conducted by practitioners and researchers, depend on static or fixed factors. In contrast to this traditional design of experiments, an increasing number of modern experiments involve factors whose settings can vary as a function of an indexing variable, such as space or, commonly, time, within a single run of the experiment. This is due to the increased use of dynamic systems in many disciplines including biology, medicine, and engineering, for which time is an inevitable component; see Titterington (1980). In the Statistics literature, such factors are often referred to as profile, and sometimes as functional or dynamic, factors; see Georgakis (2013) and Klebanov and Georgakis (2016).

Ferraty and Vieu (2006) refer to profile factors as factors that appear in the form of curves rather than scalar objects. A common example of a profile factor is temperature, being varied monotonically or as a step function through every run of an experiment. Moreover, examples of experiments that involve profile factors include: varying the temperature and the humidity during a chemical reaction (Putranto et al., 2011), and controlling factors that mimic dynamic wind speed and direction in engineering to develop wind turbines (Boukhezzar and Siguerdidjane, 2005).

For experiments depending on profile factors, the design problem is to choose the right function to vary each of the profile factors in every run of the experiment. A challenge faced when experimentation involves profile factors is that the function space is very general. Thus, it is required to restrict the function space; see Section 3.1. For ease of exposition, throughout this thesis it is assumed that time, $t \in [0, \mathcal{T}]$, is the continuous single input to the profile factors. The methods extend naturally to situations where there are profile factors with multiple inputs, e.g., spatio-temporal studies.

Statistical modelling with functional data is well established in the Statistics literature; see the work by Ramsay and Silverman (2005) for an introduction. Despite the increased use of profile factors in modern experiments, minor consideration was given on experimental design strategies for experiments with profile factors. Two main approaches have been proposed: response surface methods using dimension-reduction techniques, and optimal design for dynamic models, typically derived from differential equations; see Section 3.3. In fact, there is no work addressing functional empirical models depending on multiple profile factors, where the parameters requiring estimation are themselves functions of time.

This thesis extends the optimal design paradigm to experiments with profile factors and time-varying parameters, so that the design consists of combinations of functions for each run of the experiment. As contribution to the Statistics literature, this work involves researching suitable modelling methods, both empirical and mechanistic, and optimal design strategies along with associated computational efficient techniques.

1.3 Examples of experiments with profile factors

1.3.1 Biopharmaceutical experiment to study cell growth

A common experiment in the biopharmaceutical industry, aims to study the cell culture, to monitor and optimise the cell growth. The cell growth is studied using an automated bioreactor system for process development, called Ambr250. As indicated by Tai et al. (2015), the importance of the use of design of experiments with the Ambr250 bioreactor system is to allow a more efficient exploration of the design space. The experiment is performed with 12 or 24 cell lines and each cell line corresponds to a run of the experiment. The response in the i^{th} run of the experiment, denoted as y_i, for $i = 1, 2, \ldots, n$, include the final measurement of cell growth in the i^{th} cell line. The controllable factors are the initial cell concentration x_{i1}, the carbon dioxide level x_{i2}, and the temperature x_{i3}, at the i^{th} run of the experiment. This experiment can be considered as a standard design of experiments problem represented by a linear model such that,

$$y_i = \beta_1 + \beta_2 x_{i1} + \beta_3 x_{i2} + \beta_4 x_{i3} + \epsilon_i, \quad i = 1, 2, \ldots, n,$$

where levels of each factor are chosen.

Several scientists in the literature mentioned that considering factors that vary with respect to time could be beneficial; see Yoon et al. (2003), Trummer et al. (2006), and Rameez et al. (2014). In context, Yoon et al. (2003) indicated the importance of changing temperature, and discussed that lowering the values of temperature dynamically promoted high cell concentration. Trummer et al. (2006) found that a maximisation of cell growth can be obtained by varying temperature using a biphasic process strategy. In a more recent work, Rameez et al. (2014) concluded that a downward shift in temperature caused an increase in the concentration of the cell.

Based on the latter findings, the biopharmaceutical experiment can be seen as a design of experiments problem with two scalar factors which are the initial cell concentration and the CO2 level, and a single profile factor which is the temperature. For example, a single run of the experiment could be the choice of levels for CO2 and initial cell concentration, and a function of time for temperature, meaning changing the values of temperature dynamically. Under this scenario, the experiment can be modelled as a functional linear model and take the form,

$$y_i = \beta_1 + \beta_2 x_{i1} + \beta_3 x_{i2} + \int_0^{\mathcal{T}} \beta_4(t) x_{i3}(t)\, dt + \epsilon_i, \quad i = 1, 2, \ldots, n,\ t \in [0, \mathcal{T}], \tag{1.5}$$

with $\beta_4(t)$ and $x_{i3}(t)$ the functional parameter and profile factor respectively, integrated with respect to time.

A similar experiment in the biopharmaceutical industry, vary the feed volume, which can be considered as a profile factor that is varied dynamically. Models of the nature of the model for the biopharmaceutical experiment in (1.5) are of main interest in this thesis. A detailed methodology is developed in Chapter 5, with multiple examples in Chapter 6.

In chemistry, the actively growing cells in a sample can be identified by viable cell count. Hence, the data from the biopharmaceutical experiment, to measure the final cell growth can be modelled using a Poisson log-linear model, which is a generalised linear model used to model counts. If temperature is assumed static, the experiment can be modelled as a standard log-linear Poisson model such that,

$$\begin{aligned} y_i &\sim \text{Poisson}(\mu_i), \\ \eta_i &= \beta_1 + \beta_2 x_{i1} + \beta_3 x_{i2} + \beta_4 x_{i3} \quad i = 1, 2, \ldots, n, \end{aligned}$$

with a link function $g(\mu_i) = \log(\mu_i)$ and μ_i representing the mean of the i^{th} response. However, if the temperature is allowed to vary over time, then the experiment is modelled as a functional Poisson model and takes the form,

$$\begin{aligned} y_i &\sim \text{Poisson}(\mu_i), \\ \eta_i &= \beta_1 + \beta_2 x_{i1} + \beta_3 x_{i2} + \int_0^{\mathcal{T}} \beta_4(t) x_{i3}(t)\, dt \quad i = 1, 2, \ldots, n,\ t \in [0, \mathcal{T}]. \end{aligned} \tag{1.6}$$

For experiments modelled by generalised linear models depending on profile factors as in (1.6), a design of experiments methodology is developed and discussed in Chapter 7.

1.3.2 Batch reactor experiment to study single compound conversion

A common problem in chemical kinetics focus on the optimisation of a batch reactor. A batch reactor is a process used to study the behaviour and the relationship between a reactant and a product or material. Design of experiments contributes to the latter problem through the provision of a methodology to study the relationship between an output and input factors or processes. However, in batch processes a dynamic movement of factors, such as temperature, have a substantial effect on the output; see Rippin (1983), Georgakis (2013) and Klebanov and Georgakis (2016). For this reason, the use of profile factors could be beneficial.

An example tackled by Georgakis (2013) considers a batch reactor with a reversible reaction of a reactant and a single product. The aim of the study is to study the effect of temperature on the conversion of the compound. For instance, the response is the conversion and the input factor is the temperature x_{i1}. Georgakis (2013) and Klebanov and Georgakis (2016) assumed a fixed batch time at 2 and 1 hours respectively. A possible experiment could assume scalar levels of temperature for every run of the experiment. However, due to the advantage of allowing the temperature to change with respect to time, this problem can be seen as a design of experiments problem depending on a profile factor $x_{i1}(t)$.

Under this scenario, the experiment is modelled as a functional linear model and takes the form,

$$y_i = \beta_1 + \int_0^{\mathcal{T}} \beta_2(t) x_{i1}(t)\, dt + \epsilon_i, \quad i = 1, 2, \ldots, n,\ t \in [0, \mathcal{T}], \tag{1.7}$$

with y_i the conversion response at the end of the batch, β_1 the constant parameter, and $\beta_2(t)$ and $x_{i1}(t)$ the functional parameter and profile factor respectively, integrated with respect to time. In context, Georgakis (2013) indicated that a linear change in temperature in every run of the experiment promoted the conversion of the compound. A quadratic change in temperature has also been investigated, but did not provide any benefit.

In addition to the model in (1.7), different batch times can be considered. Batch reactor examples with different batch times are mentioned in the work by Bajpai and Reuss (1980) and Riascos and Pinto (2004). The duration of the batch can be added to the model as a scalar factor x_{i2} of different levels; see Georgakis (2013). Subsequently, the experiment is modelled as a functional linear model that depends on a profile factor and a scalar factor.

1.4 Outline of the thesis

The next chapter, Chapter 2, discusses optimal experimental designs under the frequentist and the Bayesian approach; along with a brief introduction to Bayesian inference and exchange algorithms. Chapter 3 introduces experimentation with profile factors. The simpler form of the functional linear model is stated at first, followed by variations of the form of the functional linear model, discussed by several scientists to consider polynomials and interactions. The challenges confronted are defined, and the use of basis functions to restrict the function space of functions and encounter the challenges is discussed. A review of the previous work on experimentation for models that involve profile factors is also provided. Next, Chapter 4 introduces the reader to polynomial splines and a number of basis systems are described. Primary focus is given on the B-spline basis.

A new methodology for design of experiments for functional linear and functional generalised linear models, using basis function expansions, is developed in Chapter 5 and Chapter 7, respectively. In these chapters, interactions and polynomials are modelled through univariate parameter functions, integrated on single integrals. The methodology is applied to find optimal designs for functions of profile factors in Chapter 6 and Chapter 7. Sensitivity studies are conducted and the settings of experimentation including the number of runs and the settings of profile factors and functional parameters are varied. Also, the sensitivity of the optimal designs to variations of the settings of experimentation is investigated. The software code developed to find optimal designs throughout the thesis has been used to build an R package called **fdesigns**. The **fdesigns** package is able to find optimal experimental designs for functional models. The R functions included in the package are described and demonstrated in Chapter 8. Related future goals and potential research directions are suggested in Chapter 9, including a preliminary investigation to expand the methodology to functional models for which interactions and polynomials are modelled through multivariate time index parameter functions, integrated on multiple integrals.

To improve the ease of understanding, a glossary of notation is available in the list of symbols and abbreviations are listed in the list of abbreviations. Figures and tables are also listed in the list of figures and the list of tables. Finally, any supplementary work is available as appendices at the end of the thesis.

Chapter 2

Optimal experimental designs

An experimental design represents the settings under which an experiment takes place, to obtain the responses and estimate the unknown parameters of interest. Designs that provide the maximum possible information are defined as optimal experimental designs; see Wu and Hamada (2011), Montgomery (2017) and Dean et al. (2017). To arrive in optimal experimental designs, it is important to know the purpose of the experiment because the purpose of the experiment needs to be reflected in the experimental designs. This is usually achieved through the use of design criteria, many of which, but not limited to, are known as "alphabetic" optimality criteria. Famous "alphabetic" optimality criteria include the A-, D-, E-, G- and T- optimality, and they are usually convex functions of the Fisher information matrix. The "alphabetic" optimality criteria are defined and reviewed by Atkinson et al. (2007, Chapter 10). In this thesis, focus is given on the A- and D- optimality.

The frequentist approach for optimal experimental designs is discussed in Section 2.1, and the A- and D- optimality are defined in Section 2.2. Prior to the Bayesian approach, Bayesian inference is discussed in Section 2.3, to assist the understanding of finding Bayesian optimal experimental designs. The Bayesian approach for optimal experimental designs, and utility functions that represent the experimental aim, are discussed in Section 2.4. The coordinate exchange algorithm, that is used for finding optimal experimental designs throughout the thesis, is described in Section 2.6.

2.1 Frequentist optimal experimental designs

In the traditional design of experiments, a model representing the relationship between the responses and the controllable scalar factors is the linear model, as in (1.1), and in matrix form, as in (1.4). The parameter estimates are derived using the ordinary least squares (OLS) method, which minimises the residual sum of squares,

$$RSS = (\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}). \tag{2.1}$$

The least squares estimator of $\boldsymbol{\beta}$ is,

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{F}^T\boldsymbol{y}, \tag{2.2}$$

with variance-covariance matrix,

$$\begin{aligned} \mathrm{Var}(\hat{\boldsymbol{\beta}}) &= \mathrm{Var}\left[(\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{F}^T\boldsymbol{y}\right] \\ &= \sigma^2(\boldsymbol{F}^T\boldsymbol{F})^{-1}. \end{aligned} \tag{2.3}$$

The matrix $(1/\sigma^2)\boldsymbol{F}^T\boldsymbol{F}$ is called the Fisher information matrix. The Fisher information is a measure for the amount of information about parameters provided by experimental data. It is a well-established characteristic of an experimental design used to assess and optimize the design for maximizing the expected accuracy of parameter estimates. Optimality functions including the A- and D- optimality are functions of (2.3) and equivalently of the Fisher information matrix. Thus, an analytic derivation of the variance-covariance matrix of the parameter estimates is useful. Under the assumption that the errors follow a normal distribution, that it is assumed later in Section 2.3.1, the above results can be identified from the likelihood perspective.

Independent of the choice of optimality criteria, an optimal design is defined as the solution of an optimisation problem; see Fedorov (2010). The optimisation problem is either a minimisation or a maximisation problem, depending on the objective of the experiment. An optimal design for a minimisation optimisation problem, is a design $\boldsymbol{X}^* \in \mathcal{X}$ for which,

$$\Psi(\boldsymbol{X}^*) = \min_{\boldsymbol{X} \in \mathcal{X}} \Psi(\boldsymbol{X}), \tag{2.4}$$

where $\mathcal{X}$ is the design space, i.e., the set of all applicable designs. Similarly, an optimal design for a maximisation optimisation problem, is a design $\boldsymbol{X}^* \in \mathcal{X}$ for which,

$$\Psi(\boldsymbol{X}^*) = \max_{\boldsymbol{X} \in \mathcal{X}} \Psi(\boldsymbol{X}). \tag{2.5}$$

For most problems there is not a unique optimal design to be found. For this reason, scientists usually refer to an optimal design instead of the optimal design. Moreover, optimal designs are usually called locally optimal designs. This is to highlight that for certain models, i.e., non-linear models, the Fisher information matrix and so, the design, may depend on the unknown parameters; see Dragalin et al. (2008). To impose some prior specification on the parameters (and drop the dependency on the unknown parameters if it exists), a Bayesian approach is usually followed; see Chaloner and

Verdinelli (1995). Experimental designs for non-linear models are discussed in more details in Section 2.5.

2.2 A- and D- optimality

The A-optimality objective criterion is a function of the Fisher information matrix and it has a straightforward statistical interpretation. The objective function for A-optimality is defined as the trace of the inverse of the information matrix,

$$\Psi_A(\boldsymbol{X}) = \operatorname{tr}\left[(\boldsymbol{F}^T\boldsymbol{F})^{-1}\right], \tag{2.6}$$

and a design $\boldsymbol{X}^* \in \mathcal{X}$ that minimises the A-optimality objective function,

$$\Psi_A(\boldsymbol{X}^*) = \min_{\boldsymbol{X}\in\mathcal{X}} \operatorname{tr}\left[(\boldsymbol{F}^T\boldsymbol{F})^{-1}\right], \tag{2.7}$$

is called an A-optimal design. Thus, A-optimal designs minimise the total variance of the parameter estimates or equivalently the average variance of the parameter estimators.

The D-optimality objective criterion is also a function of the Fisher information matrix. According to Atkinson et al. (2007, Chapter 10), D-optimality is the most important objective criterion in experimental design applications. The objective function for D-optimality is defined as the determinant of the information matrix raised to the power $1/Q$,

$$\Psi_D(\boldsymbol{X}) = \det(\boldsymbol{F}^T\boldsymbol{F})^{1/Q} = |\boldsymbol{F}^T\boldsymbol{F}|^{1/Q}, \tag{2.8}$$

and a design $\boldsymbol{X}^* \in \mathcal{X}$ that maximises the D-optimality objective function,

$$\Psi_D(\boldsymbol{X}^*) = \max_{\boldsymbol{X}\in\mathcal{X}} |\boldsymbol{F}^T\boldsymbol{F}|^{1/Q}, \tag{2.9}$$

is called a D-optimal design. D-optimal designs equivalently minimise the determinant of the variance-covariance matrix of the parameter estimators, or the volume of a confidence ellipsoid for the parameter estimators, i.e., provides the best accuracy of the parameter estimators. Moreover, D-optimality considers the covariance between the estimators, unlike A-optimality which does not.

There exists many more criteria, with many, but not all of them, depending on the assumed model through the information matrix. In addition, for simple models, a design may be optimal under multiple objective criteria, even though this is not usually the case; see Jones et al. (2021). Also, a design that is optimal with respect to one model is usually not optimal with respect to another model. Several other objective criteria, as well as further details on A- and D- optimality are available in the book by Atkinson et al. (2007, Chapters 6, 9, and 10).

2.3 Bayesian inference

Bayesian statistics, named after Thomas Bayes, provide a different approach to statistical inference. The basic theory underlying Bayesian inference is that the only measure of uncertainty is probability; see Gelman et al. (2013, Chapter 1). In other words, Bayesian inference is a statistical approach that uses subjective probability statements in order to quantify uncertainties. A general notation in Bayesian inference is that $\pi(\cdot|\cdot)$ represents a conditional probability distribution, $\pi(\cdot)$ represents a marginal distribution, and $\pi(\cdot,\cdot)$ represents a joint probability distribution. A joint probability distribution is expanded as the product of two distributions, for example,

$$\pi(\boldsymbol{\beta}, \boldsymbol{y}) = \pi(\boldsymbol{\beta})\pi(\boldsymbol{y}|\boldsymbol{\beta}). \tag{2.10}$$

Data are assumed to come from a family of distribution and they are presented as a likelihood function $\pi(\boldsymbol{y}|\boldsymbol{\beta})$, exactly as in the classical approach. The likelihood function is a function of the parameters for observed data, and is a measure of support for the parameters given the observed data. In simple words, it represents how likely the observed data are, given the unknown parameters $\boldsymbol{\beta}$. Unlike the classical approach, where the unknown parameters $\boldsymbol{\beta}$ are considered as constant, in Bayesian statistics the unknown parameters are considered as random. Prior to obtaining any observed data it is assumed that there exists some information about the unknown parameters represented in the form of a prior distribution $\pi(\boldsymbol{\beta})$. The likelihood function and the prior distribution are used to develop the posterior distribution $\pi(\boldsymbol{\beta}|\boldsymbol{y})$, which represents the modified or updated belief for the true value of the parameters in light of the observed data. Derivation of the posterior distribution which is a probability statement about $\boldsymbol{\beta}$ given $\boldsymbol{y}$ requires the use of the Bayes' rule and the property of conditional probability; see Gelman et al. (2013, Chapter 1) and Bolstad and Curran (2016, Chapter 4),

$$\begin{aligned}
\pi(\boldsymbol{\beta}|\boldsymbol{y}) &= \frac{\pi(\boldsymbol{\beta}, \boldsymbol{y})}{\pi(\boldsymbol{y})} \\
&= \frac{\pi(\boldsymbol{\beta})\pi(\boldsymbol{y}|\boldsymbol{\beta})}{\pi(\boldsymbol{y})} \\
&= \frac{\pi(\boldsymbol{\beta})\pi(\boldsymbol{y}|\boldsymbol{\beta})}{\int_{-\infty}^{\infty} \pi(\boldsymbol{\beta})\pi(\boldsymbol{y}|\boldsymbol{\beta})\, d\boldsymbol{\beta}} \\
&\propto \pi(\boldsymbol{\beta})\pi(\boldsymbol{y}|\boldsymbol{\beta}).
\end{aligned} \tag{2.11}$$

The last line in (2.11) corresponds to the unnormalised posterior distribution, and it omits the denominator $\pi(\boldsymbol{y})$ because it does not depend on the parameters $\boldsymbol{\beta}$. Equivalently, this means that for some models the posterior distribution is known up to a normalising constant.

Finally, suppose that inference is not of immediate interest on every parameter. The parameters that are not of interest are called nuisance parameters; see Gelman et al. (2013, Chapter 3). For example, suppose that $\boldsymbol{\beta}^T = (\beta_1 \ \beta_2)$, and that the parameter of interest is β_1, then β_2 is a nuisance parameter. A distribution of the parameter of interest given the data, is called a marginal posterior distribution and it is derived by integrating the posterior distribution with respect to the nuisance parameters. For instance, a marginal posterior distribution for β_1 is,

$$\pi(\beta_1|\boldsymbol{y}) = \int \pi(\boldsymbol{\beta}|\boldsymbol{y}) \, d\beta_2. \tag{2.12}$$

Further elaboration on Bayesian inference is given in the books by Gelman et al. (2013) and Box and Tiao (1992).

2.3.1 Likelihood, Joint Prior and Joint Posterior for the linear model

For the standard linear model in (1.4), assuming $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n)$, the likelihood function is $\boldsymbol{y} \sim N(\boldsymbol{F\beta}, \sigma^2 \boldsymbol{I}_n)$, expanded as,

$$\begin{aligned} \pi(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2) &= (2\pi)^{-n/2} |\sigma^2 \boldsymbol{I}_n|^{-1/2} \exp\Big(-\frac{1}{2\sigma^2} (\boldsymbol{y} - \boldsymbol{F\beta})^T (\boldsymbol{y} - \boldsymbol{F\beta}) \Big) \\ &\propto (\sigma^2)^{-n/2} \exp\Big(-\frac{1}{2\sigma^2} (\boldsymbol{y} - \boldsymbol{F\beta})^T (\boldsymbol{y} - \boldsymbol{F\beta}) \Big). \end{aligned} \tag{2.13}$$

For certain prior choices, called conjugate prior choices, the posterior end up follow the same distribution family. This is useful because it is easy to interpret the change in parameters from the prior to the posterior. A conjugate prior choice for the unknown parameters $\boldsymbol{\beta}$ and σ^2 is a normal inverse gamma distribution,

$$\pi(\boldsymbol{\beta}, \sigma^2) = \pi(\boldsymbol{\beta}|\sigma^2) \pi(\sigma^2),$$

with $\pi(\boldsymbol{\beta}|\sigma^2) \sim N(\boldsymbol{\mu}, \sigma^2 \boldsymbol{V})$ and $\pi(\sigma^2) \sim IG(a/2, b/2)$. The $Q \times 1$ vector $\boldsymbol{\mu}$ is the prior mean of $\boldsymbol{\beta}$, the $Q \times Q$ matrix $\boldsymbol{V}$ is a known and symmetric matrix, and a, b are known hyperparameters. Explicitly, the prior distributions are given by,

$$\begin{aligned} \pi(\boldsymbol{\beta}|\sigma^2) &= (2\pi)^{-Q/2} |\sigma^2 \boldsymbol{V}^{-1}|^{-1/2} \exp\Big(-\frac{1}{2\sigma^2} (\boldsymbol{\beta} - \boldsymbol{\mu})^T \boldsymbol{V}^{-1} (\boldsymbol{\beta} - \boldsymbol{\mu}) \Big) \\ &\propto (\sigma^2)^{-Q/2} \exp\Big(-\frac{1}{2\sigma^2} (\boldsymbol{\beta} - \boldsymbol{\mu})^T \boldsymbol{V}^{-1} (\boldsymbol{\beta} - \boldsymbol{\mu}) \Big), \end{aligned}$$

and

$$\begin{aligned}\pi(\sigma^2) &= \frac{(b/2)^{\frac{a}{2}}}{\Gamma(\frac{a}{2})} \frac{1}{(\sigma^2)^{\frac{a}{2}+1}} \exp\Big(- \frac{b}{2\sigma^2}\Big)\\ &\propto (\sigma^2)^{-(\frac{a}{2}+1)} \exp\Big(- \frac{b}{2\sigma^2}\Big).\end{aligned}$$

A joint prior distribution of $\boldsymbol{\beta}$ and σ^2 is,

$$\begin{aligned}\pi(\boldsymbol{\beta},\sigma^2) &= \pi(\boldsymbol{\beta}|\sigma^2)\pi(\sigma^2)\\ &\propto (\sigma^2)^{-(\frac{a+Q}{2}+1)} \exp\Bigg(- \frac{1}{2\sigma^2}\Big((\boldsymbol{\beta}-\boldsymbol{\mu})^T\boldsymbol{V}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}) + b\Big)\Bigg),\end{aligned} \tag{2.14}$$

which is a normal inverse gamma distribution such that,

$$\pi(\boldsymbol{\beta},\sigma^2) \sim NIG(\boldsymbol{\mu},\boldsymbol{V},a/2,b/2). \tag{2.15}$$

Using Bayes theorem that was discussed in Section 2.3, the joint prior in (2.14) and the likelihood in (2.13) are combined to derive the joint posterior distribution,

$$\begin{aligned}\pi(\boldsymbol{\beta},\sigma^2|\boldsymbol{y}) &\propto \pi(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\pi(\boldsymbol{\beta},\sigma^2)\\ &\propto (\sigma^2)^{-(\frac{a^*+Q}{2}+1)} \exp\Bigg(- \frac{1}{2\sigma^2}\Big((\boldsymbol{\beta}-\boldsymbol{\beta}_N)^T\boldsymbol{V}_N^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_N) + b^*\Big)\Bigg),\end{aligned}$$

which is also a normal inverse gamma distribution such that,

$$\pi(\boldsymbol{\beta},\sigma^2|\boldsymbol{y}) \sim NIG(\boldsymbol{\beta}_N,\boldsymbol{V}_N,a^*/2,b^*/2), \tag{2.16}$$

where,

$$\begin{aligned}\boldsymbol{V}_N &= (\boldsymbol{F}^T\boldsymbol{F} + \boldsymbol{V}^{-1})^{-1},\\ \boldsymbol{\beta}_N &= \boldsymbol{V}_N(\boldsymbol{V}^{-1}\boldsymbol{\mu} + \boldsymbol{F}^T\boldsymbol{y}),\\ a^* &= a+n,\\ b^* &= b + (\boldsymbol{\mu}^T\boldsymbol{V}^{-1}\boldsymbol{\mu} + \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{\beta}_N^T\boldsymbol{V}_N^{-1}\boldsymbol{\beta}_N).\end{aligned} \tag{2.17}$$

2.3.2 Marginal posterior distributions for the linear model

The marginal posterior distribution for $\boldsymbol{\beta}$ is obtained by integrating the posterior with respect to σ^2 which takes the form of a nuisance parameter,

$$\begin{aligned}
\pi(\boldsymbol{\beta}|\boldsymbol{y}) &= \int \pi(\boldsymbol{\beta},\sigma^2|\boldsymbol{y})\; d\sigma^2 \\
&\propto \int (\sigma^2)^{-(\frac{a^*+Q}{2}+1)} \exp\Big(-\frac{1}{2\sigma^2}\Big[(\boldsymbol{\beta}-\boldsymbol{\beta}_N)^T \boldsymbol{V}_N^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_N)+b^*\Big]\Big)\; d\sigma^2 \\
&\propto \left[1+\frac{(\boldsymbol{\beta}-\boldsymbol{\beta}_N)^T \boldsymbol{V}_N^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_N)}{b^*}\right]^{-\frac{a^*+Q}{2}}.
\end{aligned} \tag{2.18}$$

Hence, the marginal posterior distribution for $\boldsymbol{\beta}$ is a multivariate t-distribution,

$$\pi(\boldsymbol{\beta}|\boldsymbol{y}) \sim t_{a^*}\Big(\boldsymbol{\beta}_N, \frac{b^*}{a^*}\boldsymbol{V}_N\Big) \tag{2.19}$$

with a^* degrees of freedom, mean $\boldsymbol{\beta}_N$, and scale $\frac{b^*}{a^*}\boldsymbol{V}_N$, for $a^*, b^*, \boldsymbol{\beta}_N$, and $\boldsymbol{V}_N$ as in (2.17).

The marginal posterior distribution for σ^2 is obtained by integrating the posterior with respect to $\boldsymbol{\beta}$, which takes the form of a nuisance parameter,

$$\begin{aligned}
\pi(\sigma^2|\boldsymbol{y}) &= \int \pi(\boldsymbol{\beta},\sigma^2|\boldsymbol{y})\; d\boldsymbol{\beta} \\
&\propto \int (\sigma^2)^{-(\frac{a^*+Q}{2}+1)} \exp\Big(-\frac{1}{2\sigma^2}\Big[(\boldsymbol{\beta}-\boldsymbol{\beta}_N)^T \boldsymbol{V}_N^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_N)+b^*\Big]\Big)\; d\boldsymbol{\beta} \\
&= (\sigma^2)^{-(\frac{a^*}{2}+1)} \exp\Big(\frac{-b^*}{2\sigma^2}\Big) \\
&\qquad \times \int (\sigma^2)^{-\frac{Q}{2}} \exp\Big(-\frac{1}{2\sigma^2}\Big[(\boldsymbol{\beta}-\boldsymbol{\beta}_N)^T \boldsymbol{V}_N^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_N)\Big]\Big)\; d\boldsymbol{\beta} \\
&\propto (\sigma^2)^{-(\frac{a^*}{2}+1)} \exp\Big(\frac{-b^*}{2\sigma^2}\Big).
\end{aligned} \tag{2.20}$$

The integrand in the third equality of (2.20) is proportional to the density of a $N\big(\boldsymbol{\beta}_N, \sigma^2\boldsymbol{V}_N\big)$ distribution, which integrates to one. Hence, the marginal posterior distribution for σ^2 is an inverse gamma distribution,

$$\pi(\sigma^2|\boldsymbol{y}) \sim IG(a^*/2, b^*/2), \tag{2.21}$$

for a^* and b^* as in (2.17).

The marginal distributions of $\boldsymbol{y}$, $\boldsymbol{y}$ conditional on $\boldsymbol{\beta}$, and $\boldsymbol{y}$ conditional on σ^2, used later in the chapter, are derived in Appendix B in Sections B.3, B.1 and B.2 respectively.

2.4 Bayesian optimal experimental designs

Design of experiments is a process taking place prior to the experiment, thus, prior to the data collection. Thus, experimental designs fit naturally in the Bayesian framework. The experimental aim in the Bayesian framework is represented through a utility function $u(\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{X})$. A utility function defines the gain of the experimenter from using the design $\boldsymbol{X}$, to obtain responses $\boldsymbol{y}$, assuming values for the parameters $\boldsymbol{\beta}$. Thus, a Bayesian optimal design is a design $\boldsymbol{X}^* \in \mathcal{X}$, that is maximising the expected utility with respect to the joint distribution of the unknown responses and unknown parameters; see Chaloner and Verdinelli (1995),

$$\begin{aligned}
\Psi(\boldsymbol{X}) &= \mathbb{E}_{\boldsymbol{\beta},\boldsymbol{y}}[u(\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{X})] \\
&= \int_{\boldsymbol{\beta}} \int_{\boldsymbol{y}} u(\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{X}) \pi(\boldsymbol{\beta}, \boldsymbol{y} | \mathbf{X}) \; d\boldsymbol{y} \; d\boldsymbol{\beta} \\
&= \int_{\boldsymbol{\beta}} \int_{\boldsymbol{y}} u(\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{X}) \pi(\boldsymbol{\beta} | \boldsymbol{y}, \boldsymbol{X}) \pi(\boldsymbol{y} | \boldsymbol{X}) \; d\boldsymbol{y} \; d\boldsymbol{\beta} \\
&= \int_{\boldsymbol{\beta}} \int_{\boldsymbol{y}} u(\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{X}) \pi(\boldsymbol{y} | \boldsymbol{\beta}, \boldsymbol{X}) \pi(\boldsymbol{\beta}) \; d\boldsymbol{y} \; d\boldsymbol{\beta}.
\end{aligned} \tag{2.22}$$

The unknown parameters are assigned prior distributions according to the available information and beliefs. For instance, $\pi(\boldsymbol{\beta}|\boldsymbol{X})$ corresponds to the available information of the parameters prior to the experiment, i.e., a prior distribution of the unknown parameters. A combination of the joint prior and the likelihood using Bayes theorem allows the posterior and marginal densities to be found; see Section 2.3. Common utility functions used in the Bayesian framework for optimal experimental designs are the Negative Squared Error Loss (NSEL) and the Shannon Information Gain (SIG). Both utility functions are discussed in the next subsections, and optimality criteria for the linear model are derived.

2.4.1 Negative Squared Error Loss

NSEL is a common utility function in the Bayesian framework. It is a utility function in quadratic form such that,

$$u(\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{X}) = -\left[\boldsymbol{\beta} - \mathbb{E}(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X})\right]^T \left[\boldsymbol{\beta} - \mathbb{E}(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X})\right],$$

and optimal designs are the designs that maximise the expected NSEL. The expected utility is derived by taking the expectation over the responses and the parameters. For the linear model, the expected utility is,

$$\begin{aligned}
\Psi_{nsel}(\boldsymbol{X}) &= \mathbb{E}_{\boldsymbol{y},\boldsymbol{\beta},\sigma^2}\Big(-\left[\boldsymbol{\beta} - \mathbb{E}(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X})\right]^T\left[\boldsymbol{\beta} - \mathbb{E}(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X})\right]\Big) \\
&= \mathbb{E}_{\boldsymbol{y},\sigma^2}\left(\mathbb{E}_{\boldsymbol{\beta}|\boldsymbol{y},\sigma^2}\Big[-\sum_{q=1}^{Q}\left[\beta_q - \mathbb{E}(\beta_q|\boldsymbol{y},\boldsymbol{X})\right]^2\Big]\right) \\
&= \mathbb{E}_{\boldsymbol{y},\sigma^2}\Big[-\sum_{q=1}^{Q}\operatorname{var}_{\boldsymbol{\beta}|\boldsymbol{y},\sigma^2}(\beta_q)\Big] \\
&= \mathbb{E}_{\boldsymbol{y},\sigma^2}\Big[-\operatorname{tr}\big(\sigma^2\boldsymbol{V}_N\big)\Big] \\
&= -\frac{b}{a-2}\operatorname{tr}\big[\big(\boldsymbol{F}^T\boldsymbol{F} + \boldsymbol{V}^{-1}\big)^{-1}\big].
\end{aligned} \tag{2.23}$$

The objective function in (2.23), for the linear model, is also known as Bayesian A-optimality, and a design that maximises the expected utility is defined as a Bayesian A-optimal design. This criterion is proportional to the trace of the posterior variance-covariance matrix,

$$\Psi_{nsel}(\boldsymbol{X}) \propto -\operatorname{tr}\big[\big(\boldsymbol{F}^T\boldsymbol{F} + \boldsymbol{V}^{-1}\big)^{-1}\big]. \tag{2.24}$$

and a design $\boldsymbol{X}^*$ is optimal if it minimises the trace of the posterior variance-covariance matrix,

$$\Psi_{nsel}(\boldsymbol{X}^*) = \min_{\boldsymbol{X}\in\mathcal{X}} \operatorname{tr}\big[\big(\boldsymbol{F}^T\boldsymbol{F} + \boldsymbol{V}^{-1}\big)^{-1}\big]. \tag{2.25}$$

Under a non-informative prior where $\boldsymbol{V}^{-1}$ would be zero, this criterion would minimise the trace of the inverse of the information matrix as in (2.7).

2.4.2 Shannon Information Gain

Another important and well known utility function is the Shannon Information Gain (SIG). In this section, the SIG on the marginal distribution of $\boldsymbol{\beta}$ is considered. However, the procedure followed and the results shown are identical for the SIG on the joint distribution of $\boldsymbol{\beta}$ and σ^2. The SIG utility function is defined as,

$$\begin{aligned}
u(\boldsymbol{\beta},\boldsymbol{y},\boldsymbol{X}) &= \log\pi(\boldsymbol{\beta}|\boldsymbol{y}) - \log\pi(\boldsymbol{\beta}) \\
&= \log\frac{\pi(\boldsymbol{y}|\boldsymbol{\beta})\pi(\boldsymbol{\beta})}{\pi(\boldsymbol{y})} - \log\pi(\boldsymbol{\beta}) \\
&= \log\pi(\boldsymbol{y}|\boldsymbol{\beta}) + \log\pi(\boldsymbol{\beta}) - \log\pi(\boldsymbol{y}) - \log\pi(\boldsymbol{\beta}) \\
&= \log\pi(\boldsymbol{y}|\boldsymbol{\beta}) - \log\pi(\boldsymbol{y}),
\end{aligned} \tag{2.26}$$

with any rearrangements resulting from an application of Bayes' theorem from (2.11). Compared to the first, the last equation of (2.26) is sometimes more useful for computations. To begin with, the marginal distributions involved are,

$$\boldsymbol{y} \sim t_a\Big(\boldsymbol{F\mu}, \frac{b}{a}(\boldsymbol{I} + \boldsymbol{FVF}^T)\Big),$$

$$\boldsymbol{y}|\boldsymbol{\beta} \sim t_a\Big(\boldsymbol{F\beta}, \frac{b}{a}\boldsymbol{I}\Big),$$

from (B.7) and (B.2), available in Appendix B in Sections B.3 and B.1, respectively. An optimal design is a design that maximises the expected utility of SIG. The expected utility is derived by taking the expectation over the responses and the parameter $\boldsymbol{\beta}$. For the linear model, the expected utility is,

$$\begin{aligned}\Psi_{sig}(\boldsymbol{X}) &= \mathbb{E}_{\boldsymbol{y},\boldsymbol{\beta}}\big(\log \pi(\boldsymbol{y}|\boldsymbol{\beta}) - \log \pi(\boldsymbol{y})\big)\\ &= \frac{1}{2}\log|\boldsymbol{I} + \boldsymbol{FVF}^T| - \underbrace{(\frac{a+n}{2})\mathbb{E}_{\boldsymbol{y},\boldsymbol{\beta}}\left[\log\left[1 + \frac{1}{a}(\boldsymbol{y} - \boldsymbol{F\beta})^T\Big(\frac{b}{a}\Big)^{-1}(\boldsymbol{y} - \boldsymbol{F\beta})\right]\right]}_{(1)}\\ &\quad + \underbrace{(\frac{a+n}{2})\mathbb{E}_{\boldsymbol{y}}\left[\log\left[1 + \frac{1}{a}(\boldsymbol{y} - \boldsymbol{F\mu})^T\Big[\frac{b}{a}(\boldsymbol{I} + \boldsymbol{FVF}^T)\Big]^{-1}(\boldsymbol{y} - \boldsymbol{F\mu})\right]\right]}_{(2)}.\end{aligned}$$

Working out (1) and (2),

$$\begin{aligned}(1) : (\frac{a+n}{2})&\mathbb{E}_{\boldsymbol{y},\boldsymbol{\beta}}\left[\log\left[1 + \frac{1}{a}(\boldsymbol{y} - \boldsymbol{F\beta})^T\Big(\frac{b}{a}\Big)^{-1}(\boldsymbol{y} - \boldsymbol{F\beta})\right]\right]\\ &= (\frac{a+n}{2})\mathbb{E}_{\boldsymbol{\beta}}\left[\mathbb{E}_{\boldsymbol{y}|\boldsymbol{\beta}}\Big[\log\Big[1 + \frac{n}{a}\frac{(\boldsymbol{y} - \boldsymbol{F\beta})^T\Big(\frac{b}{a}\Big)^{-1}(\boldsymbol{y} - \boldsymbol{F\beta})}{n}\Big]\Big]\right]\\ &= (\frac{a+n}{2})\mathbb{E}\left[\log\left[1 + \frac{n}{a}\mathcal{F}\right]\right],\end{aligned}$$

where $\mathcal{F} \sim \mathcal{F}(n, a)$ distribution with n and a degrees of freedom.

$$\begin{aligned}(2) : (\frac{a+n}{2})&\mathbb{E}_{\boldsymbol{y},\sigma^2,\boldsymbol{\beta}}\left[\log\left[1 + \frac{1}{a}(\boldsymbol{y} - \boldsymbol{F\mu})^T\Big[\frac{b}{a}(\boldsymbol{I} + \boldsymbol{FVF}^T)\Big]^{-1}(\boldsymbol{y} - \boldsymbol{F\mu})\right]\right]\\ &= (\frac{a+n}{2})\mathbb{E}\left[\log\left[1 + \frac{n}{a}\mathcal{F}\right]\right],\end{aligned}$$

where $\mathcal{F} \sim \mathcal{F}(n, a)$ distribution with n and a degrees of freedom as before. The result in the calculations of (1) and (2) follow from the theorem in Kotz and Nadarajah (2004, p. 19), that if:

$$\boldsymbol{y} \sim t_a\Big(\boldsymbol{F\mu}, \frac{b}{a}(\boldsymbol{I} + \boldsymbol{FVF}^T)\Big),$$

then,

$$\frac{(\boldsymbol{y}-\boldsymbol{F\mu})^T\left[\frac{b}{a}(\boldsymbol{I}+\boldsymbol{FVF}^T)\right]^{-1}(\boldsymbol{y}-\boldsymbol{F\mu})}{n} \sim \mathcal{F}(n,a),$$

which does not depend on the design. Since (1) and (2) do not depend on the design, this criterion is equivalent to the determinant of the inverse of the posterior variance-covariance matrix,

$$\Psi_{sig}(\boldsymbol{X}) = |\boldsymbol{F}^T\boldsymbol{F}+\boldsymbol{V}^{-1}|, \tag{2.27}$$

and a design $\boldsymbol{X}^*$ is optimal if it maximises the determinant of the inverse of the posterior variance-covariance matrix,

$$\Psi_{sig}(\boldsymbol{X}^*) = \max_{\boldsymbol{X}\in\mathcal{X}} |\boldsymbol{F}^T\boldsymbol{F}+\boldsymbol{V}^{-1}|. \tag{2.28}$$

The objective function in (2.28), for the linear model, is known as Bayesian D-optimality and a design that achieves to maximise the expected utility is defined as a Bayesian D-optimal design. Under a non-informative prior where $\boldsymbol{V}^{-1}$ would be zero, this criterion maximises the determinant of the information matrix $\boldsymbol{F}^T\boldsymbol{F}$.

2.5 Optimal designs for non-linear models

An important feature of designs for non-linear models, including generalised linear models (GLMs) that are discussed in Chapter 7, is that they depend on the parameters. This is because the Fisher information matrix is a function of the unknown parameters $\boldsymbol{\beta}$. As a consequence, the design also depends on the unknown parameters, thus, identifying optimal designs for non-linear models is not straightforward. For this reason, in order to identify optimal experimental designs for non-linear models including GLMs, or generally for models for which the information matrix depends on the unknown parameters, prior information of the model parameters is required.

This problem can be approached in different ways, including locally optimal designs, sequential designs, maximin designs, and pseudo-Bayesian optimal designs; see Atkinson and Woods (2015). In this thesis, the approach that is followed to incorporate the prior information into the model and define the optimality objective functions in Chapter 7 is the pseudo-Bayesian approach; see Chaloner and Verdinelli (1995), Overstall and Woods (2017), and Woods et al. (2017).

In the pseudo-Bayesian approach, it is assumed that the unknown parameters $\boldsymbol{\beta}$ follow a prior distribution $\pi(\boldsymbol{\beta})$. After that, optimal designs are the designs that achieve to minimise the expectation of the objective functions with respect to the prior distribution of the parameters. Assuming a general objective function $\psi(\boldsymbol{\beta},\boldsymbol{X})$ that is a function of the parameters, a pseudo-Bayesian optimal design minimises the expectation of the objective function with respect to the prior of the parameters,

$$\Psi(\boldsymbol{X}) = \mathbb{E}_{\boldsymbol{\beta}}\Big\{\psi(\boldsymbol{\beta},\boldsymbol{X})\Big\} = \int_{\boldsymbol{\Theta}} \psi(\boldsymbol{\beta},\boldsymbol{X})\ \pi(\boldsymbol{\beta})\ d\boldsymbol{\beta}, \tag{2.29}$$

with $\boldsymbol{X}$ the design matrix from the basis expansion of the profile factors, and $\boldsymbol{\Theta}$ the parameter space. Specifically, the pseudo-Bayesian A- and D- optimality objective functions are defined as,

$$\begin{aligned}\Psi_A(\boldsymbol{X}) &= \mathbb{E}_{\boldsymbol{\beta}}\Big\{\mathrm{tr}\big[\mathcal{I}(\boldsymbol{\beta},\boldsymbol{X})^{-1}\big]\Big\}\\ &= \int_{\boldsymbol{\Theta}} \mathrm{tr}\big[\mathcal{I}(\boldsymbol{\beta},\boldsymbol{X})^{-1}\big]\pi(\boldsymbol{\beta})\ d\boldsymbol{\beta}\end{aligned} \tag{2.30}$$

$$\begin{aligned}\Psi_D(\boldsymbol{X}) &= \mathbb{E}_{\boldsymbol{\theta}}\Big\{\det\big[(\mathcal{I}(\boldsymbol{\theta},\boldsymbol{X})\big]^{-1/Q}\Big\}\\ &= \int_{\boldsymbol{\Theta}} \det\big[(\mathcal{I}(\boldsymbol{\beta},\boldsymbol{X})\big]^{-1/Q}\pi(\boldsymbol{\beta})\ d\boldsymbol{\beta}\end{aligned} \tag{2.31}$$

with Q the total number of parameters, and $\mathcal{I}(\boldsymbol{\theta},\boldsymbol{X})$ the information matrix that depends on the parameters. The designs that minimise the pseudo-Bayesian A- and D-optimality objective functions are known as pseudo-Bayesian A- and D- optimal designs, respectively; see Chaloner and Verdinelli (1995), Woods et al. (2006), and Van De Ven and Woods (2014).

2.6 Exchange algorithms

Optimisation refers to the ability to formulate a problem in mathematical terms, and find the best allocation of limited resources, aiming to maximise or minimise an objective function. It is considered a development in Mathematics with origins back in 1947; see Dantzig (2002). Since then, it received great attention. Several approaches and algorithms exist in the Statistics literature for finding optimal designs, however, most statistical software packages shifted to the computational efficient use of exchange algorithms; see Meyer and Nachtsheim (1995), Gotwalt et al. (2009), and Cuervo et al. (2016). The optimal designs presented in examples in later chapters are optimised using the coordinate exchange algorithm followed by the point exchange algorithm; see Atkinson et al. (2007, Chapter 12). Both algorithms are discussed in the following sections.

2.6.1 Coordinate exchange algorithm

The coordinate exchange algorithm builds an optimal design using the following steps:

1. The coordinate exchange algorithm begins with a randomly generated design, called the initial design. The initial design must satisfy the lower and upper bounds,

   $$x_{ij} \in [u, v], \quad i = 1, 2, \ldots, n \; j = 1, 2, \ldots, J,$$

   for x_{ij} the ij^{th} entry of the design matrix $\boldsymbol{X}$, and u, v the scalar values for the interval of the entries of $\boldsymbol{X}$ as before.

2. The objective value $\Psi(\boldsymbol{X})$ of the design is calculated. The objective function to use depends on the choice of the objective criterion.

3. The algorithm tackles each coordinate separately and optimise it according to the objective function. This involves the optimisation of every x_{ij}, i.e., x_{ij} the ij^{th} entry of the design matrix $\boldsymbol{X}$, with i representing the row/run and j the column of the design matrix. For instance, if minimising,

   $$arg \min_{x_{ij}} \Psi(\boldsymbol{X}), \text{ subject to } x_{ij} \in [u, v], \quad i = 1, 2, \ldots, n \; j = 1, 2, \ldots, J,$$

   and if maximising,

   $$arg \max_{x_{ij}} \Psi(\boldsymbol{X}), \text{ subject to } x_{ij} \in [u, v], \quad i = 1, 2, \ldots, n \; j = 1, 2, \ldots, J.$$

   For example, if the objective function is D-optimality, then the optimisation of each point depends on the determinant of the information matrix.

4. The difference in the objective values of the designs before and after the optimisation, i.e., $\Delta\Psi(\boldsymbol{X})$, is compared to a pre-specified tolerance value, denoted as δ.

5. While $\Delta\Psi(\boldsymbol{X}) > \delta$, the algorithm repeats the same procedure from step 3, i.e., the latest design becomes the initial design, otherwise the algorithm stops.

The initial design usually affects the resulting optimal design. For this reason, repeating the algorithm for multiple starting designs increases the likelihood of finding optimal designs. Following the recommendation of Goos and Jones (2011), the results presented in the thesis are based on optimised designs using the coordinate exchange algorithm and 1000 randomly generated initial designs.

2.6.2 Point exchange algorithm

Point exchange algorithm is an exchange algorithm, that uses the standard operations of exchange algorithms, from points of a candidate list, to identify optimal designs; as in Atkinson et al. (2007, Chapter 12). The candidate list of points can be any possible combination of points.

In this thesis, the point exchange algorithm is used along the coordinate exchange, aiming to take an already optimised coefficient design matrix $\boldsymbol{X}$ as input, and try to improve it. In optimal experimental designs, replication of design points and identical runs is common; see Wu and Hamada (2011, Chapter 1). For this reason, the choice of the candidate list in this thesis contains the exact points of the optimal design found from the coordinate exchange algorithm. Thus, the candidate list of points is the same dimension as the design matrix. The combination of the coordinate exchange and the point exchange algorithms is discussed in Overstall and Woods (2017), as phase 1 and phase 2 of the design optimisation.

The procedure followed by the point exchange algorithm is described in the following steps:

1. The candidate list, which is the design points identified in the coordinate exchange, is formed.
2. Each of the n runs of the candidate list, is added to the design found from the coordinate exchange, creating $n+1$ runs designs. The objective value for those new designs is obtained, and the new design that performs better is found.
3. The new design is evaluated $n+1$ times, each time with a run deleted, to identify which run deletion performs better.
4. The new n run design is the design that performs better posterior to the swap of the design runs.
5. The process is replicated to find an optimal design, until no swaps affect the performance of the design.

2.7 Summary

This chapter has reviewed optimal experimental designs. At first, the frequentist approach has been considered. Optimality functions including the A- and D- optimality have been introduced and optimality criteria for A- and D- optimal designs have been derived. After briefly expanding on Bayesian inference, the Bayesian approach for optimal experimental designs has been described. For the linear model of normally distributed responses and normal inverse gamma prior specifications for the parameters, the posterior along with marginal distributions have been derived. Moreover, utility functions to represent the experimental aim have been discussed and Bayesian optimality criteria for Bayesian A- and D- optimal designs have been defined. At the end of the chapter, exchange algorithms used for the optimisation of the designs in later examples have been introduced.

Chapter 3

Experimentation with profile factors

In this chapter, the aim is to introduce to the reader a functional model that describes the relationship between scalar responses, and one or more profile factors or, functions of profile factors. In literature, scientists refer to such models as functional linear models or, sometimes as scalar-on-function linear models. A linear model involving profile factors in this thesis, is henceforth referred as a functional linear model (FLM). A model that represents this relationship is the FLM model with profile factors and functional parameters, integrated over time. However, the additive FLM carries two main problems for estimation and design. At first, a problem is to estimate infinite dimensional objects from finite response data. After that, a problem is to choose the functions of the profile factors from an unrestricted, and hence, very general function space.

The simpler form of the FLM is introduced in Section 3.1, followed by a review on extensions and variations of the FLM. Primarily in the thesis, attention is given to the form of the FLM that depends on functions of the profile factors, i.e., main effects, interactions, and polynomials, and single time index parameter functions, integrated over a single integral of time. This form of the model is described in Section 3.2. To overcome the problems in estimation and design, basis function expansions are applied on the functional parameters and the profile factors. After that, related literature to experimentation for models involving profile factors is reviewed and discussed in Section 3.3.

3.1 Introduction to the functional linear model

An experiment is assumed to take place from time 0 to time $\mathcal{T}$, i.e., $0 \leq t \leq \mathcal{T}$. The experiment is assumed to consist of n runs. The i^{th} run of the experiment involves specifying the controllable functions of the profile factors $\boldsymbol{x}_i(t)$ to measure the scalar responses y_i at time $\mathcal{T}$. A way of modelling the relationship between the scalar response and J profile factors, is an additive FLM. The FLM was first introduced by Ramsay and Dalzell (1991), a model that considered the main effects of the profile factors. Later, the FLM was discussed in Hastie and Mallows (1993), and written in its most commonly used, and simpler, form,

$$y_i = \int_0^{\mathcal{T}} \boldsymbol{x}_i^T(t)\ \boldsymbol{\beta}(t)\ dt + \epsilon_i, \quad i = 1, 2, \ldots, n,\ t \in [0, \mathcal{T}], \tag{3.1}$$

for i representing the i^{th} run of the experiment, and ϵ_i representing the independent and identically distributed errors with mean zero and variance σ^2. The $J \times 1$ vector $\boldsymbol{x}_i(t)$, represents the functions of the profile factors at the i^{th} run of the experiment, and it is defined as,

$$\boldsymbol{x}_i^T(t) = \begin{pmatrix} x_{i1}(t) & x_{i2}(t) & \cdots & x_{iJ}(t) \end{pmatrix}, \quad i = 1, 2, \ldots, n, \tag{3.2}$$

with each $x_{ij}(t), i = 1, 2, \ldots, n, j = 1, 2, \ldots, J$, the function of the j^{th} profile factor at the i^{th} run of the experiment. The functions of the profile factors are usually in an interval $[u, v]$, i.e., $x_{ij}(t) \in [u, v]$, with u and v scalar values. A special case of a function of a profile factor is $x_{ij}(t) = x_{ij}$, for all $t \in [0, \mathcal{T}]$, representing a scalar factor. The vector $\boldsymbol{\beta}(t)$ in (3.1), is a $J \times 1$ vector of the unknown functional parameters. Each functional parameter $\beta_j(t) : [0, \mathcal{T}] \to \mathbb{R}, j = 1, 2, \ldots, J$, is an unknown function of time $0 \leq t \leq \mathcal{T}$.

Yao and Müller (2010) proposed an updated form of the FLM, to allow polynomial effects, if the model with only main effects is not adequate. At first, they consider a functional quadratic model for a single profile factor. The adjusted quadratic FLM for multiple factors is given by,

$$\begin{aligned} y_i = \int_0^{\mathcal{T}} \boldsymbol{x}_i^T(t)\ \boldsymbol{\beta}(t)\ dt + \sum_{j=1}^{J} \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} x_{ij}(t_1)\ x_{ij}(t_2)\ \beta_j(t_1, t_2)\ dt_1 dt_2 + \epsilon_i, \\ i = 1, 2, \ldots, n,\ t, t_1, t_2 \in [0, \mathcal{T}], \end{aligned} \tag{3.3}$$

with the additional term, compared to (3.1), the quadratic part that involves the quadratic polynomial of the functions of the profile factors, and square integrable bivariate parameter functions $\beta(t_1, t_2)$. After that, they expand (3.3) to a full k^{th} order FLM for a single profile factor. The adjusted full k^{th} order FLM for multiple factors is given by,

$$\begin{aligned} y_i = \int_0^{\mathcal{T}} \boldsymbol{x}_i^T(t)\ \boldsymbol{\beta}(t)\ dt + \sum_{j=1}^{J} \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} x_{ij}(t_1)\ x_{ij}(t_2)\ \beta_j(t_1, t_2)\ dt_1 dt_2 \\ + \sum_{j=1}^{J} \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} x_{ij}(t_1)\ x_{ij}(t_2)\ x_{ij}(t_3)\ \beta_j(t_1, t_2, t_3)\ dt_1 dt_2 dt_3 \\ + \cdots + \sum_{j=1}^{J} \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} \cdots \int_0^{\mathcal{T}} x_{ij}(t_1)\ x_{ij}(t_2)\ \cdots\ x_{ij}(t_k)\ \beta_j(t_1, t_2, \ldots, t_k) \\ dt_1 dt_2 \cdots dt_k + \epsilon_i, \quad i = 1, 2, \ldots, n,\ t, t_1, t_2, t_3 \ldots, t_k \in [0, \mathcal{T}], \end{aligned} \tag{3.4}$$

with the additional terms being the cubic part that involves the cubic polynomial of the functions of the profile factors, with 3-variate parameter functions $\beta(t_1, t_2, t_3)$, up to the k^{th} order polynomial of the functions of the profile factors, with $\beta(t_1, t_2, \ldots, t_k)$ the k^{th} order parameter function. Following the same idea, a model that depends on a k-way interaction between the $j_1, j_2, \ldots, j_k$ profile factors, is modelled as,

$$\begin{aligned} y_i = \int_0^{\mathcal{T}} \boldsymbol{x}_i^T(t)\ \boldsymbol{\beta}(t)\ dt + \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} \cdots \int_0^{\mathcal{T}} x_{ij_1}(t_1)\ x_{ij_2}(t_2)\ \cdots\ x_{ij_k}(t_k) \\ \beta(t_1, t_2, \ldots, t_k)\ dt_1 dt_2 \cdots dt_k + \epsilon_i, \quad i = 1, 2, \ldots, n,\ t, t_1, t_2, \ldots, t_k \in [0, \mathcal{T}]. \end{aligned} \tag{3.5}$$

An example of a FLM with two profile factors and their interaction, using the bivariate parameter function is discussed in Usset et al. (2016), in which they approximate the integrals numerically.

Additionally, Yao and Müller (2010) discussed a variation of the functional quadratic model (3.3), that is first proposed in the earlier work of Li and Marx (2008) for signal regression models. The variation omits some of the interaction terms by restricting the quadratic parameter function to its diagonal form. The restriction in the parameter function can be achieved through properties of the Dirac delta. The Dirac delta is defined as a function on the real line that is always zero, except when at the origin; see Balakrishnan (2003). A property of the Dirac delta is that for a general function $g(t)$,

$$\int_0^{\mathcal{T}} g(t)\ \delta(t)\ dt = g(0),$$

with $\delta(\cdot)$ the Dirac delta function; see Balakrishnan (2003). Expanding the property through shifting the Dirac delta function along the axis gives,

$$\int_0^{\mathcal{T}} g(t_1, t_2)\ \delta(t_2 - t_1)\ dt_2 = g(t_1, t_1).$$

Using the property of the Dirac delta discussed above, the double integral depending on the quadratic parameter function is restricted to,

$$\begin{aligned}\int_0^{\mathcal{T}}\int_0^{\mathcal{T}} x_1(t_1)\ x_2(t_2)\ \beta(t_1,t_2)\ \delta(t_2-t_1)\ dt_1dt_2 &= \int_0^{\mathcal{T}} x_1(t_1)x_2(t_1)\beta(t_1,t_1)\ dt_1\\ &= \int_0^{\mathcal{T}} x_1(t_1)x_2(t_1)\beta(t_1)\ dt_1,\end{aligned}$$

for $\beta(t_1) = \beta(t_1, t_1)$, as the second argument becomes redundant. Thus, the variation of the functional quadratic model is modelled through integrating the quadratic term on a single time indexing,

$$\begin{aligned} y_i = \int_0^{\mathcal{T}} \boldsymbol{x}_i^T(t)\ \boldsymbol{\beta}(t)\ dt + \sum_{j=1}^{J}\int_0^{\mathcal{T}} x_{ij}(t)\ x_{ij}(t)\ \beta_j(t)\ dt + \epsilon_i,\\ i = 1,2,\ldots,n,\ t \in [0,\mathcal{T}]. \end{aligned} \tag{3.6}$$

The functional quadratic model in (3.3), and the variation in (3.6), are reviewed in Morris (2015), in which they mention that the variation allows the diagonal cross product in the quadratic term. In this thesis, the form of the primary model considered is similar to the single index quadratic model in (3.6); expanded in the next section to allow any polynomial or interaction through a functional of the functions of the profile factors. The decision to focus on the single integral case first, is for simplicity. However, 2^{nd} (and higher) order FLMs are preliminary investigated later in the thesis, in Section 9.1.

3.2 FLM development

In this section, the aim is to introduce and describe the FLM of interest. The form of the FLM tackled, is an extension to the model in (3.6), in which the structure of the model, including polynomials and interactions, is specified through a functional of the functions of the profile factors,

$$y_i = \int_0^{\mathcal{T}} \boldsymbol{f}^T(\boldsymbol{x}_i(t))\ \boldsymbol{\beta}(t)\ dt + \epsilon_i, \quad i = 1,2,\ldots,n. \tag{3.7}$$

The functional form of the functions of the profile factors $\boldsymbol{f}^T(\boldsymbol{x}_i(t))$ is a function that has to be specified. The specification of $\boldsymbol{f}^T(\boldsymbol{x}_i(t))$ forms the FLM equation with the functions of interest; including the main effects, as well as interactions and polynomials of the functions of the profile factors. The vector of functions of the profile factors $\boldsymbol{f}^T(\boldsymbol{x}_i(t))$ is a $Q \times 1$ vector, with Q being the total number of terms in the model, and it is defined as,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \Big(f_1(\boldsymbol{x}_i(t)) \quad f_2(\boldsymbol{x}_i(t)) \quad \cdots \quad f_Q(\boldsymbol{x}_i(t))\Big), \quad i = 1,2,\ldots,n. \tag{3.8}$$

For example, assuming that the model of interest is a FLM with the main effects of $J = 2$ profile factors, then,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \begin{pmatrix} x_{i1}(t) & x_{i2}(t) \end{pmatrix}, \quad i = 1, 2, \ldots, n. \tag{3.9}$$

For the rare case with only main effects, the number of terms in the model is equal to the number of profile factors J, i.e., $Q = J = 2$. However, if an intercept is included, which it is usually included, or if interactions and polynomial effects are considered, then the number of terms in the model is not equal to the total number of profile factors, i.e., $Q \neq J$.

An intercept is incorporated in the FLM through the function $\boldsymbol{f}^T(\boldsymbol{x}_i(t))$. If the intercept is included in the model, the first component of the vector of the function $\boldsymbol{f}^T(\boldsymbol{x}_i(t))$ is 1, i.e., $f_1(\boldsymbol{x}_i(t)) = 1$. For example, when considering the FLM with the main effects of $J = 2$ profile factors from (3.9), but now including the intercept, the function $\boldsymbol{f}^T(\boldsymbol{x}_i(t))$ is defined as,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \begin{pmatrix} 1 & x_{i1}(t) & x_{i2}(t) \end{pmatrix}, \quad i = 1, 2, \ldots, n.$$

Assuming that additional to the main effects the interaction between the two profile factors is also of interest, the functional of the functions of the profile factors is defined as,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \begin{pmatrix} 1 & x_{i1}(t) & x_{i2}(t) & x_{i1}(t)x_{i2}(t) \end{pmatrix}, \quad i = 1, 2, \ldots, n,$$

and the number of terms is $Q = 4$ compared to the $J = 2$ profile factors. If the model gets more complicated and the quadratic polynomial of the first profile factor $x_{i1}(t)$ is also of interest, then another term is added to $\boldsymbol{f}^T(\boldsymbol{x}_i(t))$ and it becomes,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \begin{pmatrix} 1 & x_{i1}(t) & x_{i2}(t) & x_{i1}(t)x_{i2}(t) & x_{i1}^2(t) \end{pmatrix}, \quad i = 1, 2, \ldots, n,$$

meaning, $Q = 5$ while $J = 2$.

The functional parameters are a $Q \times 1$ vector $\boldsymbol{\beta}(t)$, representing the unknown parameter of every term in the model. Additionally, the parameter corresponding to the intercept is a scalar parameter. For instance, in the scenarios considered beforehand, the vector of functional parameters is defined as,

$$\boldsymbol{\beta}^T(t) = \begin{pmatrix} \beta_1(t) & \beta_2(t) \end{pmatrix}$$
$$\boldsymbol{\beta}^T(t) = \begin{pmatrix} \beta_1 & \beta_2(t) & \beta_3(t) & \beta_4(t) \end{pmatrix}$$
$$\boldsymbol{\beta}^T(t) = \begin{pmatrix} \beta_1 & \beta_2(t) & \beta_3(t) & \beta_4(t) & \beta_5(t) \end{pmatrix},$$

respectively. Thus, the number of functional parameters increases as the functional form of the functions of the profile factors gets more complicated. For the FLM with the main effects of the profile factors, the number of functional parameters is equal to the number of profile factors. A special case is $\beta_q(t) = \beta_q$ when the parameter corresponds to a scalar factor.

The first problem arising under this set up is estimation of the functional parameters. The FLM in (3.7) can be viewed as essentially having to estimate an infinite number of unknown parameters. This is because functions in general, and subsequently the functional parameters, are infinite dimensional objects. However, the experiment only returns a finite number of observed responses. As a result, the system is underdetermined; it consists of fewer equations than unknown parameters. Thus, there exists an infinite number of solutions for the unknown parameters that fit perfectly to the observed responses. To overcome this problem, the function space of the functional parameters is restricted via basis function expansions.

The use of basis function expansions is a common dimension reduction technique, under which, functions can be represented as linear combinations of basis functions. A basis expansion of a function, say $\boldsymbol{x}(t)$, is given by Ramsay and Silverman (2005, p. 44),

$$\boldsymbol{x}(t) = \sum_{\kappa=1}^{N} w_\kappa \phi_\kappa(t) = \boldsymbol{w}^T \boldsymbol{\phi}(t), \tag{3.10}$$

with $\boldsymbol{w}$ a vector of coefficients, $\boldsymbol{\phi}(t)$ a system of independent and known basis functions, and N the total number of basis functions. The total number of basis functions is discussed in Chapter 4. Several basis systems exist, including Fourier basis, polynomial splines and wavelets; see Ramsay and Silverman (2005, Chapter 3). One choice can be a collection of power series, $1, t, t^2, \ldots,$ i.e., linear basis, quadratic basis and so on. For instance, a quadratic function can be represented by $N = 3$ basis functions as,

$$\boldsymbol{x}(t) = \sum_{\kappa=1}^{3} w_\kappa t^{\kappa-1} = w_1 + w_2 t + w_3 t^2.$$

To this extent, the basis function expansions of the functional parameters in the FLM, are defined as,

$$\beta_q(t) = \sum_{l=1}^{n_{\beta,q}} \theta_{ql} b_{ql}(t) = \boldsymbol{b}_q^T(t)\boldsymbol{\theta}_q, \quad q = 1, 2, \ldots, Q. \tag{3.11}$$

The functions $\boldsymbol{b}_q^T(t) = [b_{q1}(t), b_{q2}(t), \ldots, b_{qn_{\beta,q}}(t)]$ are known basis functions and $\boldsymbol{\theta}_q^T = (\theta_{q1}, \theta_{q2}, \ldots, \theta_{qn_{\beta,q}})$ is a vector of unknown coefficients. The special case of a scalar parameter β_q, or the scalar parameter of the intercept, is represented through a single basis function $n_{\beta,q} = 1$ which is constantly equal to one, i.e., $b_{q1}(t) = 1$. As a result, the problem of estimating the unknown functions of the parameters has been reduced to the problem of estimating $\sum_{q=1}^{Q} n_{\beta,q}$ coefficients.

The second problem arising is the design of experiments problem to appropriately choose the functions $\boldsymbol{x}_i(t)$, $i = 1, 2, \ldots, n$. Choosing the best functions is essential to achieve optimal conditions for the experiment. The function space for each profile factor, may be very general, or may be restricted to particular classes of functions, including polynomials to a certain degree or step functions with particular break points, based on the operation of the experiments. Inference from a given experiment requires some restrictions on the function space of the profile factors, also achieved via basis function expansions,

$$x_{ij}(t) = \sum_{l=1}^{n_{x,j}} \gamma_{ijl} c_{jl}(t), \quad i = 1, 2, \ldots, n, j = 1, 2, \ldots, J. \tag{3.12}$$

The basis expansion for each profile factor can be written in vector form as, $\boldsymbol{x}_{\cdot j}(t) = \boldsymbol{\Gamma}_j \boldsymbol{c}_j(t)$, with $\boldsymbol{x}_{\cdot j}$ the function of the j^{th} profile factor in every run of the experiment, known basis functions $\boldsymbol{c}_j^T(t) = [c_{j1}(t), \ldots, c_{jn_{x,j}}(t)]$, and $\boldsymbol{\Gamma}_j$ a $n \times n_{x,j}$ coefficient matrix with γ_{ijl} the il^{th} entry, $i = 1, 2, \ldots, n, j = 1, 2, \ldots, J, l = 1, 2, \ldots, n_{x,j}$. In addition, the basis expansion can handle the special case of scalar factors. A scalar factor x_{ij} is represented through a single basis function $n_{x,j} = 1$ which is constantly equal to one, i.e., $c_{j1}(t) = 1$. After that, $x_{ij} = \gamma_{ij1}$ is a single value that needs to be specified in every run of the experiment. The methodology developed later in the thesis is based on the use of polynomial splines and specifically B-spline basis functions. Polynomial splines and some basis systems, including the B-spline basis, are described in Chapter 4.

In the next section, the related literature on experimentation with models that depend on profile factors is discussed. A new methodology that takes into consideration the challenges involved with the FLM and allows to identify optimal experimental designs is described in Chapter 5. Closed form expressions for integrals of the form in (3.7) and specific basis expansions, are derived later in the thesis.

3.3 Previous work on experimentation with profile factors

Design of experiments and functional data analysis are well established topics in the Statistics literature. However, design of experiments for models involving profile factors and functional parameters has received much less attention, with two main approaches being proposed. The two approaches are: response surface methodology using dimension reduction techniques; and optimal design for dynamic models, typically derived from differential equations. This section serves to introduce to the reader any previous work on experimentation involving profile factors. The approach to be developed in later chapters is related to the response surface approach to find optimal functions for profile factors.

3.3.1 Response Surface Methodology

The first approach is an adaptation of the Response Surface Methodology (RSM) using dimension-reduction techniques; see Georgakis (2013), Roche (2015), and Roche (2018). RSM is a strategy of experimentation and optimisation, introduced by Box and Wilson (1951). Depending on the objective, the aim is usually to find the settings of factors where the expected minimum or maximum response occurs. RSM has broadly three phases: phase 1 requires experimentation to examine whether the current operating conditions are near optimal, achieved by fitting a first-order linear model; phase 2 identifies the direction towards an optimum region, by fitting the first-order model and estimating the gradient of the fitted surface; and Phase 3 requires fitting a second-order model to take into account the curvature of the surface. A detailed description of RSM is given by Wu and Hamada (2011, Chapter 10).

Georgakis (2013) was the first to notice the limited research for optimal experimental designs that depend on profile factors. Their motivation is to provide a data-driven approach, that optimise batch processes for models that involve profile factors, referred to as time-varying decision variables, or dynamic factors, in the article. The author discusses the excess use of knowledge-based models in chemical and petroleum processes, over the past decades. They go on to emphasize that in other processes, for instance batch processes, where large volume of information is not available, the development of knowledge-based models is not justified. In contrast, they indicate that for these processes, the use of data-driven models is preferred. In context, Georgakis (2013) developed a data-driven approach, which is an extension of the response surface methodology, with typically a second-order model, to allow experimentation with a combination of profile and static factors. Their model of interest is very similar to the FLM introduced in the previous section of the thesis. Functions of space or time, are assumed to belong to the Hilbert space of squared integrable functions; see Young (1988) for an introduction to Hilbert spaces. To address the problem of infinite dimensional objects, they introduce the use of basis functions. In fact, they use polynomial basis and they assume the same basis expansion for the profile factors and the functional parameters. Additionally, they indicate that their choice of basis is orthogonal and normalised, so that calculations are simpler. However, they mention that a basis that is not orthogonal still works, but it can be a more complex process. To achieve specific bounds for the functions of the profile factors, they introduce constraints on the coefficients of the basis expansion. With a view to evaluate the performance of their approach, they conduct examples, for which very accurate knowledge-based models are available. They design the experiment using central composite designs. It is worth noting that the identified results of the two approaches were very close. In a more recent work, Klebanov and Georgakis (2016) expanded the data-driven approach to models with dynamic responses.

Roche (2015) and Roche (2018) discussed applications for which an optimisation of a function is required. They continue to say that typically, the space of functions belongs to a family of functions, i.e., to a subset of the general space of functions. In such situations, they mention that functions can be approximated locally by a polynomial regression model of order 1 or 2, in the context of RSM. On another note, they indicate that often, the function space may be very general, specifically when functions are dependent on space, or time, i.e., infinite dimensional objects. The model they used to specify the relationship between a scalar response and profile factors, is the FLM. In context, they proposed an adaptation of RSM using dimension reduction techniques, assuming that the functions belong to a general Hilbert space of squared integrable functions. The choice of basis for the profile factors in their experiments is the Fourier basis, and they indicate that the basis is orthonormal. After that, they discuss a data-driven direction of optimisation, under the assumption that some data are available prior to the experiment. They use principal component analysis (PCA), which targets patterns and locates features in the data; see Hall (2011), and partial leasts squares (PLS) regression, that is similar to PCA, but involves information from both the explanatory and the response factors.

3.3.2 Dynamic models

A related design problem to functional models, described in the previous sections, is optimal designs for dynamic models, typically derived from differential equations. A dynamic model mainly deals with time dependent profiles and is formulated as a general dynamic optimisation problem with the aim of minimising or maximising an objective function. Dynamic models are often used to study reaction rates; and are extensively used in the field of chemical kinetics and food engineering.

At first, the work by Titterington (1980) focused on a single input, single output, dynamic system, with a view to expand static optimal design strategies to systems for which time is an inevitable component. However, the system can be seen as time invariant, since the parameters do not vary with time. Experiments with single profile factors, like temperature and pressure in chemical reactions have been tackled by Uciński and Bogacka (2005) and Uciński and Bogacka (2007), to find the levels of the factor to take observations and the proportions at each level, to estimate static parameters. The single profile factor is expanded as in the equation of basis expansions in (3.10). After that, the design becomes the choice of coefficients from the basis expansion and the initial conditions of the response factors. For instance, the design is given by,

$$\boldsymbol{X} = (\boldsymbol{w}, \boldsymbol{y}_0), \tag{3.13}$$

for $\boldsymbol{y}_0$ being a vector of the initial conditions and $\boldsymbol{w}$ the vector of coefficients from the basis expansion of the profile factor. The coefficients are chosen in a way that certain constraints on the bounds of the profile factor hold. The known function from the expansion form a basis for the profile factor. The main interest of this work was not to find optimal functions for the profile factors, but to discriminate between competing models.

The work by Asprey and Macchietto (2002) considers dynamic experiments involving profile factors for obtaining robust optimal designs that improve the parameter precision. The profile factors are assumed to be piecewise constant, piecewise linear or piecewise quadratic functions of time, defined over a number of time intervals. For example, if the function of the profile factor is piecewise constant, the profile factor is controlled through the coefficient of each time interval. Consequently, the design becomes the initial conditions $\boldsymbol{y}_0$ and the coefficients on every time interval defined by the choice of knots; as in (3.13).

A mathematical formulation for optimal experimental designs for dynamic models was also considered by Balsa-Canto and Banga (2007). The optimal designs tackled are for single-variable dynamic models with static parameters, from ordinary differential and partial differential equation systems. The design problem is to find the time processing profile factor, the initial conditions, the sampling times and the experiment durations; to estimate kinetic parameters. The high-dimensionality problem arising from the time-varying profile factor was tackled using control vector parametrisation and non-linear optimisation solver; see Balsa-Canto and Banga (2007) and references therein. The control vector parametrisation approach was used to transform the infinite dimension problem into a non linear programming problem. The duration of the experiment was divided in multiple intervals and the profile factors involved were expressed as low order polynomials. After that, non linear solvers were used to identify optimal conditions for the coefficients from the expansion of the profile factors, the initial conditions, and the sampling times. After that, the design problem gets similar to the aforementioned cases.

3.4 Conclusion

In this chapter, the FLM has been introduced, to model the relationship between scalar responses and functions of profile factors and functional parameters. After bringing out the challenges faced under functional linear models, the functions of the profile factors and the functional parameters have been represented as linear combinations of basis functions, to restrict the function space. After that, this chapter has reviewed articles that discuss experimentation for models with profile factors. There is not extensive literature on experimental designs for models that involve profile factors. Specifically, there are some articles discussing an extension of the RSM using basis expansions; and experimental designs for dynamic models. The methodology that is developed later in the thesis is similar to the RSM approach. Dynamic models are a slightly different, yet related design problem, to the design problem discussed in this thesis. Dynamic models are typically derived from a system of differential equations. To construct the system of equations, information and knowledge is needed. For experiments that knowledge to construct a system of equations is not available, the approach developed in this thesis is preferred.

Chapter 4

Polynomial splines

Profile factors are factors whose settings vary as functions of time. Functions in general are infinite dimensional objects. Basis expansions are always able to represent functions as linear combinations of basis functions within a given function space; see Ramsay and Silverman (2005, Chapter 3). The choice of basis functions ideally should match features of the functions of the profile factors. This is to achieve better approximations and computational efficiency.

In general, functions are divided into the periodic and the non-periodic functions. A common basis system for periodic functions is a Fourier basis system; see Ramsay and Silverman (2005, p. 45). For the purpose of this work, the main focus is non-periodic functions that does not recur at regular intervals, to account for low order derivatives. Non-periodic functions can be modelled by polynomial splines, which are functions defined piecewise by polynomials. Splines are reviewed in Section 4.1. Basis systems including the step function basis, the truncated power series basis, and the B-spline basis are described in Sections 4.2, 4.3 and 4.4, respectively. The preference on the use of B-splines is also discussed.

4.1 Introduction to splines

Consider a closed interval $[a, b]$ and real numbers lying in this interval such that,

$$a = \lambda_0 < \lambda_1 < \lambda_2 < \cdots < \lambda_K < \lambda_{K+1} = b. \tag{4.1}$$

A function $x(t)$ on the closed interval $[a, b]$ is said to be a spline of degree d if two conditions hold; see Wood (2017*a*, p. 196). The two conditions are listed below:

(i) The function $x(t)$ has to be a polynomial of degree d on each and every interval $[\lambda_k, \lambda_{k+1}]$, for $k = 0, 1, \ldots, K$.

(ii) Continuity, i.e., the function $x(t)$ has to be at least continuous at the inner points $\lambda_1, \lambda_2, \ldots, \lambda_K$ with up to $(d-1)$ derivatives being continuous at each of the inner points.

The whole interval is divided into subintervals using the breakpoints $\lambda_1, \lambda_2, \ldots, \lambda_K$, called knots. Each subinterval is allowed a different polynomial. For example, suppose a choice of $K = 3$, then there exist three knots $\lambda_1, \lambda_2, \lambda_3$, and the whole interval is divided into four subintervals with each of the four subintervals allowed a different polynomial.

The smoothness of each knot, denoted as s_h, for $h = 1, 2, \ldots, K$, is such that $0 \leq s_h \leq d-1$, with d being the degree of the polynomial spline basis. If $s_h = 0$, then the function is unconstrained, meaning that the function is continuous at the knot but none of its derivatives are continuous. If $s_h \geq 1$, then all the function derivatives at that knot are continuous up to and including the s_h^{th} derivative. A special case where the spline is said to be maximally smooth, is the case of $s_h = d - 1$. The special case of maximally smooth splines is commonly used, and the examples in this thesis assume maximally smooth spline basis functions as well.

The total number of basis functions N, which are the elements of a basis system, depend on: the degree of the spline, the choice of knots, and the degree of continuity of the knots,

$$N = (K+1)d + 1 - \sum_{h=1}^{K} s_h, \tag{4.2}$$

given in Ramsay and Silverman (2005, Chapter 3). From the last term in equation (4.2), it is important to notice that as the smoothness of the knots increases, the total number of basis functions drops by an equivalent amount. This happens because, as the smoothness increases, the continuity of the derivatives is satisfied up to a higher degree. Thus, extra constraints are added to the problem causing the number of functions to drop; see Woods et al. (2003). For a maximally smooth basis the total number of basis functions is given by,

$$N = K + d + 1, \tag{4.3}$$

derived using a combination of (4.2) and the definition of a maximally smooth basis; see Grove et al. (2004) and Perperoglou et al. (2019).

Finally, a spline function of degree d is expressed as a linear combination of basis functions $\phi_{\kappa,d}(t), \kappa = 1, 2, \ldots, N$, along with constants $\boldsymbol{w} = (w_1, w_2, \ldots, w_N)$ called control points such that,

$$x(t) = \sum_{\kappa=1}^{N} w_\kappa \, \phi_{\kappa,d}(t), \tag{4.4}$$

for d representing the required degree.

To assist the understanding of polynomial splines, consider an example for the sine function. The aim is to use spline functions of piecewise linear and cubic degree to fit the sine function $\sin(t)$ over the time interval $t \in [0, 2\pi]$. Three equally spaced interior knots are used, thus, the time interval is divided in four equally spaced subintervals, and each subinterval is allowed a different polynomial. As the complexity of the spline increases, i.e., from linear spline to cubic spline, the smoothness increases, i.e., more continuous derivatives at the knots, ensuring a smooth estimate of the function. Thus, the cubic spline approximates the sine function better, compared to the linear spline; see Figure 4.1.

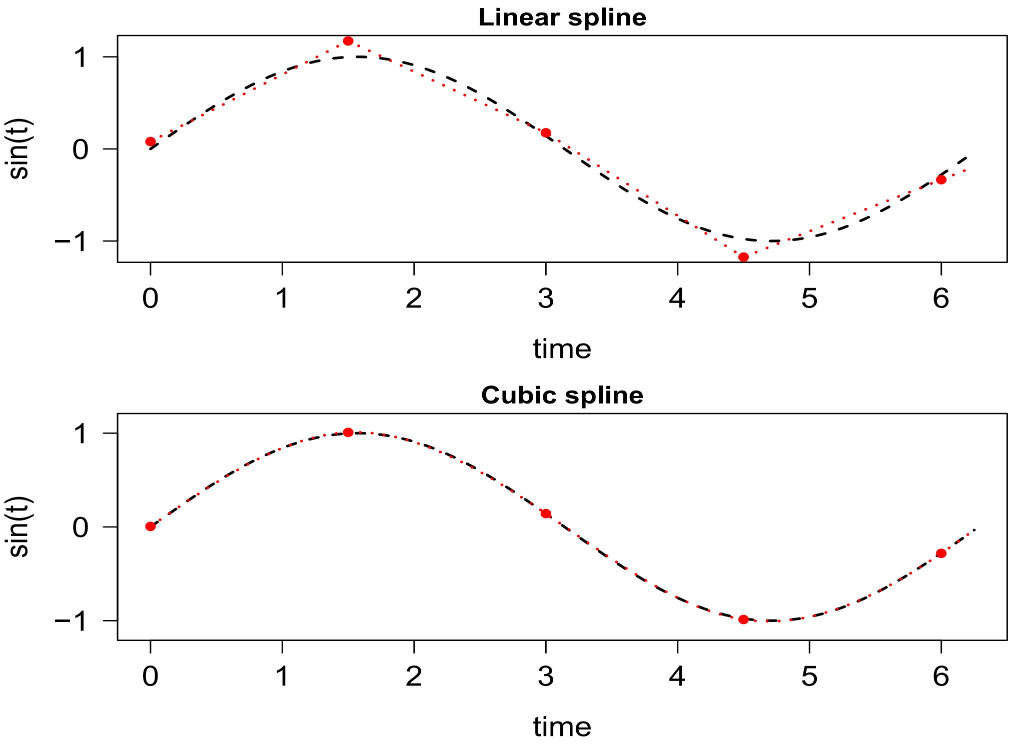


FIGURE 4.1: The dashed black lines represents the sine function in the time interval $t \in [0, 2\pi]$ and the red dashed lines represent the piecewise linear and cubic splines used to fit the sine function. The linear spline is on the top panel and the cubic spline is on the bottom panel of the figure.

4.2 Step function basis

The simplest spline is a spline of degree zero, i.e., $d = 0$. The degree zero spline is also known as a step function, and it is an appropriate basis to use if it is assumed that control of the functions is represented via step functions. A step function can be expressed as a linear combination of interval functions. Formally, a function $x(t)$ defined on the real line, is a step function if it is expressed in the form,

$$x(t) = \sum_{\kappa=1}^{N} w_\kappa 1_\kappa(t), \quad t \in [a, b], \tag{4.5}$$

where w_κ are constants and 1_κ is an indicator function of the knot intervals. A step function basis is a spline of degree $d = 0$, thus, the total number of basis functions is $N = K + 1$, using (4.3). To derive the basis functions, an extended knot vector which is a combination of the boundary and interior knots is required,

$$\hat{\boldsymbol{\lambda}} = (a,\ \lambda_1, \ldots, \lambda_K,\ b)^T. \tag{4.6}$$

and then, the indicator function is defined as,

$$1_k(t) = \begin{cases} 1 & \text{if } \hat{\lambda}_\kappa \le t < \hat{\lambda}_{\kappa+1} \\ 0 & \text{otherwise.} \end{cases} \tag{4.7}$$

For example, if the function $x(t)$ is represented by a step function of three basis functions, there are two interior knots and three intervals; see Figure 4.2.

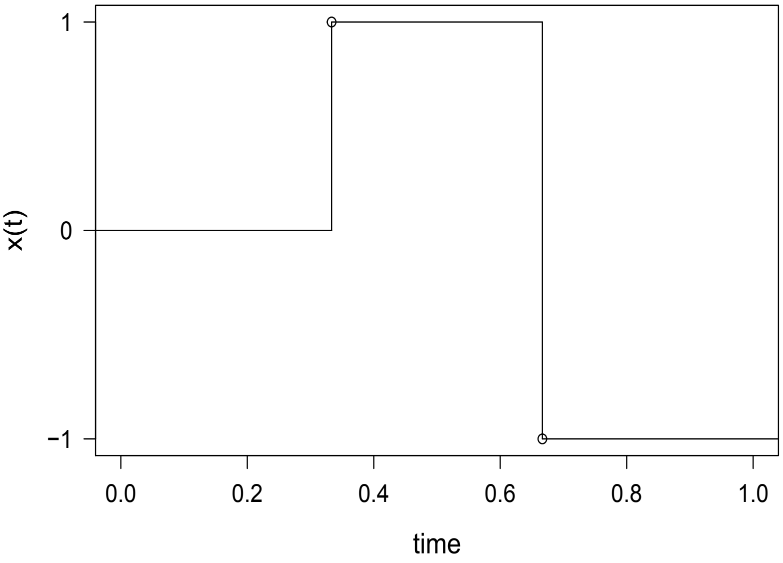


FIGURE 4.2: Basis functions example for $x(t)$ with 3 basis functions with constants $\boldsymbol{w} = (0, 1, -1)$ and two interior knots $\boldsymbol{\lambda} = (0.33, 0.66)$.

4.3 Truncated power series basis

Truncated power series (TPS) basis is a simple and easily interpretable basis system. A complete set of N basis functions for degree d and knots $\lambda_1, \lambda_2, \dots, \lambda_K$ is given by,

$$\begin{aligned} \Upsilon_d(t) = \{t^j\}_{j=0}^d, \{(t-\lambda_1)_+^j\}_{j=s_1+1}^d, \{(t-\lambda_2)_+^j\}_{j=s_2+1}^d, \dots, \{(t-\lambda_K)_+^j\}_{j=s_K+1}^d, \\ t \in [a,b], \end{aligned} \tag{4.8a}$$

$$(t-\lambda_h)_+^j = \begin{cases} (t-\lambda_h)^j & \text{if } t \geq \lambda_h \\ 0 & \text{if } t < \lambda_h, \end{cases} \quad h = 1, 2, \dots, K. \tag{4.8b}$$

For the special case of a maximally smooth spline, where $s_h = (d-1), h = 1, 2, \dots, K$, a set of N truncated power series basis functions is given by,

$$\Upsilon_d(t) = \{1, t, t^2, \dots, t^d, (t-\lambda_1)_+^d, (t-\lambda_2)_+^d, \dots, (t-\lambda_K)_+^d\}. \tag{4.9}$$

Hence, a TPS basis consists of two subsets of functions, which are: the monomials up to degree d, and the truncated power functions (TPF). The TPF part depends on the number of knots and forms K basis functions; see (4.8b).

For example, for choice of degree $d = 1$ and a single knot $\lambda_1 = 0.5$, a linear truncated power series basis of $N = 3$ basis functions takes the form,

$$\{1, t, (t-0.5)_+\}. \tag{4.10}$$

From (4.10), the monomials part is $(1, t)$ and the TPF part is $(t-0.5)_+$. The basis from (4.10) is demonstrated in Figure 4.3(a). Similarly, for choice of degree $d = 3$ and a knot vector $\boldsymbol{\lambda} = (0.33, 0.66)$, a cubic truncated power series basis of $N = 6$ basis functions takes the form,

$$\{1, t, t^2, t^3, (t-0.33)_+^3, (t-0.66)_+^3\}. \tag{4.11}$$

From (4.11), the monomials part is $(1, t, t^2, t^3)$ and the TPF part is $(t-0.33)_+^3, (t-0.66)_+^3$. The basis from (4.11) is demonstrated in Figure 4.3(b).

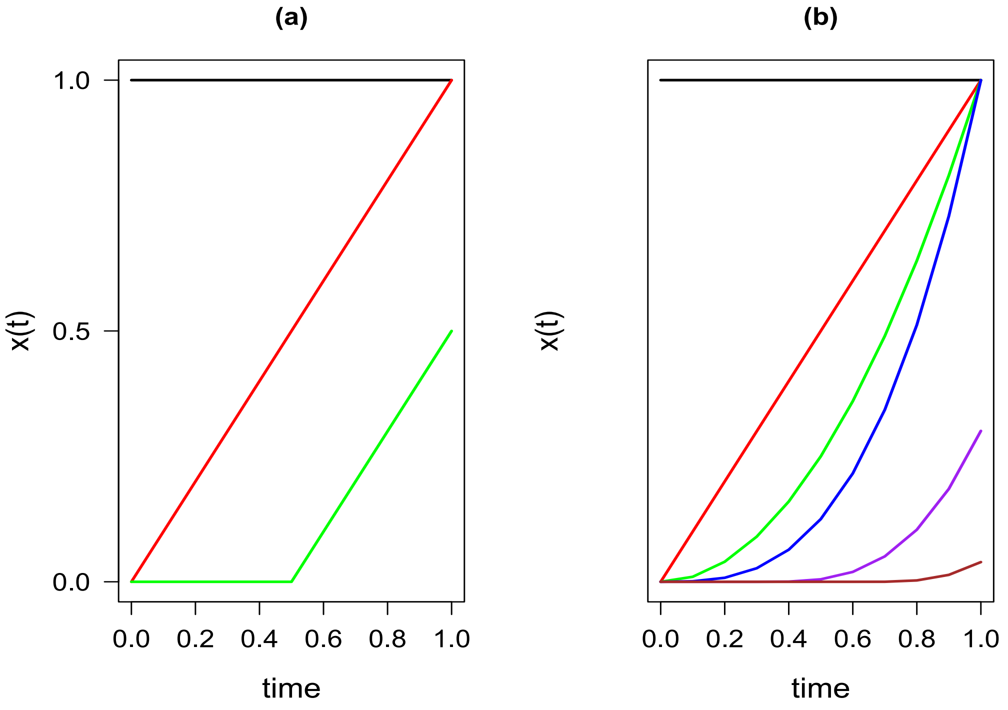


FIGURE 4.3: (a) Linear with knot at 0.5 and (b) Cubic with knots at 0.33 and 0.66 TPS basis functions.

A set of basis functions from a TPS basis is an attractive basis system, since it allows a straightforward interpretation of each basis function. Moreover, an addition of extra knots does not change the existing basis functions. However, if knots are very close to each other or very close to the boundaries, the basis functions may be close to linearly dependent. Linear dependencies increase the possibility of ill-conditioning, causing problems finding designs due to difficulties inverting the information matrix.

4.4 B-splines basis

B-spline (BS) basis is another univariate basis which is less interpretable than TPS basis, but more computationally stable. It gives a numerically stable scheme to evaluate a spline function by the sum of a set of BS basis functions and vector valued constants, as in equation (4.4). A set of BS basis functions, denoted as $B_{\kappa,d}$, $\kappa = 1, 2, \ldots, N$, forms a spline basis of degree d with K knots on an interval $[a, b]$.

The BS basis functions are computed by the use of a straightforward recursion relation formula, known as the Cox-de Boor recursion formula; see (De Boor, 1978, p. 90). The idea of the recursion formula is an expansion of the formal definition of a normalised

BS which depends on divided differences of TPFs; see De Boor (1978, p. 87) and Eilers and Marx (2010). Thus, the BS and TPS basis systems belong to the same family of basis, and for every TPS basis there exists an equivalent BS basis in the same space of functions; see Friedman et al. (2001, Chapter 6) and Appendix A for more details. In general, the use of the Cox-de Boor formula provides a more practical method for generating the basis functions. For that reason, it is usually preferred over the formal definition using divided differences.

The BS recursion formula requires the use of an extended knot vector. This is because every knot subinterval needs to have at least d knots before and after the span. This is achieved by repeating the boundary knots $(d+1)$ times and the interior knots by $(d-s_h), h = 1, 2, \ldots, K$ times. The extended knot vector denoted as $\boldsymbol{\lambda}^*$ and it is given by,

$$\boldsymbol{\lambda}^* = \big(\lambda_1^*, \lambda_2^*, \ldots, \lambda^*_{2(d+1)+Kd-\sum_{h=1}^K s_h}\big),$$

and its expansion is such that,

$$\Big(\underbrace{\lambda_0, \ldots, \lambda_0,}_{d+1} \quad \underbrace{\lambda_1, \ldots, \lambda_1,}_{d-s_1} \quad \underbrace{\lambda_2, \ldots, \lambda_2,}_{d-s_2} \quad \ldots, \quad \underbrace{\lambda_K, \ldots, \lambda_K,}_{d-s_K} \quad \underbrace{\lambda_{K+1}, \ldots, \lambda_{K+1}}_{d+1}\Big).$$

Under the assumption of a maximally smooth B-spline, the extended vector is given by,

$$\boldsymbol{\lambda}^* = \big(\lambda_1^*, \lambda_2^*, \ldots, \lambda^*_{2(d+1)+K}\big)$$

and its expansion is such that,

$$\Big(\underbrace{\lambda_0, \ldots, \lambda_0,}_{d+1} \quad \lambda_1, \quad \lambda_2, \quad \ldots, \quad \lambda_K, \quad \underbrace{\lambda_{K+1}, ..., \lambda_{K+1}}_{d+1}\Big).$$

By definition in (De Boor, 1978, p. 89), the degree $d = 0$ B-spline is defined as,

$$B_{\kappa,0}(t) = \begin{cases} 1 & \text{if } \lambda_\kappa^* \leq t < \lambda_{\kappa+1}^* \\ 0 & \text{otherwise.} \end{cases} \tag{4.12}$$

Higher degree B-splines are computed with the aid of the recursion formula given by,

$$B_{\kappa,p}(t) = \frac{t - \lambda_\kappa^*}{\lambda_{\kappa+p}^* - \lambda_\kappa^*} B_{\kappa,p-1}(t) + \frac{\lambda_{\kappa+p+1}^* - t}{\lambda_{\kappa+p+1}^* - \lambda_{\kappa+1}^*} B_{\kappa+1,p-1}(t),$$
$$t \in [a,b], \ \kappa = 1, 2, \ldots, N, \ p = 1, 2, \ldots, d, \tag{4.13}$$

where $\boldsymbol{\lambda}^*$ is the extended knot vector.

A special case of B-splines is that a BS basis of degree $d = 0$ is equivalent to a step function basis; see equations (4.7) and (4.12). For a BS basis of degree $d = 0$, the extended knot vector is the interior knots plus the boundary knots, exactly as for the step function basis. Also, the number of basis functions for the degree zero BS basis is $K + d + 1 = K + 1$, again exactly as for the step function basis. This is also true for the TPS of degree 0.

For example, consider a BS basis of degree $d = 0$ and 2 equally spaced interior knots for $t \in [0, 1]$. Suppose also, that the coefficients are $\boldsymbol{w} = (0, 1, -1)$. The extended knot vector consists of the boundary knots repeated $d + 1 = 1$ times, and the interior knots, i.e., $\boldsymbol{\lambda}^* = (0, \lambda_1, \lambda_2, 1) = (0, 0.33, 0.66, 1)$. The total number of basis functions is 3, derived from the knot spans $[0, 0.33), [0.33, 0.66)$, and $[0.66, 1)$. Thus, the basis functions are: $B_{1,0}(t) = 1$ on $[0, 0.33)$ and 0 elsewhere, $B_{2,0}(t) = 1$ on $[0.33, 0.66)$ and 0 elsewhere, and $B_{3,0}(t) = 1$ on $[0.66, 1)$ and 0 elsewhere. This is equivalent to expanding the indicator function in (4.7). The function is calculated using (3.10), and it is such that:

$$x(t) = \begin{pmatrix} 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & -1 \end{pmatrix},$$

which is an identical function to the step function in Figure 4.2. Hence, in the examples presented later in the thesis, if the control of a function is represented via a BS basis of degree zero, it is equivalently represented via a step function basis.

In the framework of spline basis systems, B-splines are less interpretable to truncated power basis, but more computationally efficient. Several other properties of B-splines are advantageous to their user when building statistical models; see Grove et al. (2004) and Pan and Saleh (2019). A useful property for the scope of this work is that the sum of the basis functions at every value of time add up to 1 such that,

$$\sum_{\kappa=1}^{N} B_{\kappa,d}(t) = 1, \quad \text{for } t \in [a, b]. \tag{4.14}$$

This property is an efficient tool for constrained bounds for the functions of the profile factors. The reason is that any bound constraints on the design are equivalent constraints on the functions of the profile factors.

Following the advice from De Boor (1978, Chapter 9), BS basis in this thesis are computed using the recursion formula instead of the differences formula. Finding BS basis using the recursion formula is less complicated and avoids any numerical instabilities. As discussed in Sections 4.3 and 4.4, the TPS basis system is more interpretable, but the

BS basis system is more computationally efficient. For the computationally efficiency of the BS basis system and some of its properties allowing constrained optimisation of the functions of the profile factors, the BS basis system is preferred in the thesis.

Finally, remember that a condition for a function to be a spline of degree d is to be up to $(d-1)$ derivatives continuous at each of the interior knots. The derivative of a BS basis function of degree d is also a BS basis function of degree $(d-1)$ with exactly the same knots in the interval $[a,b]$. Let $\tilde{B}_{\kappa,d-1}(t)$ be a B-spline of one degree lower than $B_{\kappa,d}(t), \kappa = 1,2,\ldots,N$, but with the same extended knot vector. The first derivative of $B_{\kappa,d}(t)$ is defined as,

$$D^{(1)}\left[B_{\kappa,d}(t)\right] = d\left[\frac{\tilde{B}_{\kappa,d-1}(t)}{\lambda^*_{\kappa+d}-\lambda^*_{\kappa}} - \frac{\tilde{B}_{\kappa+1,d-1}(t)}{\lambda^*_{\kappa+d+1}-\lambda^*_{\kappa+1}}\right]. \tag{4.15}$$

A recurring application of (4.15) provides the computation of higher derivatives of a B-spline basis. For instance, the m^{th} derivative of a BS basis is given in Butterfield (1976),

$$D^{(m)}\left[B_{\kappa,d}(t)\right] = d\left[\frac{D^{(m-1)}[\tilde{B}_{\kappa,d-1}(t)]}{\lambda^*_{\kappa+d}-\lambda^*_{\kappa}} - \frac{D^{(m-1)}[\tilde{B}_{\kappa+1,d-1}(t)]}{\lambda^*_{\kappa+d+1}-\lambda^*_{\kappa+1}}\right], \tag{4.16}$$

with $D^{(m)}$ the m^{th} derivative.

4.5 Conclusion

This chapter has reviewed definitions and described polynomial splines and basis systems as a method of restricting the function space and achieve dimension reduction. Detailed descriptions have been given for the truncated power series basis and the B-spline basis. Advantages, disadvantages and the connection between the two systems has been explored. The step function basis, which is a special case of the truncated power and B-spline basis, is also defined. Finally, reasons for the preference on the use of the B-spline basis in this thesis, are stated. The B-spline basis is more suitable for the design of experiments methodology developed in this thesis for certain properties and computational efficiency, which get more coherent in the next chapters.

Chapter 5

Design of Experiments for functional linear models

In this chapter, the focus is to develop a new methodology for identifying optimal experimental designs for FLMs. The main challenges of the FLM, in estimation and design, have been discussed in Chapter 3. To overcome the challenges, basis expansions were introduced on the functional parameters and the profile factors. The new methodology is based on the use of the basis functions, that simplifies the FLM into the form of the traditional linear model in Section 5.1. This allows the use of common experimental design strategies to identify optimal designs. The special case of functional linear models with the main effects of the profile factors is discussed in Section 5.2. Functional linear models that involve interactions and polynomials of the profile factors which are rather more challenging, are discussed in Section 5.3.

A frequentist approach to fit the FLM is discussed in Section 5.4. The design problem for finding optimal functions of the profile functions is reduced to optimisation of scalar values through the methodology developed. The reduced design problem is described in Section 5.4, also. A Bayesian approach to fit the FLM through roughness penalties that penalise the complex functions, is discussed in Sections 5.5 and 5.6. Moreover, to tailor the design problem to the FLM; a Bayesian design criterion for profile factors is described in Section 5.7.

Objective functions are often functions of the Fisher information matrix, and sometimes of its inverse. To limit the likelihood of non-invertibility of the information matrix, constraints on the experimental settings are introduced in Section 5.8. The methodology developed involves integrals of products of B-spline basis functions with respect to time. An analytic evaluation of integrals involving the product of an unlimited number of B-spline basis functions is discussed in Section 5.9. Finally, a method for valuing the complexity of functions of profile factors is described in Section 5.10.

5.1 FLM methodology as an extension to the linear model

In this section, the basis expansions of the functions of the profile factors and of the functional parameters are substituted into the FLM. The aim is to present the FLM as an extension to the traditional linear model. The latter will allow common statistical methods to be applied on the FLM and support the development of an efficient approach for functional experimental designs.

Recall from Section 3.1 that the relationship between a scalar response and multiple profile factors is modelled by a FLM. At the i^{th} run of the experiment, the FLM is defined as in (3.7), for $i = 1, 2, \ldots, n$. In matrix form, the FLM is given by,

$$\boldsymbol{y} = \int_0^{\mathcal{T}} \boldsymbol{f}^T(\boldsymbol{X}(t))\, \boldsymbol{\beta}(t)\, dt + \boldsymbol{\epsilon}, \tag{5.1}$$

for $\boldsymbol{y}^T = (y_1, y_2, \ldots, y_n)$ the $n \times 1$ vector of responses, $\boldsymbol{\beta}(t)$ the vector of the functional parameters as before, and $\boldsymbol{\epsilon}^T = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$ the $n \times 1$ vector of independent error terms with mean zero and variance-covariance $\sigma^2 \boldsymbol{I}_n$.

The matrix $\boldsymbol{X}(t)$ is a matrix of dimensions $n \times J$ with the ij^{th} entry containing the function of the j^{th} profile factor at the i^{th} run of the experiment, $i = 1, 2, \ldots, n, j = 1, 2, \ldots, J$. Thus, every row of $\boldsymbol{X}(t)$ is the vector $\boldsymbol{x}_i^T(t)$. Equivalently, the j^{th} column of $\boldsymbol{X}(t)$ is the vector $\boldsymbol{x}_{\cdot j}(t), j = 1, 2, \ldots, J$, that represents the function of the j^{th} profile factor in every run of the experiment such that,

$$\boldsymbol{X}(t) = \begin{pmatrix} \boldsymbol{x}_1^T(t) \\ \boldsymbol{x}_2^T(t) \\ \vdots \\ \boldsymbol{x}_n^T(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_{\cdot 1}(t) & \boldsymbol{x}_{\cdot 2}(t) & \cdots & \boldsymbol{x}_{\cdot J}(t) \end{pmatrix} = \begin{pmatrix} x_{11}(t) & x_{12}(t) & \cdots & x_{1J}(t) \\ x_{21}(t) & x_{22}(t) & \cdots & x_{2J}(t) \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1}(t) & x_{n2}(t) & \cdots & x_{nJ}(t) \end{pmatrix}. \tag{5.2}$$

After applying the profile factor basis expansions from (3.12), the $n \times J$ matrix $\boldsymbol{X}(t)$ is defined as,

$$\begin{aligned}\boldsymbol{X}(t) &= \begin{pmatrix} \sum_{l=1}^{n_{x,1}} \gamma_{11l}c_{1l}(t) & \sum_{l=1}^{n_{x,2}} \gamma_{12l}c_{2l}(t) & \cdots & \sum_{l=1}^{n_{x,J}} \gamma_{1Jl}c_{Jl}(t) \\ \sum_{l=1}^{n_{x,1}} \gamma_{21l}c_{1l}(t) & \sum_{l=1}^{n_{x,2}} \gamma_{22l}c_{2l}(t) & \cdots & \sum_{l=1}^{n_{x,J}} \gamma_{2Jl}c_{Jl}(t) \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{l=1}^{n_{x,1}} \gamma_{n1l}c_{1l}(t) & \sum_{l=1}^{n_{x,2}} \gamma_{n2l}c_{2l}(t) & \cdots & \sum_{l=1}^{n_{x,J}} \gamma_{nJl}c_{Jl}(t) \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{\Gamma}_1\boldsymbol{c}_1(t) & \boldsymbol{\Gamma}_2\boldsymbol{c}_2(t) & \cdots & \boldsymbol{\Gamma}_J\boldsymbol{c}_J(t) \end{pmatrix},\end{aligned} \tag{5.3}$$

with $\boldsymbol{\Gamma}_j$ and $\boldsymbol{c}_j(t)$ for $j = 1, 2, \ldots, J$, as in Section 3.2.

The function $\boldsymbol{f}^T(\boldsymbol{X}(t))$ is a functional of the functions of the profile factors that needs to be specified to define the structure of the FLM as before. However, the function $\boldsymbol{f}$ in (5.1) is acting row-wise on $\boldsymbol{X}(t)$. The function $\boldsymbol{f}^T(\boldsymbol{X}(t))$ is an $n \times Q$ matrix, with the q^{th} column representing the q^{th} term in the model, $q = 1, 2, \ldots, Q$. In other words, the iq^{th} entry of $\boldsymbol{f}^T(\boldsymbol{X}(t))$ is a function of the profile factors, specified by f_q, in the i^{th} run of the experiment,

$$\begin{aligned}\boldsymbol{f}^T(\boldsymbol{X}(t)) &= \begin{pmatrix} f_1(\boldsymbol{X}(t)) & f_2(\boldsymbol{X}(t)) & \cdots & f_Q(\boldsymbol{X}(t)) \end{pmatrix} \\ &= \begin{pmatrix} f_1(\boldsymbol{x}_1^T(t)) & f_2(\boldsymbol{x}_1^T(t)) & \cdots & f_Q(\boldsymbol{x}_1^T(t)) \\ f_1(\boldsymbol{x}_2^T(t)) & f_2(\boldsymbol{x}_2^T(t)) & \cdots & f_Q(\boldsymbol{x}_2^T(t)) \\ \vdots & \vdots & \vdots & \vdots \\ f_1(\boldsymbol{x}_n^T(t)) & f_2(\boldsymbol{x}_n^T(t)) & \cdots & f_Q(\boldsymbol{x}_n^T(t)) \end{pmatrix}.\end{aligned} \tag{5.4}$$

Equivalently, the q^{th} column of $\boldsymbol{f}^T(\boldsymbol{X}(t))$ is a function of the profile factors in every run of the experiment such that,

$$\boldsymbol{f}_q(\boldsymbol{X}(t)) = \boldsymbol{f}_q \begin{pmatrix} \boldsymbol{x}_{\cdot 1}(t) & \boldsymbol{x}_{\cdot 2}(t) & \cdots & \boldsymbol{x}_{\cdot J}(t) \end{pmatrix}, \quad q = 1, 2, \ldots, Q.$$

If the q^{th} function represents the intercept, then $\boldsymbol{f}_q(\boldsymbol{X}(t)) = \mathbf{1}_n$, with $\mathbf{1}_n$ being the $n \times 1$ vector of 1's.

After applying the profile factor basis expansions from (3.12), the iq^{th} entry of (5.4) is a function of the basis expansions of the J profile factors in the i^{th} run of the experiment such that,

$$\begin{aligned}f_q(\boldsymbol{x}_i^T(t)) &= f_q \begin{pmatrix} x_{i1} & x_{i2} & \cdots & x_{iJ} \end{pmatrix} \\ &= f_q \begin{pmatrix} \sum_{l=1}^{n_{x,1}} \gamma_{i1l}c_{1l}(t) & \sum_{l=1}^{n_{x,2}} \gamma_{i2l}c_{2l}(t) & \cdots & \sum_{l=1}^{n_{x,J}} \gamma_{iJl}c_{Jl}(t) \end{pmatrix} \\ & i = 1, 2, \ldots, n, q = 1, 2, \ldots, Q,\end{aligned} \tag{5.5}$$

and equivalently, the q^{th} column of (5.4) becomes the function,

$$\boldsymbol{f}_q(\boldsymbol{X}(t)) = \boldsymbol{f}_q\left(\boldsymbol{\Gamma}_1\boldsymbol{c}_1(t) \quad \boldsymbol{\Gamma}_2\boldsymbol{c}_2(t) \quad \cdots \quad \boldsymbol{\Gamma}_J\boldsymbol{c}_J(t)\right), \quad q = 1, 2, \ldots, Q, \tag{5.6}$$

with $\boldsymbol{\Gamma}_j$ and $\boldsymbol{c}_j(t)$ for $j = 1, 2, \ldots, J$, as in Section 3.2.

In addition, using the basis expansion of the functional parameters from (3.11), the vector of functional parameters $\boldsymbol{\beta}(t)$ is expanded as,

$$\boldsymbol{\beta}(t) = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_Q \end{pmatrix} = \begin{pmatrix} \sum_{l=1}^{n_{\beta,1}} \theta_{1l} b_{1l}(t) \\ \sum_{l=1}^{n_{\beta,2}} \theta_{2l} b_{2l}(t) \\ \vdots \\ \sum_{l=1}^{n_{\beta,Q}} \theta_{Ql} b_{Ql}(t) \end{pmatrix} = \boldsymbol{b}^T(t)\boldsymbol{\theta}. \tag{5.7}$$

The matrix $\boldsymbol{b}(t)$ is a $\sum_{q=1}^{Q} n_{\beta,q} \times Q$ block matrix containing the known basis functions from the expansion of the functional parameters and blocks of zeros,

$$\boldsymbol{b}(t) = \begin{pmatrix} \boldsymbol{b}_1(t) & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{b}_2(t) & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{b}_Q(t) \end{pmatrix}, \tag{5.8}$$

with $\boldsymbol{b}_q(t)$ as in (3.11). The vector $\boldsymbol{\theta}$ is the $\sum_{q=1}^{Q} n_{\beta,q} \times 1$ vector of coefficients from the expansion of the functional parameters,

$$\boldsymbol{\theta}^T = \left(\boldsymbol{\theta}_1^T \quad \boldsymbol{\theta}_2^T \quad \cdots \quad \boldsymbol{\theta}_Q^T\right), \tag{5.9}$$

with $\boldsymbol{\theta}_q$ as in (3.11).

The substitution of the basis expansions from (3.11) and (3.12) result in a FLM which takes the form an extended linear model. The extended linear model depending functions of profile factors is given by,

$$\begin{aligned} \boldsymbol{y} &= \int_0^{\mathcal{T}} \boldsymbol{f}^T(\boldsymbol{X}(t))\ \boldsymbol{\beta}(t)\ dt + \boldsymbol{\epsilon} \\ &= \int_0^{\mathcal{T}} \boldsymbol{f}^T(\boldsymbol{X}(t))\ \boldsymbol{b}^T(t)\ dt\ \boldsymbol{\theta} + \boldsymbol{\epsilon} \\ &= \boldsymbol{Z}\boldsymbol{\theta} + \boldsymbol{\epsilon}, \end{aligned} \tag{5.10}$$

with $\boldsymbol{Z}$ the $n \times \sum_{q=1}^{Q} n_{\beta,q}$ model matrix, and $\boldsymbol{\theta}$ the $\sum_{q=1}^{Q} n_{\beta,q} \times 1$ vector of unknown parameters from (5.9). The function $\boldsymbol{f}^T(\boldsymbol{X}(t))$ is the function of the profile factors as defined in (5.4), (5.5), and (5.6), and $\boldsymbol{b}(t)$ the basis functions from the expansion of the functional parameters as defined in (5.8).

The model matrix $\boldsymbol{Z}$ is the solution to an integral with respect to time, of the product of the function $\boldsymbol{f}^T(\boldsymbol{X}(t))$ and the vector of basis functions $\boldsymbol{b}^T(t)$,

$$\int_0^{\mathcal{T}} \boldsymbol{f}^T(\boldsymbol{X}(t))\ \boldsymbol{b}^T(t)\ dt. \tag{5.11}$$

Moreover, $\boldsymbol{Z}$ is partitioned in Q column blocks, with the q^{th} column block a $n \times n_{\beta,q}$ matrix $\boldsymbol{Z}_{\cdot q}$, which is the solution to an integral of the form,

$$\begin{aligned}
\boldsymbol{Z}_{\cdot q} &= \int_0^{\mathcal{T}} f_q(\boldsymbol{X}(t))\ \boldsymbol{b}_q^T(t)\ dt \\
&= \int_0^{\mathcal{T}} f_q\begin{pmatrix} \boldsymbol{x}_{\cdot 1}(t) & \boldsymbol{x}_{\cdot 2}(t) & \cdots & \boldsymbol{x}_{\cdot J}(t) \end{pmatrix}\ \boldsymbol{b}_q^T(t)\ dt \\
&= \int_0^{\mathcal{T}} f_q\begin{pmatrix} \boldsymbol{\Gamma}_1\boldsymbol{c}_1(t) & \boldsymbol{\Gamma}_2\boldsymbol{c}_2(t) & \cdots & \boldsymbol{\Gamma}_J\boldsymbol{c}_J(t) \end{pmatrix}\ \boldsymbol{b}_q^T(t)\ dt, \\
q &= 1, 2, \ldots, Q.
\end{aligned} \tag{5.12}$$

The form of the integrals in (5.12) depends on the specification of the function of the profile factors $\boldsymbol{f}$, i.e., main effects, higher order polynomials and interactions. Finally, $\boldsymbol{b}_q(t)$ is the vector of basis functions from the basis expansion of the q^{th} functional parameter, as before.

The methodology is further expanded in two scenarios. The first scenario is that the FLM depends only on main effects of the profile factors; see Section 5.2. The second scenario is that in addition to the main effects, the FLM depends on higher order polynomials and interactions of the profile factors; see Section 5.3. For both scenarios examples are developed to assist the description of the methodology.

5.2 FLM with main effects of the profile factors

The simplest form of the FLM is the model considering only the main effects for the profile factors. For this case, $\boldsymbol{f}^T(\boldsymbol{X}(t)) = \boldsymbol{X}(t)$ and the number of profile factors is equal to the number of functional parameters, i.e., $J = Q$. The FLM with the main effects of the J profile factors is given by,

$$\begin{aligned}
\boldsymbol{y} &= \int_0^{\mathcal{T}} \boldsymbol{X}(t)\ \boldsymbol{\beta}(t)\ dt + \boldsymbol{\epsilon} \\
&= \int_0^{\mathcal{T}} \boldsymbol{X}(t)\ \boldsymbol{b}^T(t)\ dt\ \boldsymbol{\theta} + \boldsymbol{\epsilon} \\
&= \boldsymbol{Z\theta} + \boldsymbol{\epsilon},
\end{aligned} \tag{5.13}$$

with $\boldsymbol{X}(t)$ and $\boldsymbol{b}(t)$ as defined in (5.3) and (5.8), respectively. The model matrix $\boldsymbol{Z}$ is given by the integral,

$$\int_0^{\mathcal{T}} \boldsymbol{X}(t)\ \boldsymbol{b}^T(t)\ dt,$$

that is the integration of the product of the functions of the profile factors $\boldsymbol{X}(t)$ and the basis functions of the parameters $\boldsymbol{b}^T(t)$, with respect to time. As described in Section 5.1, $\boldsymbol{Z}$ is partitioned in Q column blocks, with the q^{th} block an $n \times n_{\beta,q}$ matrix $\boldsymbol{Z}_{\cdot q}$, $q = 1, 2, \ldots, Q$. For the FLM with only main effects of the profile factors, $\boldsymbol{Z}$ is partitioned in $J = Q$ column blocks, with the j^{th} block an $n \times n_{\beta,j}$ matrix $\boldsymbol{Z}_{\cdot j}$ which is the solution to the integral of the form,

$$\begin{aligned} \boldsymbol{Z}_{\cdot j} &= \int_0^{\mathcal{T}} \boldsymbol{x}_{\cdot j}(t)\ \boldsymbol{b}_j^T(t)\ dt \\ &= \int_0^{\mathcal{T}} \boldsymbol{\Gamma}_j\ \boldsymbol{c}_j(t)\ \boldsymbol{b}_j^T(t)\ dt \\ &= \boldsymbol{\Gamma}_j \int_0^{\mathcal{T}} \boldsymbol{c}_j(t)\ \boldsymbol{b}_j^T(t)\ dt, \quad j = 1, 2, \ldots, J. \end{aligned} \tag{5.14}$$

To assist the understanding of the model development of a FLM with main effects, a few examples are illustrated. Suppose a model with $J = 2$ profile factors with interest only on their main effects. Under this set up, the model is represented as the model in (5.13). The matrix $\boldsymbol{X}(t)$ containing the profile factors is an $n \times 2$ matrix such that,

$$\boldsymbol{X}(t) = \begin{pmatrix} \boldsymbol{x}_{\cdot 1}(t) & \boldsymbol{x}_{\cdot 2}(t) \end{pmatrix},$$

with $\boldsymbol{x}_{\cdot 1}(t)$ and $\boldsymbol{x}_{\cdot 2}(t)$ being $n \times 1$ vectors representing the functions of the profile factors. The vector $\boldsymbol{\beta}(t)$ containing the functional parameters is a 2×1 vector such that,

$$\boldsymbol{\beta}(t) = \begin{pmatrix} \beta_1(t) & \beta_2(t) \end{pmatrix}^T,$$

with $\beta_1(t)$ and $\beta_2(t)$ the functional parameters. The FLM representing the main effects of two profile factors is given by,

$$\begin{aligned} \boldsymbol{y} &= \int_0^{\mathcal{T}} \boldsymbol{X}(t)\ \boldsymbol{\beta}(t)\ dt + \boldsymbol{\epsilon} \\ &= \int_0^{\mathcal{T}} \begin{pmatrix} \boldsymbol{x}_{\cdot 1}(t) & \boldsymbol{x}_{\cdot 2}(t) \end{pmatrix} \begin{pmatrix} \beta_1(t) \\ \beta_2(t) \end{pmatrix}\ dt + \boldsymbol{\epsilon} \\ &= \int_0^{\mathcal{T}} \boldsymbol{x}_{\cdot 1}(t)\ \beta_1(t)\ dt + \int_0^{\mathcal{T}} \boldsymbol{x}_{\cdot 2}(t)\ \beta_2(t)\ dt + \boldsymbol{\epsilon}. \end{aligned} \tag{5.15}$$

To restrict the function space, the basis expansions for the two profile factors are,

$$\boldsymbol{x}_{\cdot 1}(t) = \boldsymbol{\Gamma}_1 \boldsymbol{c}_1(t) \tag{5.16}$$

$$\boldsymbol{x}_{\cdot 2}(t) = \boldsymbol{\Gamma}_2 \boldsymbol{c}_2(t), \tag{5.17}$$

and the basis expansions for the functional parameters are,

$$\beta_1(t) = \boldsymbol{b}_1^T(t)\boldsymbol{\theta}_1 \tag{5.18}$$

$$\beta_2(t) = \boldsymbol{b}_2^T(t)\boldsymbol{\theta}_2. \tag{5.19}$$

After that, the model matrix $\boldsymbol{Z}$ is a $n \times (n_{\beta,1} + n_{\beta,2})$ matrix, that is partitioned in two column blocks. The column blocks have dimensions $n \times n_{\beta,1}$ and $n \times n_{\beta,2}$, and they are the solutions to the integrals,

$$\boldsymbol{Z}_{\cdot 1} = \boldsymbol{\Gamma}_1 \int_0^{\mathcal{T}} \boldsymbol{c}_1(t)\ \boldsymbol{b}_1^T(t)\ dt \tag{5.20}$$

$$\boldsymbol{Z}_{\cdot 2} = \boldsymbol{\Gamma}_2 \int_0^{\mathcal{T}} \boldsymbol{c}_2(t)\ \boldsymbol{b}_2^T(t)\ dt. \tag{5.21}$$

This kind of integrals involve the basis functions for the profile factors, which are assumed to be represented by B-spline basis functions, and the basis functions for the functional parameters, which are assumed to be represented by power series or B-spline basis functions. Analytic computations of integrals that involve products of B-spline basis functions or B-spline and power series basis functions are described in Section 5.9.

Finally, the $(n_{\beta,1} + n_{\beta,2})$ vector of parameters $\boldsymbol{\theta}$, consists of the coefficient vectors from the basis expansion of the functional parameters such that,

$$\boldsymbol{\theta}^T = \begin{pmatrix} \boldsymbol{\theta}_1^T & \boldsymbol{\theta}_2^T \end{pmatrix}.$$

In general, the methodology described is straightforward, especially for the FLM with only main effects of the profile factors. However, when the FLM depends on functions of the profile factors to allow interactions and polynomials of the profile factors, the methodology is more complicated. This is because the number of profile factors is not identical with the number of terms in the model. Further description of the methodology to account for interactions and higher order polynomials is given in the next section.

5.3 FLM with polynomials and interactions

In this section, focus is given on expanding the methodology to functional linear models that include interactions and polynomial effects of the profile factors. Both the interactions and the polynomials are handled identically, as a polynomial can be seen as an interaction of a profile factor with it self. Henceforth, when referring to interaction from this point onwards, the same stands for polynomial effects, and the opposite.

Recall from (5.10) that in matrix form the FLM is formulated as an extended linear model such that,

$$
\begin{aligned}
\boldsymbol{y} &= \int_0^{\mathcal{T}} \boldsymbol{f}^T(\boldsymbol{X}(t))\ \boldsymbol{\beta}(t)\ dt + \boldsymbol{\epsilon} \\
&= \int_0^{\mathcal{T}} \boldsymbol{f}^T(\boldsymbol{X}(t))\ \boldsymbol{b}^T(t)\ dt\ \boldsymbol{\theta} + \boldsymbol{\epsilon} \\
&= \boldsymbol{Z\theta} + \boldsymbol{\epsilon},
\end{aligned}
$$

with $\boldsymbol{y}, \boldsymbol{X}(t), \boldsymbol{f}^T(\boldsymbol{X}(t)), \boldsymbol{\beta}(t), \boldsymbol{\theta}$ and $\boldsymbol{\epsilon}$ as defined before. Interactions and polynomials are embodied in the FLM through the function $\boldsymbol{f}$. In contrast to models with only main effects, models that include the intercept, or interactions and polynomials, imply that $\boldsymbol{f}(\boldsymbol{X}(t)) \neq \boldsymbol{X}(t)$.

Moreover, as a result of adding interactions and polynomials, the number of profile factors is not equal to the number of functional parameters. The reason for the latter is that the number of functional parameters is identical to the number of terms in the model. Thus, the difference between the number of profile factors and the number of functional parameters is the additional terms in the model, i.e., the interactions and polynomials and the intercept if included.

Each functional parameter is allowed a different basis. This means that the basis of the parameter of an interaction term can be different from the basis of the parameters of the main effect terms involved in the interaction. On the other hand, the basis expansion for the profile factors is identical to the model with only main effects. Every profile factor is expanded as in (3.12). For example, the basis of two profile factors involved in an interaction is the same as the basis assumed for the main effects of the two profile factors.

The model matrix $\boldsymbol{Z}$ is a $n \times \sum_{q=1}^{Q} n_{\beta,q}$ matrix, partitioned in Q column blocks, with the q^{th} block an $n \times n_{\beta,q}$ matrix $\boldsymbol{Z}_{\bullet q}$. The J column blocks of the model matrix that correspond to the main effect terms of the profile factors, are defined in the exact same way as the J column blocks of $\boldsymbol{Z}$ in (5.14). The remaining $Q - J$ column blocks $\boldsymbol{Z}_{\bullet q}$, for $q = J+1, J+2, \ldots, Q$ correspond to the interaction and the polynomial terms. Following the latter, a column block $\boldsymbol{Z}_{\bullet q}$ of a K-way interaction or a K-order polynomial of the profile factors $\boldsymbol{x}_{\boldsymbol{\cdot} j_1}(t), \boldsymbol{x}_{\boldsymbol{\cdot} j_2}(t), \ldots, \boldsymbol{x}_{\boldsymbol{\cdot} j_K}(t)$, is the solution to an integral of the form,

$$\begin{aligned} \boldsymbol{Z}_{\boldsymbol{\cdot} q} &= \int_0^{\mathcal{T}} f_q(\boldsymbol{X}(t))\, \boldsymbol{b}_q^T(t)\, dt \\ &= \int_0^{\mathcal{T}} f_q\left(\boldsymbol{x}_{\boldsymbol{\cdot} j_1}(t) \quad \boldsymbol{x}_{\boldsymbol{\cdot} j_2}(t) \quad \cdots \quad \boldsymbol{x}_{\boldsymbol{\cdot} j_K}(t)\right)\, \boldsymbol{b}_q^T(t)\, dt \\ &= \int_0^{\mathcal{T}} \left(\boldsymbol{x}_{\boldsymbol{\cdot} j_1}(t) \circ \boldsymbol{x}_{\boldsymbol{\cdot} j_2}(t) \circ \cdots \circ \boldsymbol{x}_{\boldsymbol{\cdot} j_K}(t)\right) \boldsymbol{b}_q^T(t)\, dt \\ &= \int_0^{\mathcal{T}} \left(\boldsymbol{\Gamma}_{j_1}\boldsymbol{c}_{j_1}(t) \circ \boldsymbol{\Gamma}_{j_2}\boldsymbol{c}_{j_2}(t) \circ \cdots \circ \boldsymbol{\Gamma}_{j_K}\boldsymbol{c}_{j_K}(t)\right) \boldsymbol{b}_q^T(t)\, dt, \quad q = J+1, J+2, \ldots, Q, \end{aligned} \tag{5.22}$$

with $\circ$ the Hadamard product, $f_q(\boldsymbol{X}(t))$ the q^{th} column of $\boldsymbol{f}^T(\boldsymbol{X}(t))$, and $\boldsymbol{b}_q(t)$ the basis functions from the expansion of the q^{th} functional parameter. The Hadamard product of the functions of the profile factors, $\boldsymbol{x}_{\boldsymbol{\cdot} j_1}(t) \circ \boldsymbol{x}_{\boldsymbol{\cdot} j_2}(t) \circ \cdots \circ \boldsymbol{x}_{\boldsymbol{\cdot} j_K}(t)$, represents a K-way interaction of the functions of the profile factors $\boldsymbol{x}_{\boldsymbol{\cdot} j_1}(t), \boldsymbol{x}_{\boldsymbol{\cdot} j_2}(t), \ldots, \boldsymbol{x}_{\boldsymbol{\cdot} j_K}(t)$ when $j_1 \neq j_2 \neq \cdots \neq j_K$, or a K-order polynomial when $j_1 = j_2 = \cdots = j_K$.

The equation (5.22) can also be expressed as,

$$\begin{aligned} \boldsymbol{Z}_{\boldsymbol{\cdot} q} &= \boldsymbol{\Gamma}_{j_1 j_2 \cdots j_K} \int_0^{\mathcal{T}} \left(\boldsymbol{c}_{j_1}(t) \otimes \boldsymbol{c}_{j_2}(t) \otimes \cdots \otimes \boldsymbol{c}_{j_K}(t)\right) \boldsymbol{b}_q^T(t)\, dt \\ &= \boldsymbol{\Gamma}_{j_1 j_2 \cdots j_K} \int_0^{\mathcal{T}} \boldsymbol{c}_{j_1 j_2 \cdots j_K}(t)\, \boldsymbol{b}_q^T(t)\, dt, \quad q = J+1, J+2, \ldots, Q, \end{aligned} \tag{5.23}$$

with $\otimes$ the Kronecker product, and $\boldsymbol{\Gamma}_{j_1 j_2 \cdots j_K}$ and $\boldsymbol{c}_{j_1 j_2 \cdots j_K}(t)$ as described below. For an example see Appendix C.

First, the matrix $\boldsymbol{\Gamma}_{j_1 j_2 \cdots j_K}$ is a $n \times \prod_{k=1}^{K} n_{x,j_k}$ coefficient matrix for which each column is the Hadamard product of the form:

$$col_{l_1}(\boldsymbol{\Gamma}_{j_1}) \circ col_{l_2}(\boldsymbol{\Gamma}_{j_2}) \circ \cdots \circ col_{l_k}(\boldsymbol{\Gamma}_{j_K}),$$

with $col_l(\boldsymbol{\Gamma})$ the l^{th} column of the matrix $\boldsymbol{\Gamma}$, and $l_1, l_2, \ldots, l_K$ arbitrary choices of column index for matrices $\boldsymbol{\Gamma}_{j_1}, \boldsymbol{\Gamma}_{j_2}, \ldots, \boldsymbol{\Gamma}_{j_K}$, respectively. The complete set of columns $\boldsymbol{\Gamma}_{j_1}, \boldsymbol{\Gamma}_{j_2}, \ldots, \boldsymbol{\Gamma}_{j_K}$ is formed by considering all possible choices of $l_1, l_2, \ldots, l_K$, arranged in lexicographical order.

Second, the vector $\boldsymbol{c}_{j_1 j_2 \cdots j_k}(t)$ is a $\prod_{k=1}^{K} n_{x,j_k} \times 1$ vector, that is defined as:

$$\boldsymbol{c}_{j_1 j_2 \cdots j_k}(t) = \boldsymbol{c}_{j_1}(t) \otimes \boldsymbol{c}_{j_2}(t) \otimes \cdots \otimes \boldsymbol{c}_{j_K}(t),$$

with $\otimes$ the Kronecker product, for which for $m \times h$ matrix $\boldsymbol{A}$ and $p \times s$ matrix bmB is defined as:

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{pmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \cdots & a_{1h}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \cdots & a_{2h}\boldsymbol{B} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1}\boldsymbol{B} & a_{m2}\boldsymbol{B} & \cdots & a_{mh}\boldsymbol{B} \end{pmatrix}.$$

Each entry of $\boldsymbol{c}_{j_1 j_2 \cdots j_k}(t)$ is a $\prod_{j=j_1}^{j_k} n_{x,j} \times 1$ is a product of j_K basis functions, one for each different profile factor, and all possible combinations of choices are considered in a way similar to the construction of the columns of $\boldsymbol{\Gamma}_{j_1 j_2 \cdots j_K}$. Since the basis chosen for the profile factors are B-spline basis functions, then each entry is the product of j_k B-spline basis functions. An analytic derivation of integrals involving products of B-spline basis functions is described in Section 5.9.

The example with two profile factors from Section 5.2 is continued, to assist the understanding of functional linear models with interactions and polynomials of the profile factors. To avoid overcomplicating the methodology, and to keep the computations simple, a two-way interaction and the quadratic effect of a profile factor are considered. The two-way interaction is the interaction of the first and second profile factor which is $\boldsymbol{x}_{\cdot 1}(t) \circ \boldsymbol{x}_{\cdot 2}(t)$, and the quadratic polynomial is the quadratic effect of the first profile factor which is $\boldsymbol{x}_{\cdot 1}(t) \circ \boldsymbol{x}_{\cdot 1}(t)$. Moreover, it is assumed that the FLM includes the intercept as well. Thus, the matrix $\boldsymbol{f}^T(\boldsymbol{X}(t))$ that contains the functions of the profile factors, is a $n \times 5$ matrix defined as,

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{X}(t)) &= \begin{pmatrix} f_1(\boldsymbol{X}(t)) & f_2(\boldsymbol{X}(t)) & f_3(\boldsymbol{X}(t)) & f_4(\boldsymbol{X}(t)) & f_5(\boldsymbol{X}(t)) \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{1}_n & \boldsymbol{x}_{\cdot 1}(t) & \boldsymbol{x}_{\cdot 2}(t) & \boldsymbol{x}_{\cdot 1}(t) \circ \boldsymbol{x}_{\cdot 1}(t) & \boldsymbol{x}_{\cdot 1}(t) \circ \boldsymbol{x}_{\cdot 2}(t) \end{pmatrix}, \end{aligned}$$

with $\boldsymbol{1}_n$ being the $n \times 1$ vector of 1's, $\boldsymbol{x}_{\cdot 1}(t)$ and $\boldsymbol{x}_{\cdot 2}(t)$ being $n \times 1$ vectors representing the functions of the profile factors, and $\boldsymbol{x}_{\cdot 1}(t) \circ \boldsymbol{x}_{\cdot 2}(t)$ and $\boldsymbol{x}_{\cdot 1}(t) \circ \boldsymbol{x}_{\cdot 1}(t)$ being $n \times 1$ vectors representing the interaction and the quadratic effect. Finally, the vector $\boldsymbol{\beta}(t)$ is a 5×1 vector containing the functional parameters such that,

$$\boldsymbol{\beta}^T(t) = \begin{pmatrix} \beta_1 & \beta_2(t) & \beta_3(t) & \beta_4(t) & \beta_5(t) \end{pmatrix},$$

with β_1 the constant parameter, $\beta_2(t)$ and $\beta_3(t)$ the functional parameters for the two profile factors and $\beta_4(t)$ and $\beta_5(t)$ the functional parameters of the quadratic effect and interaction, respectively. Hence, the FLM representing the main effects of two profile factors, the quadratic effect of the first profile factor and the interaction between the two profile factors is formulated as,

$$\begin{aligned} \boldsymbol{y} = \mathbf{1}_n \beta_1 &+ \int_0^{\mathcal{T}} \boldsymbol{x}_{\cdot 1}(t)\ \beta_2(t)\ dt + \int_0^{\mathcal{T}} \boldsymbol{x}_{\cdot 2}(t)\ \beta_3(t)\ dt \\ &+ \int_0^{\mathcal{T}} \big(\boldsymbol{x}_{\cdot 1}(t) \circ \boldsymbol{x}_{\cdot 1}(t)\big)\ \beta_4(t)\ dt + \int_0^{\mathcal{T}} \big(\boldsymbol{x}_{\cdot 1}(t) \circ \boldsymbol{x}_{\cdot 2}(t)\big)\ \beta_5(t)\ dt + \boldsymbol{\epsilon}. \end{aligned} \tag{5.24}$$

To restrict the function space, the basis expansions for the two profile factors are identical to the basis expansions defined in (5.16) and (5.17). The basis expansions for the functional parameters are not identical to the example for the main effects. This is because, extra parameters are involved, due to the addition of the intercept, the quadratic effect and interaction terms. The parameter corresponding to the intercept is the constant parameter. The basis expansions of the functional parameters are:

$$\beta_1 = \theta_1,$$

$$\begin{aligned} \beta_2(t) &= \boldsymbol{b}_2^T(t)\boldsymbol{\theta}_2, \\ \beta_3(t) &= \boldsymbol{b}_3^T(t)\boldsymbol{\theta}_3, \\ \beta_4(t) &= \boldsymbol{b}_4^T(t)\boldsymbol{\theta}_4, \\ \beta_5(t) &= \boldsymbol{b}_5^T(t)\boldsymbol{\theta}_5. \end{aligned}$$

The number of columns of the model matrix $\boldsymbol{Z}$ depends on the choice of the number of basis functions from the expansion of the functional parameters. The first column block of the model matrix is a column vector of 1's for the intercept. The rest of the columns arise in four column blocks from the solutions of four integrals. The two column blocks for the main effects are identical to the columns blocks from the example with the main effects, and they are the solutions to the integrals defined in (5.20) and (5.21). The remaining two column blocks of the model matrix, that involve the quadratic effect and the interaction, can be found using the methodology in (5.23). For instance, the column block for the interaction term is defined as,

$$\begin{aligned} \boldsymbol{Z}_{\cdot 5} &= \int_0^{\mathcal{T}} \big(\boldsymbol{x}_{\cdot 1}(t) \circ \boldsymbol{x}_{\cdot 2}(t)\big)\ \boldsymbol{b}_5^T(t)\ dt \\ &= \int_0^{\mathcal{T}} \big(\boldsymbol{\Gamma}_1 \boldsymbol{c}_1(t) \circ \boldsymbol{\Gamma}_2 \boldsymbol{c}_2(t)\big)\ \boldsymbol{b}_5^T(t)\ dt \\ &= \boldsymbol{\Gamma}_{12} \int_0^{\mathcal{T}} \big(\boldsymbol{c}_1(t) \otimes \boldsymbol{c}_2(t)\big)\ \boldsymbol{b}_5^T(t)\ dt \\ &= \boldsymbol{\Gamma}_{12} \int_0^{\mathcal{T}} \boldsymbol{c}_{12}(t)\ \boldsymbol{b}_5^T(t)\ dt. \end{aligned} \tag{5.25}$$

The vector $\boldsymbol{c}_{12}(t)$ is a $(n_{x,1} \times n_{x,2}) \times 1$ vector. This means that, the length of the vector $\boldsymbol{c}_{12}(t)$ is the product of the number of basis functions of the two profile factors which are involved in the interaction. Moreover, each entry in the vector $\boldsymbol{c}_{12}(t)$ corresponds to the product of two B-spline basis functions. Thus, together with the basis of the functional parameter, the integral from (5.25) is an integral of the product of three B-spline basis functions, or two B-spline and a power series basis functions. In both cases, such integrals are analytically tractable using the method described in Section 5.9.

A challenging part of (5.25) is the construction of the coefficient matrix $\mathbf{\Gamma}_{12}$. The coefficient matrix $\mathbf{\Gamma}_{12}$ is a $n \times (n_{x,1} \times n_{x,2})$ matrix, with each column being the Hadamard product of basis coefficients from the expansion of $\boldsymbol{x}_{\cdot 1}(t)$ and a basis function from the expansion of $\boldsymbol{x}_{\cdot 2}(t)$. Meaning that, the columns of $\mathbf{\Gamma}_{12}$ contain all possible product combinations of the basis coefficients of the two profile factors involved in the interaction.

For instance, from the basis expansion of the profile factors, the coefficient matrices $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$ are of size $n \times n_{x,1}$ and $n \times n_{x,2}$ respectively,

$$\mathbf{\Gamma}_1 = \begin{pmatrix} \Gamma_{1_{11}} & \Gamma_{1_{12}} & \cdots & \Gamma_{1_{1n_{x,1}}} \\ \Gamma_{1_{21}} & \Gamma_{1_{22}} & \cdots & \Gamma_{1_{2n_{x,1}}} \\ \vdots & \vdots & \vdots & \vdots \\ \Gamma_{1_{n1}} & \Gamma_{1_{n2}} & \cdots & \Gamma_{1_{nn_{x,1}}} \end{pmatrix}, \qquad \mathbf{\Gamma}_2 = \begin{pmatrix} \Gamma_{2_{11}} & \Gamma_{2_{12}} & \cdots & \Gamma_{2_{1n_{x,2}}} \\ \Gamma_{2_{21}} & \Gamma_{2_{22}} & \cdots & \Gamma_{2_{2n_{x,2}}} \\ \vdots & \vdots & \vdots & \vdots \\ \Gamma_{2_{n1}} & \Gamma_{2_{n2}} & \cdots & \Gamma_{2_{nn_{x,2}}} \end{pmatrix},$$

with $\Gamma_{1_{il}}$ and $\Gamma_{2_{il}}$ the il^{th} entries of the matrices $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$, respectively. Then, the columns of the coefficient matrix $\mathbf{\Gamma}_{12}$ are the Hadamard products of all the possible combinations of the columns of $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$ such that,

$$\mathbf{\Gamma}_{12} = \begin{pmatrix} \Gamma_{1_{11}} \times \Gamma_{2_{11}} & \Gamma_{1_{11}} \times \Gamma_{2_{12}} & \cdots & \Gamma_{1_{11}} \times \Gamma_{2_{nn_{x,2}}} & \cdots & \Gamma_{1_{1n_{x,1}}} \times \Gamma_{2_{11}} & \Gamma_{1_{1n_{x,1}}} \times \Gamma_{2_{12}} & \cdots & \Gamma_{1_{1n_{x,1}}} \times \Gamma_{2_{nn_{x,2}}} \\ \Gamma_{1_{21}} \times \Gamma_{2_{21}} & \Gamma_{1_{21}} \times \Gamma_{2_{22}} & \cdots & \Gamma_{1_{21}} \times \Gamma_{2_{nn_{x,2}}} & \cdots & \Gamma_{1_{2n_{x,1}}} \times \Gamma_{2_{21}} & \Gamma_{1_{2n_{x,1}}} \times \Gamma_{2_{22}} & \cdots & \Gamma_{1_{2n_{x,1}}} \times \Gamma_{2_{nn_{x,2}}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \Gamma_{1_{n1}} \times \Gamma_{2_{n1}} & \Gamma_{1_{n1}} \times \Gamma_{2_{n2}} & \cdots & \Gamma_{1_{n1}} \times \Gamma_{2_{nn_{x,2}}} & \cdots & \Gamma_{1_{nn_{x,1}}} \times \Gamma_{2_{n1}} & \Gamma_{1_{nn_{x,1}}} \times \Gamma_{2_{n2}} & \cdots & \Gamma_{1_{nn_{x,1}}} \times \Gamma_{2_{nn_{x,2}}} \end{pmatrix}.$$

The procedure for the quadratic effect is the same, but $\boldsymbol{x}_2(t)$ has to be replaced by $\boldsymbol{x}_1(t)$.

Finally, the vector of parameters $\boldsymbol{\theta}$ is a $\sum_{q=1}^{5} n_{\beta,q} \times 1$ vector, that contains the coefficient vectors from the basis expansion of the functional parameters such that,

$$\boldsymbol{\theta}^T = \begin{pmatrix} \theta_1 & \boldsymbol{\theta}_2^T & \boldsymbol{\theta}_3^T & \boldsymbol{\theta}_4^T & \boldsymbol{\theta}_5^T \end{pmatrix}.$$

For further examples and sensitivity studies for functional models containing interactions and quadratic effects of profile factors; see Sections 6.4 and 7.8.

5.4 Frequentist approach to the design problem

The model in equation (5.10) takes the form of the traditional linear model, with $\boldsymbol{Z}$ being the model matrix and $\boldsymbol{\theta}$ the unknown coefficients, with dimensions of $\boldsymbol{Z}$ and $\boldsymbol{\theta}^T$ being $n \times \sum_{q=1}^{Q} n_{\beta,q}$ and $1 \times \sum_{q=1}^{Q} n_{\beta,q}$, respectively. As the FLM with basis expansions is an extension to the traditional linear model, the same methods and techniques can be used.

An estimator of $\boldsymbol{\theta}$, denoted as $\hat{\boldsymbol{\theta}}$, can be found by the ordinary least squares method which minimises the residual sum of squares,

$$RSS = (\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\theta})^T(\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\theta}). \tag{5.26}$$

After that, the parameter estimator is defined as,

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{Z}^T\boldsymbol{y}. \tag{5.27}$$

Another important equation to derive is the variance of the estimated parameters. This is because some objective functions including the A- and D- optimality are functions of the variance-covariance matrix. The variance of $\hat{\boldsymbol{\theta}}$ is given by,

$$\begin{aligned}\text{Var}(\hat{\boldsymbol{\theta}}) &= \text{Var}\Big[(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{Z}^T\boldsymbol{y}\Big] \\ &= \sigma^2(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}.\end{aligned} \tag{5.28}$$

For the rest of the thesis, the objective criteria considered are the A- and D- optimality. Other objective criteria including G-optimality, E-optimality and more, can be considered in a similar way; see Atkinson et al. (2007, Chapters 6,9, and 10). The A- and D-optimality objective functions for the FLM, denoted as $\Psi_A(\Gamma)$ and $\Psi_D(\Gamma)$, are the trace and determinant of the variance-covariance matrix of the parameter estimates and using (2.6) and (2.8) respectively, they are defined as,

$$\Psi_A(\boldsymbol{\Gamma}) = \text{tr}\big[(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\big] \tag{5.29}$$

$$\Psi_D(\boldsymbol{\Gamma}) = \det\big[(\boldsymbol{Z}^T\boldsymbol{Z})\big]^{-1/p} = \exp\Big\{-\frac{1}{p}\log\big[\det\big((\boldsymbol{Z}^T\boldsymbol{Z})\big)\big]\Big\}, \tag{5.30}$$

for a design $\boldsymbol{\Gamma}$, which is the coefficient matrix $\boldsymbol{\Gamma}$ from the basis expansion of the profile factors assigned to each run of the experiment, and $p = \sum_{q=1}^{Q} n_{\beta,q}$ the total number of basis functions of the functional parameters, i.e., the total number of columns of $\boldsymbol{Z}$. A design $\boldsymbol{\Gamma}^* \in \mathcal{X}$ where $\mathcal{X}$ is the design space, is A- and D- optimal if it minimises $\Psi_A(\boldsymbol{\Gamma})$ and $\Psi_D(\boldsymbol{\Gamma})$, respectively. Note that, the objective criterion in equations (5.29) require the information matrix to be invertible. The D-optimality objective function used is scaled based on the total number of basis functions of the parameters. The second equality in (5.30) is on the log scale, to increase the computational stability.

The objective functions in equations (5.29) and (5.30) depend on the model matrix $\boldsymbol{Z}$ which is partitioned in Q column blocks resulting from the solutions of integrals of the form of (5.11). The solution of the integrals, and hence of $\boldsymbol{Z}$, depend on the basis of the profile factors, the basis of the functional parameters and the coefficient matrices from the basis expansions of the profile factors which are the design matrices. Since the basis as well as the number of basis functions of the profile factors and the functional parameters are subject of choice, the missing components in $\boldsymbol{Z}$ are the coefficients. Hence, the design of experiments problem is reduced to the optimisation of the coefficient matrices from the expansion of the profile factors.

The profile factors in this thesis are assumed to be constrained by $u \leq x_{ij}(t) \leq v$. The BS basis for the functions of the profile factors are useful and contribute to achieve the required bounds imposed by the constraint. This is because the property of the BS basis in (4.14) implies that any bounds imposed on the coefficients from the expansion in (3.12) are equivalent bounds on the functions of the profile factors.

5.5 Roughness penalty approach

The use of basis expansions allowed the FLM to be expanded in a way that allows common statistical techniques to be used to fit the model. The choice of basis functions as well as the choice of degree and choice of knots is usually difficult. A bad choice of basis or a bad choice of the basis coefficients can result in rough, complicated and wiggly functions for the parameters. In this section, an alternative roughness penalty approach; of which the aim is to penalise the complexity of functions is discussed.

Penalisation of the complexity of functions is achieved via a smoothing penalty added on the residual sum of squares. As a result, the residual sum of squares defined in (5.26) is updated to the penalised residual sum of squares; see Ramsay and Silverman (2005). The penalised residual sum of squares is defined as,

$$PRSS = (\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\theta})^T(\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\theta}) + \Lambda\boldsymbol{\theta}^T\boldsymbol{R}_0\boldsymbol{\theta},$$

where $\Lambda > 0$ is a scalar smoothing parameter and $\boldsymbol{R}_0$ is a $\sum_{q=1}^{Q} n_{\beta,q} \times \sum_{q=1}^{Q} n_{\beta,q}$ block diagonal matrix, representing the roughness penalties. Under the roughness penalty approach, the parameter estimator is updated to become,

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{Z}^T\boldsymbol{Z} + \Lambda\boldsymbol{R}_0)^{-1}\boldsymbol{Z}^T\boldsymbol{y}. \tag{5.31}$$

The q^{th} block entry of the roughness penalties is a matrix of dimensions $n_{\beta,q} \times n_{\beta,q}$ representing the penalisation of the q^{th} term in the FLM and it is given by,

$$\boldsymbol{R}_{0_{\beta,q}} = \int_0^{\mathcal{T}} \left[D^{(2)}[\boldsymbol{b}_q(t)]\right] \left[D^{(2)}[\boldsymbol{b}_q(t)^T]\right] dt, \quad q = 1, 2, \ldots, Q, \tag{5.32}$$

where $\boldsymbol{b}_q(t)^T = [b_{q1}(t), \ldots., b_{q,n_{\beta,q}}(t)]$ are known basis functions for the functional parameters and $D^{(2)}(\cdot)$ is the second derivative.

Scalar parameters, including the constant parameter for the intercept, are not penalised. i.e., their roughness penalties are zero. If the q^{th} parameter represents the parameter of a scalar factor, then $n_{\beta,q} = 1$, and the q^{th} block entry is a scalar. Thus, if the q^{th} parameter represents the parameter of a scalar factor, $D^{(2)}\boldsymbol{b}_q(t) = 0$, hence the roughness penalty is zero as required.

The nature of the parameters in a FLM fall under three special cases. Assume a FLM with Q terms, meaning Q parameters. The three special cases are: all Q parameters are functional parameters representing functions of profile factors; one parameter represents the constant scalar parameter and the rest $Q-1$ parameters are functional parameters; or one parameter represents the constant scalar parameter, Q_1 parameters are scalar parameters and Q_2 parameters are functional parameters, for $1 + Q_1 + Q_2 = Q$.

In the first space case, all parameters are penalised, exactly using the definition in (5.32). In the second special case, the roughness matrix is defined as,

$$\boldsymbol{R}_0 = \begin{pmatrix} 0 & \mathbf{0}_Q^T \\ \mathbf{0}_Q & \boldsymbol{R}_p \end{pmatrix},$$

where $\mathbf{0}_Q^T$ is the $\sum_{q=2}^{Q} n_{\beta,q} \times 1$ vector of zeros, and $\boldsymbol{R}_p$ is a $\sum_{q=2}^{Q} n_{\beta,q} \times \sum_{q=2}^{Q} n_{\beta,q}$ block diagonal matrix representing the roughness penalties of the functional parameters. Finally, in the third special case, the roughness matrix is defined as,

$$\boldsymbol{R}_0 = \begin{pmatrix} 0 & \mathbf{0}_{Q_1}^T & \mathbf{0}_{Q_2}^T \\ \mathbf{0}_{Q_1} & \mathbf{0}_{Q_1Q_1} & \mathbf{0}_{Q_1Q_2} \\ \mathbf{0}_{Q_2} & \mathbf{0}_{Q_1Q_2}^T & \boldsymbol{R}_p \end{pmatrix},$$

where $\mathbf{0}_{Q_1}^T$ is the $Q_1 \times 1$ vector of zeros, $\mathbf{0}_{Q_2}^T$ is the $\sum_{q=Q_1+2}^{Q} n_{\beta,q} \times 1$ vector of zeros, $\mathbf{0}_{Q_1Q_2}$ is the $Q_1 \times \sum_{q=Q_1+2}^{Q} n_{\beta,q}$ matrix of zeros, $\mathbf{0}_{Q_1Q_1}$ is the $Q_1 \times Q_1$ matrix of zeros, and lastly, $\boldsymbol{R}_p$ is a $\sum_{q=Q_1+2}^{Q} n_{\beta,q} \times \sum_{q=Q_1+2}^{Q} n_{\beta,q}$ block diagonal matrix representing the roughness penalties of the functional parameters.

5.6 Bayesian approach to the design problem

In this section, the design problem is fitted using the Bayesian approach that have been discussed in Section 2.4. The expected utility functions derived previously are expanded to build Bayesian objective functions for the FLM. After that, a connection between the roughness penalty approach and the Bayesian experimental designs for the FLM is developed.

The FLM have been proven that it is an extension to the linear model; see (5.10). Thus, results for Bayesian experimental designs for the linear model from Section 2.4 still hold for the FLM. In comparison to the traditional linear model discussed earlier, the model matrix and vector of parameters for the FLM is $\boldsymbol{Z}$ and $\boldsymbol{\theta}$ instead of $\boldsymbol{F}$ and $\boldsymbol{\beta}$. Moreover, the design for the FLM is the coefficient matrices $\boldsymbol{\Gamma}_j$ from the basis expansion of each profile factor from $j = 1, 2, \ldots, J$ instead of $\boldsymbol{X}$ that is the design for the linear model.

The prior specification of the FLM is assumed to be identical to the linear model in Section 2.3.1. The latter allows the derivation of the likelihood and the joint prior distribution which are Normal and Normal Inverse Gamma distributions; see (2.13), (2.14) and (2.15),

$$\begin{aligned} \pi(\boldsymbol{y}|\boldsymbol{\theta}, \sigma^2) &\sim N(\boldsymbol{Z}\boldsymbol{\theta}, \sigma^2 \boldsymbol{I}_n) \\ \pi(\boldsymbol{\theta}, \sigma^2) &\sim NIG(\boldsymbol{\mu}, \boldsymbol{V}, a/2, b/2), \end{aligned} \tag{5.33}$$

respectively, where $\boldsymbol{\mu}$ is the $\sum_{q=1}^{Q} n_{\beta,q}$ prior mean vector of $\boldsymbol{\theta}$, $\boldsymbol{V}$ is a known and symmetric $\sum_{q=1}^{Q} n_{\beta,q} \times \sum_{q=1}^{Q} n_{\beta,q}$ matrix, and a, b are hyperparameters. Using the result from (2.16) and (2.17), the joint posterior distribution of the FLM is also a normal inverse gamma distribution such that,

$$\pi(\boldsymbol{\theta}, \sigma^2|\boldsymbol{y}) \sim NIG(\boldsymbol{\theta}_N, \boldsymbol{V}_N, a^*/2, b^*/2) \tag{5.34}$$

where,

$$\begin{aligned} \boldsymbol{V}_N &= (\boldsymbol{Z}^T\boldsymbol{Z} + \boldsymbol{V}^{-1})^{-1} \\ \boldsymbol{\theta}_N &= \boldsymbol{V}_N(\boldsymbol{V}^{-1}\boldsymbol{\mu} + \boldsymbol{Z}^T\boldsymbol{y}) \\ a^* &= a + n \\ b^* &= b + (\boldsymbol{\mu}^T\boldsymbol{V}^{-1}\boldsymbol{\mu} + \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{\theta}_N^T\boldsymbol{V}_N^{-1}\boldsymbol{\theta}_N). \end{aligned} \tag{5.35}$$

In addition, the marginal posterior distributions for $\boldsymbol{\theta}$ is a multivariate t-distribution such that,

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) \sim t_{a^*}\left(\boldsymbol{\theta}_N, \frac{b^*}{a^*}\boldsymbol{V}_N\right), \tag{5.36}$$

with a^* the degrees of freedom, mean $\boldsymbol{\theta}_N$ and scale $\frac{b^*}{a^*}\boldsymbol{V}_N$, for $a^*, b^*, \boldsymbol{\theta}_N$ and $\boldsymbol{V}_N$ as in (5.35). The marginal posterior distribution for σ^2 is an inverse gamma distribution,

$$\pi(\sigma^2|\boldsymbol{y}) \sim IG(a^*/2, b^*/2), \tag{5.37}$$

for a^* and b^* as in (5.35).

In the Bayesian framework, the experimental aim is represented through a utility function; see Section 2.4. Two common utility function discussed earlier are the Negative Squared Error Loss and the Shannon Information Gain. In terms of the FLM, the NSEL utility function is defined as,

$$u(\boldsymbol{\theta}, \boldsymbol{y}, \boldsymbol{\Gamma}) = -\left[\boldsymbol{\theta} - \mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{\Gamma})\right]^T\left[\boldsymbol{\theta} - \mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{\Gamma})\right], \tag{5.38}$$

and optimal designs are the designs that maximise the expected utility of NSEL. The expected utility is derived by taking the expectation over the responses and the parameter space. Using the derivation from (2.23), the expected utility of the NSEL for the FLM is,

$$\Psi_{nsel}(\boldsymbol{\Gamma}) = -\frac{b}{a-2}\mathrm{tr}\left[\left(\boldsymbol{Z}^T\boldsymbol{Z} + \boldsymbol{V}^{-1}\right)^{-1}\right]. \tag{5.39}$$

The objective function in (5.39) is known as Bayesian A-optimality, and a design that maximise the expected utility is defined as a Bayesian A-optimal design. The expected utility is proportional to the trace of the posterior variance-covariance matrix,

$$\Psi_{nsel}(\boldsymbol{\Gamma}) \propto -\mathrm{tr}\left[\left(\boldsymbol{Z}^T\boldsymbol{Z} + \boldsymbol{V}^{-1}\right)^{-1}\right]. \tag{5.40}$$

and a design $\boldsymbol{\Gamma}^*$ is optimal if it minimises the trace of the inverse of the posterior variance-covariance matrix,

$$\Psi_{nsel}(\boldsymbol{\Gamma}^*) = \min_{\boldsymbol{\Gamma}\in\mathcal{X}} \mathrm{tr}\left[\left(\boldsymbol{Z}^T\boldsymbol{Z} + \boldsymbol{V}^{-1}\right)^{-1}\right]. \tag{5.41}$$

Under a non-informative prior where $\boldsymbol{V}^{-1}$ would be zero, this criterion would minimise the trace of the inverse of the information matrix as in (5.29).

The SIG utility function in terms of the FLM is defined as,

$$\begin{aligned} u(\boldsymbol{\theta}, \boldsymbol{y}, \boldsymbol{\Gamma}) &= \log \pi(\boldsymbol{\theta}|\boldsymbol{y}) - \log \pi(\boldsymbol{\theta}) \\ &= \log \pi(\boldsymbol{y}|\boldsymbol{\theta}) - \log \pi(\boldsymbol{y}), \end{aligned} \tag{5.42}$$

with the second equality sometimes more useful for computations; see (2.26) for the middle steps and the description thereafter. The expected utility of the SIG for the FLM is equivalent to the determinant of the inverse of the posterior variance-covariance matrix,

$$\Psi_{sig}(\boldsymbol{\Gamma}) = |\boldsymbol{Z}^T\boldsymbol{Z} + \boldsymbol{V}^{-1}|. \tag{5.43}$$

and a design $\boldsymbol{\Gamma}^*$ is optimal if it maximises the determinant of the inverse of the posterior variance-covariance matrix,

$$\Psi_{sig}(\boldsymbol{\Gamma}^*) = \max_{\boldsymbol{\Gamma}\in\mathcal{X}} |\boldsymbol{Z}^T\boldsymbol{Z} + \boldsymbol{V}^{-1}|. \tag{5.44}$$

The objective function in (5.44) is known as Bayesian D-optimality and a design that achieves to maximise the expected utility is defined as a Bayesian D-optimal design. Under a non-informative prior where $\boldsymbol{V}$ would be zero, this criterion maximises the logarithm of the determinant of the information matrix $\boldsymbol{Z}^T\boldsymbol{Z}$.

Finally, using the roughness penalty approach there is a clear connection to be made between the frequentist and the Bayesian approaches. The connection comes from the equations of the parameter estimator under the roughness penalty approach in (5.31) and the posterior mean in (5.35). It is clear to notice that the prior precision matrix is defined to be $\boldsymbol{V}^{-1} = \Lambda\boldsymbol{R}_0$ when the prior mean is centered around zero. The frequentist approach is identical to choosing $\Lambda = 0$ and thus, matrix $\boldsymbol{R}_p$ corresponds to the roughness matrix and the smoothing value Λ controls the wiggliness of a function. Meaning that, a value of Λ being zero, corresponds to uncertain prior choice and no smoothness, but a value of Λ tending towards infinity corresponds to a strong prior choice and heavily penalised functions.

5.7 Bayesian design criterion tailored for profile factors

In this section, the aim is to develop a Bayesian design criterion for profile factors, to tailor the design to the FLM and tackle to minimise the average posterior variance of the functional parameters $\boldsymbol{\beta}(t)$ directly, averaged with respect to time. A Bayesian approach is followed to derive the new objective function which takes into account directly the variance of the function $\boldsymbol{\beta}(t)$.

The NSEL utility function from (5.38) is applied, but with the function of the model parameters being represented by $\gamma(t) = \boldsymbol{C}^T(t)\boldsymbol{\theta}$, instead of $\boldsymbol{\beta}(t)$. The use of $\gamma(t)$ instead of $\boldsymbol{\beta}(t)$ is to allow picking up the right parameters and functions of interest and not all of the functional parameters. This is possible through the $\sum_{q=1}^{Q} n_{\beta,q} \times Q$ matrix $\boldsymbol{C}(t)$, with each of the entries defining the functions of interest from a FLM. For example,

$\gamma(t) = \boldsymbol{\beta}(t)$ is a special case for which the matrix $\boldsymbol{C}(t)$ includes all the basis functions from the expansion of the functional parameters.

The tailored utility function for profile factors with $\gamma(t) = \boldsymbol{C}^T(t)\boldsymbol{\theta}$ representing the function of the model parameters and $\hat{\gamma}(t) = \boldsymbol{C}^T(t)\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y})$ representing the function of the model estimated parameters, is defined as,

$$\begin{aligned} u(\hat{\gamma}(t),\boldsymbol{y},\boldsymbol{\Gamma}) &= -\int_0^{\mathcal{T}} \left[\gamma(t)-\hat{\gamma}(t)\right]^T\left[\gamma(t)-\hat{\gamma}(t)\right]\,dt \\ &= -\int_0^{\mathcal{T}} \left[\boldsymbol{C}^T(t)\boldsymbol{\theta}-\boldsymbol{C}^T(t)\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{\Gamma})\right]^T\left[\boldsymbol{C}^T(t)\boldsymbol{\theta}-\boldsymbol{C}^T(t)\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{\Gamma})\right]\,dt \\ &= -\int_0^{\mathcal{T}} \left[\boldsymbol{C}^T(t)\{\boldsymbol{\theta}-\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{\Gamma})\}\right]^T\left[\boldsymbol{C}^T(t)\{\boldsymbol{\theta}-\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{\Gamma})\}\right]\,dt \\ &= -\int_0^{\mathcal{T}} \{\boldsymbol{\theta}-\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{\Gamma})\}^T\boldsymbol{C}(t)\boldsymbol{C}^T(t)\{\boldsymbol{\theta}-\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{\Gamma})\}\,dt. \end{aligned} \tag{5.45}$$

To obtain the expected utility under the Bayesian design criterion for profile factors, the quadratic form property by Mathai and Provost (1992, p. 424) is used. The quadratic form property is that, for a vector $\boldsymbol{e}$ of p random variables and Ξ a symmetric matrix of dimensions $p \times p$ then,

$$\mathbb{E}(\boldsymbol{e}^T\Xi\boldsymbol{e}) = \mathrm{tr}(\Xi\boldsymbol{\Sigma}) + \boldsymbol{\mu}^T\Xi\boldsymbol{\mu}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and variance-covariance matrix of $\boldsymbol{e}$. Hence, the expected utility of the Bayesian design criterion tailored on profile factors is given by,

$$\begin{aligned} \Psi(\boldsymbol{\Gamma}) &= \mathbb{E}_{\boldsymbol{y},\boldsymbol{\theta},\sigma^2}\Big(-\int_0^{\mathcal{T}} \left[\boldsymbol{\theta}-\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{\Gamma})\right]^T\boldsymbol{C}(t)\boldsymbol{C}^T(t)\left[\boldsymbol{\theta}-\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{\Gamma})\right]\,dt\Big) \\ &= \mathbb{E}_{\boldsymbol{y},\sigma^2}\left(\mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{y},\sigma^2}\Big[-\int_0^{\mathcal{T}} \left[\boldsymbol{\theta}-\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{\Gamma})\right]^T\boldsymbol{C}(t)\boldsymbol{C}^T(t)\left[\boldsymbol{\theta}-\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{\Gamma})\right]\Big]\,dt\right) \\ &= \mathbb{E}_{\boldsymbol{y},\sigma^2}\left(-\int_0^{\mathcal{T}} \mathrm{tr}\Big\{\sigma^2\,\boldsymbol{C}(t)\boldsymbol{C}^T(t)\big(\boldsymbol{Z}^T\boldsymbol{Z}+\boldsymbol{V}^{-1}\big)^{-1}\Big\}\,dt\right) \\ &= \mathbb{E}_{\boldsymbol{y},\sigma^2}\left(-\sigma^2\,\mathrm{tr}\Big\{\int_0^{\mathcal{T}} \boldsymbol{C}(t)\boldsymbol{C}^T(t)\big(\boldsymbol{Z}^T\boldsymbol{Z}+\boldsymbol{V}^{-1}\big)^{-1}\,dt\Big\}\right) \\ &= -\mathbb{E}_{\sigma^2}\left(\sigma^2\,\mathrm{tr}\Big\{\int_0^{\mathcal{T}} \boldsymbol{C}(t)\boldsymbol{C}^T(t)\big(\boldsymbol{Z}^T\boldsymbol{Z}+\boldsymbol{V}^{-1}\big)^{-1}\,dt\Big\}\right) \\ &\propto -\mathrm{tr}\Big\{\big(\boldsymbol{Z}^T\boldsymbol{Z}+\boldsymbol{V}^{-1}\big)^{-1}\int_0^{\mathcal{T}} \boldsymbol{C}(t)\boldsymbol{C}^T(t)\,dt\Big\} \\ &= -\mathrm{tr}\Big\{\big(\boldsymbol{Z}^T\boldsymbol{Z}+\boldsymbol{V}^{-1}\big)^{-1}\boldsymbol{A}\Big\}, \end{aligned} \tag{5.46}$$

with $\boldsymbol{A} = \int_0^{\mathcal{T}} \boldsymbol{C}(t)\boldsymbol{C}^T(t)\,dt$. A design $\boldsymbol{\Gamma}^*$ is optimal if it maximises the expected utility of the Bayesian design criterion tailored on profile factors. The criterion in (5.46) is equivalent to the L-optimality design criterion; see Atkinson et al. (2007). The A-optimality is a special case of L-optimality if $\boldsymbol{A}$ is the identity matrix.

5.8 Constraints on the number of basis functions and runs

The aim in this section is to use properties of the rank of a matrix to derive constraints in the settings of the experimental designs for the FLM. Often, the objective function of optimality criteria requires the information matrix to be invertible, i.e., non-singular. For instance, this is the case for the objective function for A-optimality; see the objective function in (5.29).

In linear algebra, a matrix is said to be invertible if it is of full rank. Full rank is achieved when the number of linearly independent columns of the matrix is at the maximum, meaning equal to the number of columns. A matrix that does not achieve full rank is called rank-deficient. For the definition of rank and full rank; see Seber (2008, Chapter 3).

To avoid singularity of the information matrix, the column vectors of $\boldsymbol{Z}^T\boldsymbol{Z}$ need to be linearly independent. A set of vectors is linearly dependent if any linear combination of these vectors returns the zero vector. Suppose having a set of vectors $\bar{v}_1, ..., \bar{v}_{\bar{n}}$. The set of vectors is linearly dependent if for some coefficients $\bar{c}_1, ..., \bar{c}_{\bar{n}}$ with at least one of them being non-zero, the following holds,

$$\sum_{\bar{i}=1}^{\bar{n}} \bar{c}_{\bar{i}} \bar{v}_{\bar{i}} = 0.$$

Three important properties of rank are discussed in Magnus and Neudecker (2019, page 9). For matrices A_r and B_r of sizes $a_{r1} \times a_{r2}$ and $b_{r1} \times b_{r2}$ respectively, the properties are given by:

$$\operatorname{rank}(A_r) \leq \min(a_{r1}, a_{r2}), \tag{5.47a}$$

$$\operatorname{rank}(A_r) = \operatorname{rank}(A_r^T A_r), \tag{5.47b}$$

$$\operatorname{rank}(A_r B_r) \leq \min(\operatorname{rank}(A_r), \operatorname{rank}(B_r)). \tag{5.47c}$$

By the property in (5.47b), it is clear that rank deficiency in $\boldsymbol{Z}$ is rank deficiency in $\boldsymbol{Z}^T\boldsymbol{Z}$. Hence, no linear dependencies in $\boldsymbol{Z}$ means that there are no linear dependencies in $\boldsymbol{Z}^T\boldsymbol{Z}$. The model matrix $\boldsymbol{Z}$ has dimensions $n \times \sum_{q=1}^{Q} n_{\beta,q}$. By the property in (5.47a),

$$\operatorname{rank}(\boldsymbol{Z}) \leq \min\left(n, \sum_{q=1}^{Q} n_{\beta,q}\right),$$

meaning that the number of rows should be greater or at least equal to the number of columns. Thus, the first constraint on the settings of the functional model is,

$$n \geq \sum_{q=1}^{Q} n_{\beta,q},$$

with n the number of runs and $\sum_{q=1}^{Q} n_{\beta,q}$ the total number of basis functions from all of the basis expansions of the functional parameters. However, increasing the number of runs n by replicating design functions already in the design, will not increase the rank of $\boldsymbol{Z}$.

Additionally, the model matrix $\boldsymbol{Z}$ is partitioned in Q column blocks, i.e., one column block for each term in the FLM; see (5.12). If the q^{th} column block represents a main effect of a profile factor, say the j^{th} profile factor, then the q^{th} partition of $\boldsymbol{Z}$ is defined as in (5.14), $j = 1, 2, \ldots, J, q = 1, 2, \ldots, Q$. In this case, the column block partition is the product of the $n \times n_{x_j}$ coefficient matrix $\boldsymbol{\Gamma}_j$ of the j^{th} profile factor and the $n_{x_j} \times n_{\beta_j}$ matrix resulting from the product of the basis functions of the profile factor and the functional parameter of the profile factor. Using the latter information and the property in (5.47c) and (5.47a), the rank of the q^{th} column block is,

$$\text{rank}(\boldsymbol{Z}_{\bullet q}) \leq \min(\min(n, n_{x_j}), \min(n_{x_j}, n_{\beta,q})), \quad j = 1, 2, \ldots, J, q = 1, 2, \ldots, Q.$$

Moreover, if the q^{th} column block represents a K-way interaction of profile factors or a K-order polynomial of a profile factor, then the q^{th} partition of $\boldsymbol{Z}$ is defined as in (5.23). In this case, the column block partition is the product of the $n \times \prod_{k=1}^{K} n_{x,j_k}$ coefficient matrix $\boldsymbol{\Gamma}_{j_1 j_2 \cdots j_K}$ from (5.23) and the $\prod_{k=1}^{K} n_{x,j_k} \times n_{\beta,q}$ matrix resulting from the product of the basis functions of the profile factors involved $\boldsymbol{c}_{j_1 j_2 \cdots j_K}(t)$ from (5.23) and the basis functions of the functional parameter. Under this scenario, the rank of the q^{th} column block is,

$$\text{rank}(\boldsymbol{Z}_{\bullet q}) \leq \min\Big(\min\Big(n, \prod_{k=1}^{K} n_{x,j_k}\Big), \min\Big(\prod_{k=1}^{K} n_{x,j_k}, n_{\beta,q}\Big)\Big),$$
$$j = 1, 2, \ldots, J, \ q = 1, 2, \ldots, Q,$$

with $j_1, j_2, \ldots, j_K$, the K profile factors involved in the interaction or the K-order polynomial if $j_1 = j_2 = \cdots = j_K$. Thus, the second constraint on the settings of the functional model is,

$$\prod_{k=1}^{K} n_{x,j_k} \geq n_{\beta,q}, \quad q = 1, 2, \ldots, Q.$$

In summary, if the objective function requires the information matrix to be invertible, the settings of the functional model must satisfy the following two constraints:

$$\begin{aligned} n &\geq \sum_{q=1}^{Q} n_{\beta,q}, \\ \prod_{k=1}^{K} n_{x,j_k} &\geq n_{\beta,q}, \\ q &= 1, 2, \ldots, Q. \end{aligned} \tag{5.48}$$

The main effects constraint is a special case of the interactions constraint, with $K = 1$.

Finally, the information matrix can still be singular even if the constraints are satisfied. However, failure to satisfy the constraints implies certain singularity of the information matrix.

5.9 Integrate the product of B-spline basis functions

In this thesis, the functions of the profile factors are assumed to be represented by B-spline basis expansions. Similarly, the functional parameters are assumed to be represented by power series basis or B-spline basis expansions. The FLM methodology developed earlier in this chapter involves integrals with respect to time that include products of basis functions. Specifically, for main effect terms the integral involves the product of the basis functions of a single profile factor and the basis functions of a functional parameter. For interaction and polynomial terms, the integral involves the product of basis functions of every profile factor involved in the interaction and the basis functions of the functional parameter. In this section, the aim is to describe a procedure for solving the integrals involved in the FLM methodology using basis expansions.

The latter description means that the integrals required to calculate under the FLM set up are: integrals involving products of BS basis functions or integrals involving products of BS basis functions and power series of time up to a certain degree. As the order of interactions or polynomials increases, then the number of BS basis functions increases as well. However, the number of BS basis functions involved in the integrals does not affect the procedure to analytically compute the integrals. This is because a product of B-splines can be expressed as a linear combination of B-splines; see Vermeulen et al. (1992). Also, later in this section proof is given that integrals involving both BS basis functions and power series basis functions are a special case of integrals with BS basis functions.

Without loss of generality, in this section a double BS integral is considered to illustrate the procedure and avoid over complication. Repetition of the technique to be presented leads to the derivation of integrals that involve the products of more BS basis functions. The integral to solve takes the form of the integral in (5.14) and it depends on partitions that take the form of integrals of the products of basis functions such that:

$$\begin{aligned}\int_0^{\mathcal{T}} \boldsymbol{c}(t)\ \boldsymbol{b}^T(t)\ dt &= \int_0^{\mathcal{T}} B_{\alpha_c,d_c}(t)\ B_{\alpha_b,d_b}(t)\ dt \\ &= \int_0^{\mathcal{T}} t^0\ B_{\alpha_c,d_c}(t)\ B_{\alpha_b,d_b}(t)\ dt \\ &= I(0, d_c, \alpha_c, d_b, \alpha_b),\end{aligned} \tag{5.49}$$

with $B^{d_c}_{\alpha_c}(t)$ and $B^{d_b}_{\alpha_b}(t)$ be BS basis of degree d_c and d_b and basis function number α_c and α_b, respectively. The function I is a function representing the settings of degrees, basis function position numbering and power on t. The importance of the second and third equality in (5.49) will make more sense after the expansion of the two basis inside the integral using the recursion formula from (4.13).

Without any loss of generality, suppose that the degree of BS for the functional parameter is lower than the degree of BS for the profile factor, i.e., $d_b < d_c$. Start expanding the BS with the lowest degree, in this case the BS for the functional parameter, using the recursion formula, yields functions of the same form with different settings for degree, positioning and power of t,

$$\begin{aligned}I(0, d_c, \alpha_c, d_b, \alpha_b) &= \int_0^{\mathcal{T}} t^0\ B_{\alpha_c,d_c}(t)\ B_{\alpha_b,d_b}(t)\ dt \\ &= \frac{1}{\lambda^*_{\alpha_b+d_b} - \lambda^*_{\alpha_b}} \int_0^{\mathcal{T}} t^1\ B_{\alpha_c,d_c}(t)\ B_{\alpha_b,d_b-1}(t)\ dt \\ &\quad - \frac{\lambda^*_{\alpha_b}}{\lambda^*_{\alpha_b+d_b} - \lambda^*_{\alpha_b}} \int_0^{\mathcal{T}} t^0\ B_{\alpha_c,d_c}(t)\ B_{\alpha_b,d_b-1}(t)\ dt \\ &\quad + \frac{\lambda^*_{\alpha_b+d_b+1}}{\lambda^*_{\alpha_b+d_b+1} - \lambda^*_{\alpha_b+1}} \int_0^{\mathcal{T}} t^0\ B_{\alpha_c,d_c}(t)\ B_{\alpha_b+1,d_b-1}(t)\ dt \\ &\quad - \frac{1}{\lambda^*_{\alpha_b+d_b+1} - \lambda^*_{\alpha_b+1}} \int_0^{\mathcal{T}} t^1\ B_{\alpha_c,d_c}(t)\ B_{\alpha_b+1,d_b-1}(t)\ dt \\ &= I(1, d_c, \alpha_c, d_b - 1, \alpha_b) - I(0, d_c, \alpha_c, d_b - 1, \alpha_b) \\ &\quad + I(0, d_c, \alpha_c, d_b - 1, \alpha_b + 1) - I(1, d_c, \alpha_c, d_b - 1, \alpha_b + 1),\end{aligned}$$

where $\boldsymbol{\lambda}^*$ is the extended knot vector; see Section 4.4. From the above expansion, it is clear to notice that interest is to solve integrals of the form,

$$\int_0^{\mathcal{T}} t^r \; B_{\alpha_c,d_c}(t) \; B_{\alpha_b,d_b}(t) \; dt = I(r, d_c, \alpha_c, d_b, \alpha_b)$$

which yield integrals of the same form with different input settings.

Following the same procedure expanding the BS basis with the lowest degree, will eventually get the BS basis down to degree zero for which there exists a formal definition; defined in (4.12). This means that, every BS basis is expanded until its degree reaches zero. After that, the integral involves one less BS basis than before, but the integral bounds are revised such that,

$$\begin{aligned} I(r, d_c, \alpha_c, 0, \alpha_b) &= \int_0^1 t^r \; B_{\alpha_c,d_c}(t) \; B_{\alpha_b,0}(t) \; dt \\ &= \int_{\lambda^*_{\alpha_b}}^{\lambda^*_{\alpha_b+1}} t^r \; B_{\alpha_c,d_c}(t) \; dt. \end{aligned} \tag{5.50}$$

After that, the same procedure is followed for the remaining BS basis that are involved in the integral. The procedure is repeated until all BS basis are down to degree zero. For the example with two BS basis, the result is given by,

$$\begin{aligned} I(r, 0, \alpha_c, 0, \alpha_b) &= \int_{\lambda^*_{\alpha_b}}^{\lambda^*_{\alpha_b+1}} t^r \; B_{\alpha_c,0}(t) \; dt \\ &= \int_{t_l}^{t_u} t^r \; dt \\ &= \begin{cases} \frac{t_u^{r+1} - t_l^{r+1}}{r+1} & \text{if } t_l < t_u \\ 0 & \text{if } t_l \geq t_u, \end{cases} \end{aligned} \tag{5.51}$$

where,

$$\begin{aligned} t_u &= \min\left(\lambda^*_{\alpha_b+1}, \lambda^*_{\alpha_c+1}\right) \\ t_l &= \max\left(\lambda^*_{\alpha_b}, \lambda^*_{\alpha_c}\right). \end{aligned}$$

The analytic derivation of integrals involving the product of BS basis covers also the special case of BS basis and power series basis. The integral to solve takes the form of (5.50), but integrating on the whole time line,

$$\int_0^{\mathcal{T}} t^r \; B_{\alpha_c,d_c}(t) \; dt.$$

5.10 A method to compare sets of functions

The optimal design is undoubtedly affected by the settings passed to the model. To be precise, the basis choice for the profile factors and the functional parameters as well as the number of basis function affects the final functions. The number of basis functions depends on the choice of the degree and the choice of the knots; so degree and knots also affect the final functions of the profile factors. The more complicated the basis is in terms of degree and choice of knots, the more complicated the design and hence the optimal functions are expected to be.

A comparison between different sets of functions of a profile factor with significant differences in the assumed settings, are usually recognised visually. Especially for step functions, the number of changes in the step functions are easily interpretable and conclude to the complexity of functions. However, a complexity comparison between different sets of functions with almost identical assumed settings is not always straightforward. Especially for smooth continuous functions, a comparison is more difficult and the difficulty increases as the degree of the polynomials increases. Having said that, a mathematically proper and valid valuation of the complexity of the functions is essential. For example, when comparing the optimal functions of profile factors resulting from models with and without interactions, values of the complexity of the functions can lead to better understanding.

The complexity of the functions in the design is measured using the P-spline penalty; see Eilers and Marx (1996). P-splines apply a difference penalty directly to the coefficients from the expansion of the basis. A spline function of degree d is expressed as a linear combination of basis functions along with vector valued constants called control points, see (4.4). The B-spline basis expansion for the j^{th} profile factor in the i^{th} run of the experiment is given by,

$$x_{ij}(t) = \sum_{l=1}^{n_{x,j}} \gamma_{ijl} B_{l,d}(t), \tag{5.52}$$

for d representing the required degree of the B-spline. Thus, P-splines apply a difference penalty directly to the coefficients γ_{ijl} from (5.52), penalising the wiggly functions. The penalty for the i^{th} run and the j^{th} profile factor is defined as the squared difference between adjacent coefficients,

$$P_{ij} = \mathbf{\Gamma}_{j_i}^T \, \boldsymbol{D}^T \boldsymbol{D} \, \mathbf{\Gamma}_{j_i}$$

where $\mathbf{\Gamma}_{j_i}$ is the i^{th} row of the coefficient matrix from the B-spline basis expansion of the j^{th} profile factor. The $n_{x,j} - 2 \times n_{x,j}$ matrix $\boldsymbol{D}$ is a matrix controlling the correct choice of coefficient differences; see Wood (2017*a*, Chapter 5),

$$\boldsymbol{D} = \begin{pmatrix} 1 & -2 & 1 & 0 & . & . & . \\ 0 & 1 & -2 & 1 & 0 & . & . \\ 0 & 0 & 1 & -2 & 1 & 0 & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \end{pmatrix}.$$

In this set up, the penalty is derived and applied to the coefficients which are the design matrix and does not measure the wiggliness of the basis functions themselves.

Penalised optimal designs are out of the scope of this thesis. However, the P-spline penalty value is a practically efficient method to value the complexity and wiggliness of functions of profile factors. For example, a high penalty value will indicate a more complicated function, and a low penalty value a less complicated function. Hence, the penalty value of the functions can be used to compare the complexity of different sets of functions, which is exactly how the material in this section is used in the thesis.

Finally, derivative-based penalties are also used in the Statistics literature to penalise wiggliness. However, derivative-based penalties require the degree of the derivative to be less or equal the degree of the B-spline basis, otherwise the penalty will make no sense; see Wood (2017*b*). This is clear from the definition of B-spline derivatives in (4.15). If the degree of derivative is higher than the degree of the B-spline then the result is always zero.

5.11 Conclusion

In this chapter, linear models depending on profile factors have been tackled. Attention has been given on the design of experiments for functional linear models depending on profile factors and/or functions of profile factors. A new approach using basis function expansions on the profile factors and the functional parameters has been developed. In fact, a different basis choice for the profile factors and the functional parameters is allowed. For the profile factors, the basis that is used ensures that the bounds of the coefficients from the basis expansion are equivalent bounds to the functions of the profile factors. No additional constraints are required. The methodology is applied to models with combinations of profile and static factors. Further, this chapter showed that such models are extensions of the traditional linear models, thus, common statistical techniques can be used. The model matrix of the FLM consists of column block partitions, for every term in the model. To derive the column blocks, the linear predictor of each term is integrated with respect to time. The more complicated the term, interaction for instance, the more complicated is to derive the column block.

A frequentist approach and a Bayesian approach for design of experiments of FLM have been described, and the connection between them has been discussed. Moreover, a new criterion tailored to functional linear models has been introduced, that allows the choice of the functions of interest.

The methodology developed involves integrating products of BS basis with respect to time. The product of BS basis is also a BS basis, thus, the integral of the product of BS basis is analytically tractable. The procedure to derive analytic solutions of such integrals have been described in detail. Common objective functions, the A-optimality for instance, depend on the inverse of the information matrix. To avoid linear dependencies in the information matrix, certain constraints have been introduced using properties of the rank. The constraints depend on the settings of the experiment, of the profile factors and of the functional parameters. The information matrix can still be non-invertible, even if the constraints are met. However, failure to satisfy the constraints, causes linear dependencies, and the information matrix is certainly non-invertible.

Finally, to allow proper comparisons on the complexity of competing set of functions of profile factors, a method for valuing the complexity of functions using squared differences is described. A comparison between optimal set of functions could be straightforward if the settings of the factors and the parameters are significantly different. However, a comparison of the complexity between similar set of functions can be hard, and that is when valuing the complexity is useful.

Chapter 6

Optimal designs for functional linear models

The focus of this chapter is to find optimal experimental designs for functional linear models using the methodology developed in Chapter 5. Several models are studied, including: models involving one or more profile factors, models with combinations of profile and scalar factors, and models with interactions of the profile factors. Optimal experimental designs are identified using the frequentist approach, the Bayesian approach and the Bayesian approach tailored to profile factors. The optimal designs derived by the latter approaches are compared amongst them to identify connections between the approaches.

The settings of the models are varied, to alter the complexity of the models and evaluate the sensitivity of the designs to changes. Attention is given on the performance of the A- and D- optimal designs after changing the basis of the functional parameters and/or after increasing the complexity of the basis of the profile factors in terms of the choice of the degree and the knots.

6.1 FLM involving one profile factor and step function basis

In this section, a simple FLM involving the main effect of a single profile factor and the intercept is considered. Thus, the functional of the profile factors $\boldsymbol{f}$ is,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \begin{pmatrix} 1 & x_{i1}(t) \end{pmatrix}, \tag{6.1}$$

with $x_{i1}(t)$ the single profile factor at the i^{th} run of the experiment. For a single profile factor, the FLM model from equation (5.10) is simplified to a FLM of the form,

$$y_i = \beta_1 + \int_0^{\mathcal{T}} \beta_2(t) x_{i1}(t)\, dt + \epsilon_i, \quad i = 1, 2, \ldots, n,\ t \in [0, 1],\ -1 \leq x_{i1}(t) \leq 1. \tag{6.2}$$

The aim is to identify A- and D- optimal designs and to perform a sensitivity study to investigate how the optimal designs are affected in changes to the settings of the experiment.

It is assumed that control of the profile factor is represented via a step function, thus, the function of the profile factor is expanded as a BS basis of degree $d = 0$; see Section 4.4. The design problem is reduced to optimise a single coefficients matrix $\mathbf{\Gamma}_1$ with dimensions $n \times n_{x,1}$. The degree of the BS basis is fixed to $d = 0$, so the complexity of the functions depend on the choice of knots, and on the choice of basis for the functional parameter. Knots are assumed to be equally spaced over the time interval $[0, 1]$. The total number of basis functions is given by $d + k + 1$ with d and k being the degree and total number of knots; see (4.3). Since the degree of the BS basis is zero, then the number of basis functions is the number of knots plus one.

For the functional parameters, the power series system for linear and quadratic basis is used. The linear and quadratic basis expansions of $\beta_2(t)$ are given by,

$$\begin{aligned}\beta_2(t) &= \theta_{21} + \theta_{22} t,\\ \beta_2(t) &= \theta_{21} + \theta_{22} t + \theta_{23} t^2,\end{aligned} \tag{6.3}$$

respectively. For the linear basis, there are 2 basis functions for the functional parameter, i.e., $n_{\beta,2} = 2$, the basis function vector is $\boldsymbol{b}_2(t) = (1, t)$, and the vector of unknown coefficients is $\boldsymbol{\theta}_2 = (\theta_{21}, \theta_{22})^T$. For the quadratic basis, there are 3 basis functions for the functional parameter, i.e., $n_{\beta,2} = 3$, the basis function vector is $\boldsymbol{b}_2(t) = (1, t, t^2)$, and the vector of unknown coefficients is $\boldsymbol{\theta}_2 = (\theta_{21}, \theta_{22}, \theta_{23})^T$. The linear and the quadratic basis functions for the functional parameter is shown in Figure 6.1.

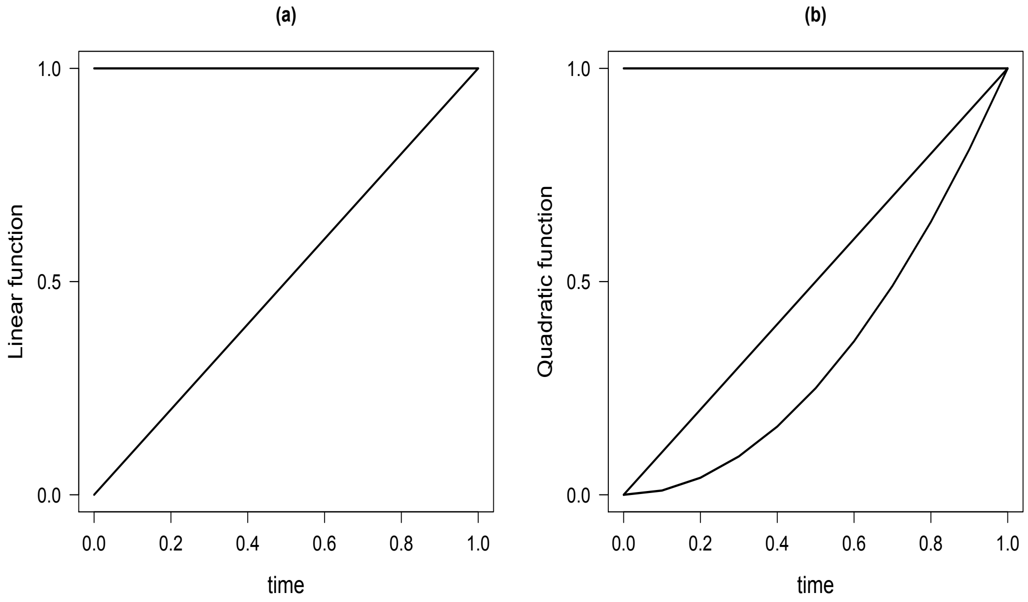


FIGURE 6.1: (a) Linear and (b) Quadratic basis functions for $\beta_2(t)$.

The A-optimality objective functions require the information matrix to be invertible; see (5.29). This implies a constraint in the number of basis functions for the function of the profile factor such that: if the basis for the parameters is linear, then it is required that $n_{x,1} \geq 2$, and if the basis is quadratic, then it is required that $n_{x,1} \geq 3$; see Section 5.8. For instance, if the basis for the parameters is linear and $n_{x,1} = 1$, there are no interior knots, i.e., only boundary knots. This causes linear dependency in the $\boldsymbol{Z}^T\boldsymbol{Z}$, thus, non-invertibility. Similarly, the same holds for the quadratic basis and $n_{x,1} = 1, 2$.

The number of runs considered is $n \in \{4, 8, 12\}$. Moreover, the number of basis functions considered is $n_{x,1} \in \{2, 3, 4, 8, 16, 100\}$; with the choice of $n_{x,1} = 2$ excluded for the quadratic basis. Knots are equally spaced. Having equally spaced knots, means that the knot vector of a certain choice of $n_{x,1}$ contains all knot vectors with the numbers of knots a factor of $n_{x,1}$. For instance, if $n_{x,1} = 16$, the knot vector includes every knot vector of size $n_{x,1} = 2, 4, 8$, since they are factors of 16. For each of the aforementioned number of runs and number of basis functions, the coordinate exchange algorithm is used to find A- and D- optimal designs for 1000 random starts.

Firstly, the linear basis for the functional parameter is considered, with the results available on Table 6.1 and Table 6.2. Based on the sensitivity study, as the number of runs increases from 4 to 8 and from 8 to 12, the objective values drop. In general, increasing the number of runs gives identical or similar patterns, in more repetitions. Interestingly, one could notice that regardless the number of runs, the objective value for $n_{x,1} = 3$ is higher than for $n_{x,1} = 2, 4$. This has to do with the choice of nested models, i.e., the knot vector for a model with $n_{x,1} = 2$, is nested in the knot vector of a model with $n_{x,1} = 4$. The design identified for $n_{x,1} = 3$ is sub-optimal, and it is the only model that does not include $t = 0.5$ in the knot vector. The optimal designs found for a single profile factor achieve at most two changes in the step function. However, most of the designs achieve a single change from -1 to 1 and vice-versa, or no change, i.e., constant function in -1 or 1. For A-optimal designs with at most two changes; see Figure 6.2 and Figure 6.3, and for D-optimal designs with at most one change; see Figure 6.4 and Figure 6.5.

Furthermore, as the number of basis function increases the designs perform better with respect to A-optimality. However, the drop in the objective values of the optimal designs as $n_{x,1}$ increases becomes insignificant; see Table 6.1. This is an indication that a large number of basis functions for the profile factor is not needed. For D-optimality this is even more clear, as the objective values does not change regardless the increase in the number of basis functions, with only exception the choice of $n_{x,1} = 3$; see Table 6.2. This is because, for linear parameters, the information matrix does not depend on $n_{x,1}$, and the runs of the design matrix are either constant or change once at $t = 0.5$; see Appendix D. The minimum objective value is achieved by the $n_{x,1} = 100$ designs for every choice of n. However, as the number of basis functions increases, the functions of the profile factor become more complicated. The A- and D- efficiency values indicate how well each optimal design with lower $n_{x,1}$ performs compared to the best found. For instance, the A- and D- efficiencies of a design $\mathbf{\Gamma}$ with respect to an optimal design $\mathbf{\Gamma}^*$ are defined as,

$$\text{A-eff} = \frac{\Psi_A(\mathbf{\Gamma})}{\Psi_A(\mathbf{\Gamma}^*)},$$

and,

$$\text{D-eff} = \frac{\Psi_D(\mathbf{\Gamma})}{\Psi_D(\mathbf{\Gamma}^*)}.$$

The objective values found by minimising the A- and D-optimality, with linear parameter basis, for $n \in \{4, 8, 12\}$ runs and $n_{x,1} \in \{2, 3, 4, 8, 16, 100\}$ basis functions as well as the A- and D-efficiencies can be found in Table 6.1 and Table 6.2 respectively.

	$n = 4$		$n = 8$		$n = 12$	
$n_{x,1}$	A-opt	A-eff	A-opt	A-eff	A-opt	A-eff
2	8.750	0.961	3.958	0.981	2.583	0.972
3	8.828	0.952	4.287	0.906	2.778	0.904
4	8.750	0.961	3.903	0.995	2.570	0.977
8	8.493	0.990	3.902	0.995	2.539	0.989
16	8.427	0.997	3.887	0.999	2.520	0.997
100	8.404	1.000	3.882	1.000	2.512	1.000

TABLE 6.1: A-optimality values and A-efficiency values with $n \in \{4, 8, 12\}$ for the linear basis for $\beta_2(t)$ for the FLM.

	$n = 4$		$n = 8$		$n = 12$	
$n_{x,1}$	D-opt	D-eff	D-opt	D-eff	D-opt	D-eff
2	1.000	1.000	0.500	1.000	0.333	1.000
3	1.062	0.942	0.522	0.958	0.348	0.957
4	1.000	1.000	0.500	1.000	0.333	1.000
8	1.000	1.000	0.500	1.000	0.333	1.000
16	1.000	1.000	0.500	1.000	0.333	1.000
100	1.000	1.000	0.500	1.000	0.333	1.000

TABLE 6.2: D-optimality values and D-efficiency values with $n \in \{4, 8, 12\}$ for the linear basis for $\beta_2(t)$ for the FLM.

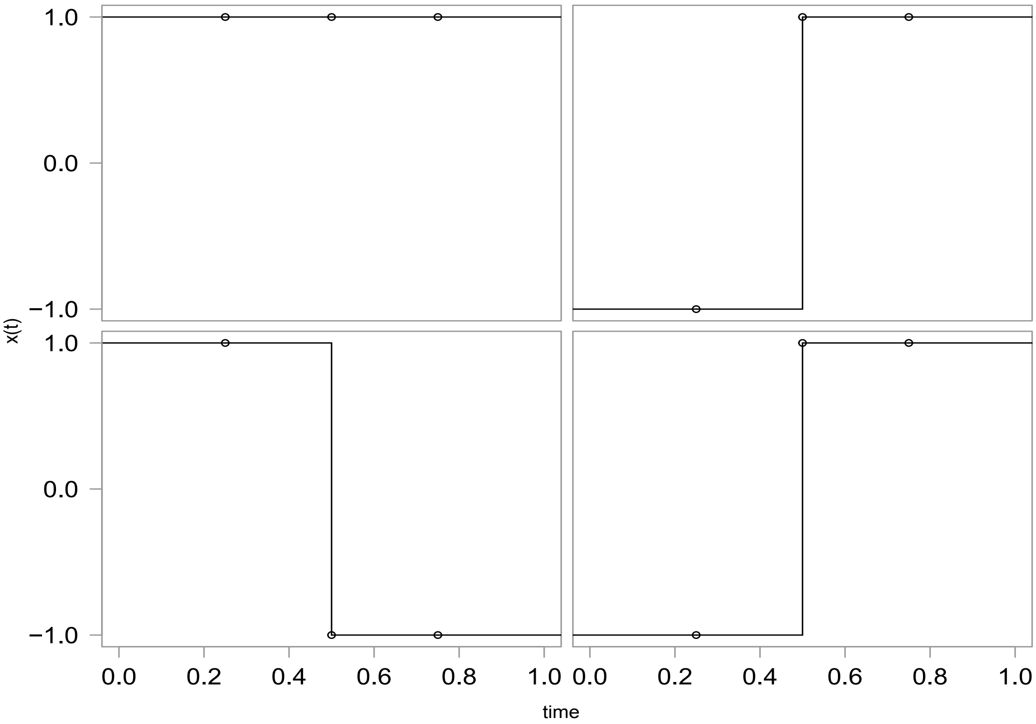


FIGURE 6.2: Four run A-optimal design for $n_{x,1} = 4$, linear basis for $\beta_2(t)$ and step function basis for $x_{\cdot 1}(t)$ for the FLM.

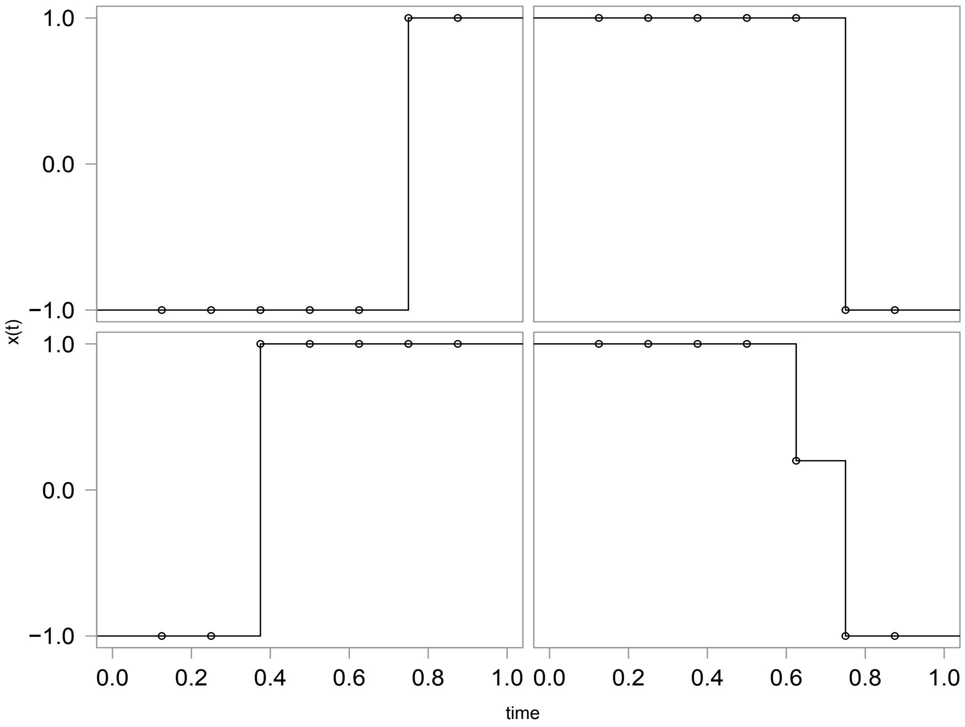


FIGURE 6.3: Four run A-optimal design for $n_{x,1} = 8$, linear basis for $\beta_2(t)$ and step function basis for $x_{\cdot 1}(t)$ for the FLM.

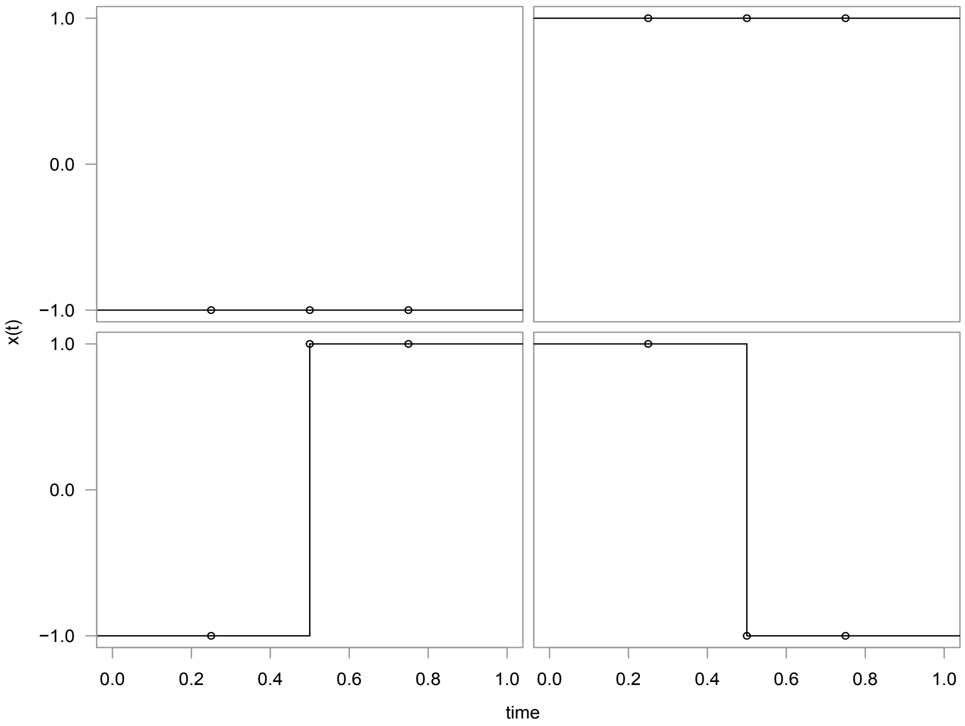


FIGURE 6.4: Four run D-optimal design for $n_{x,1} = 4$, linear basis for $\beta_2(t)$ and step function basis for $x_{\cdot 1}(t)$ for the FLM.

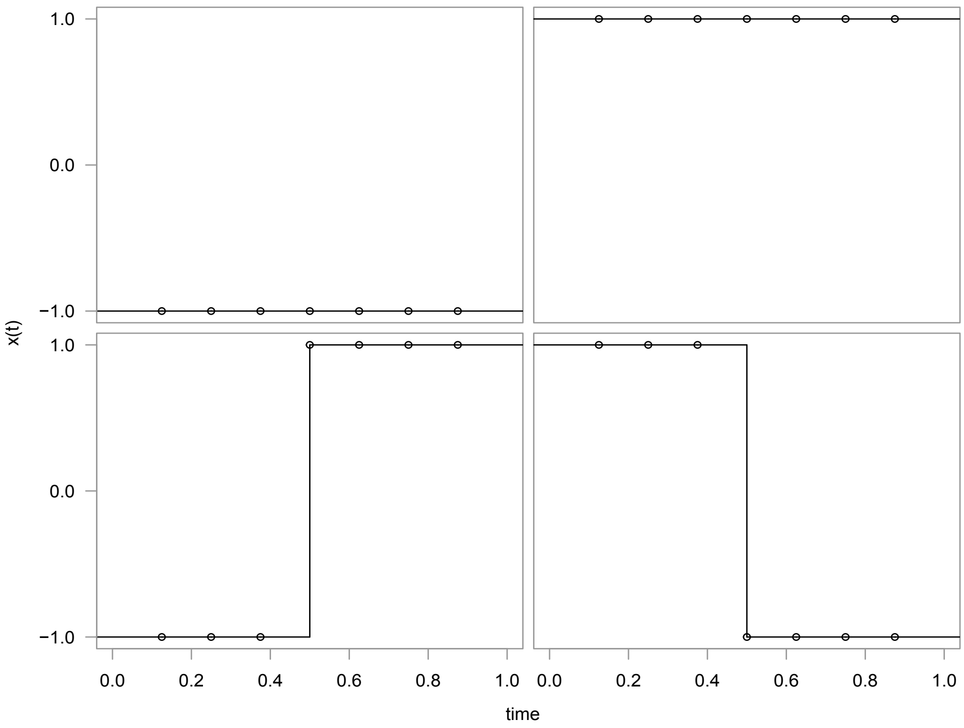


FIGURE 6.5: Four run D-optimal design for $n_{x,1} = 8$, linear basis for $\beta_2(t)$ and step function basis for $x_{\cdot 1}(t)$ for the FLM.

Secondly, the quadratic basis for the functional parameter is considered, with the results available on Table 6.3 and Table 6.4. Increasing the number of runs results in lower objective values and the designs have similar patterns. The step functions achieve at most three changes, but most of the functions achieve two changes or even a single change. Meaning that, most of the functions found move from -1 to 1 and then back to -1 or the other way round. Three changes in the step function occur mostly for large values of $n_{x,1}$, with one of the three changes being very small. To visualise the latter findings; see Figure 6.6 and Figure 6.7 for A-optimal designs with at most three changes, and Figure 6.8 and Figure 6.9 for D-optimal designs with at most two changes.

The quadratic basis for the parameters is more sensitive to changes in the number of basis functions, compared to the linear case. For instance, increasing the number of basis functions causes a more significant change in the performance of the designs; see Table 6.3 and Table 6.4. For the linear case, the efficiency values found are greater than 0.900 even for a small number of basis functions. Meaning that, the designs are at least 90% efficient. However, to achieve 90% efficiency for the quadratic case, a higher number of basis functions is needed. Thus, it makes sense to use a larger number of basis functions for the quadratic basis. On the other hand, a huge number of basis functions is still not needed. For the quadratic basis it is not possible to compare whether the objective value for $n_{x,1} = 3$ is higher than $n_{x,1} = 2, 4$, because a choice of two basis functions violates the constraint for invertibility of the information matrix. However, it is clear to notice that between three and four basis functions there is a significant drop in the objective value; especially for A-optimality. Finally, the drop in the objective values of the A-optimal designs are more significant, compared to the D-optimal designs; see A- and D- efficiencies in Table 6.3 and Table 6.4, respectively.

	$n=4$		$n=8$		$n=12$	
$n_{x,1}$	A-opt	A-eff	A-opt	A-eff	A-opt	A-eff
3	386.408	0.535	189.766	0.510	126.409	0.499
4	246.869	0.838	103.553	0.934	67.735	0.931
8	218.479	0.947	99.109	0.976	65.217	0.966
16	208.843	0.991	97.408	0.993	63.610	0.991
100	206.884	1.000	96.709	1.000	63.028	1.000

TABLE 6.3: A-optimality values and A-efficiency values with $n \in \{4, 8, 12\}$ for the quadratic basis for $\beta_2(t)$ for the FLM.

	$n=4$		$n=8$		$n=12$	
$n_{x,1}$	D-opt	D-eff	D-opt	D-eff	D-opt	D-eff
3	4.733	0.951	2.386	0.948	1.591	0.943
4	4.619	0.983	2.292	0.987	1.523	0.985
8	4.583	0.991	2.277	0.993	1.515	0.990
16	4.542	0.999	2.264	0.999	1.504	0.997
100	4.540	1.000	2.262	1.000	1.500	1.000

TABLE 6.4: D-optimality values and D-efficiency values with $n \in \{4, 8, 12\}$ for the quadratic basis for $\beta_2(t)$ for the FLM.

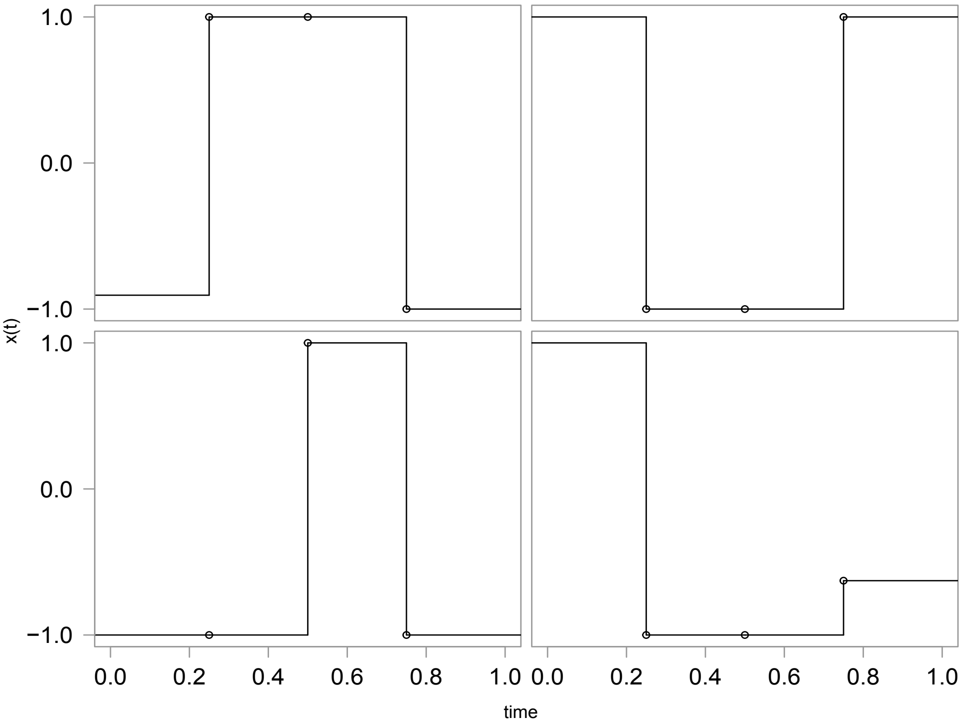


FIGURE 6.6: Four run A-optimal design for $n_{x,1} = 4$, quadratic basis for $\beta_2(t)$ and step function basis for $x_{\cdot 1}(t)$ for the FLM.

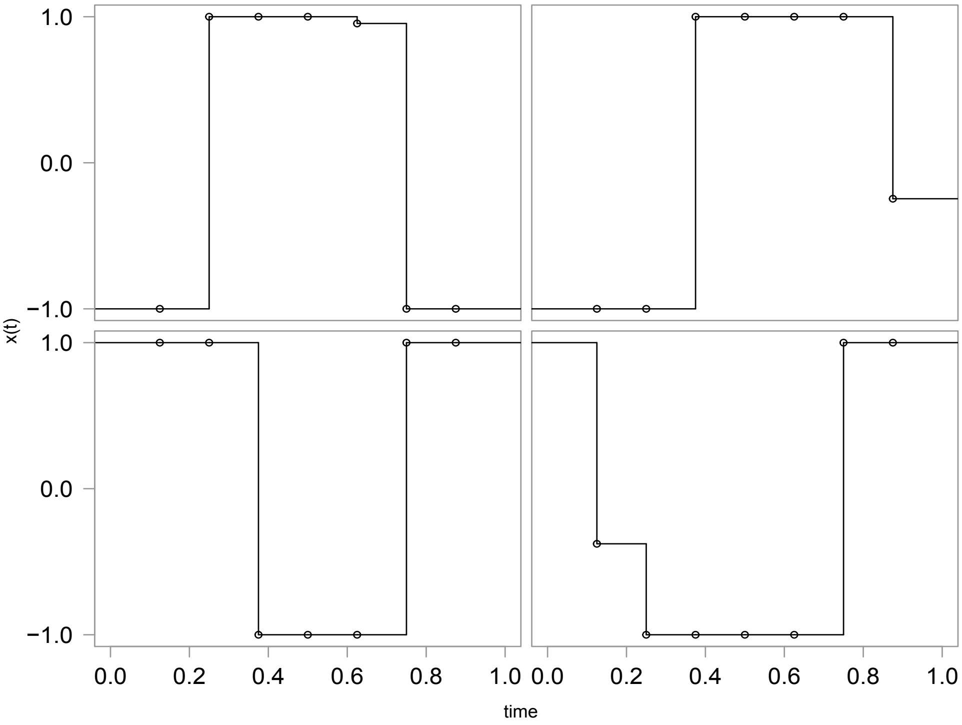


FIGURE 6.7: Four run A-optimal design for $n_{x,1} = 8$, quadratic basis for $\beta_2(t)$ and step function basis for $x_{\cdot 1}(t)$ for the FLM.

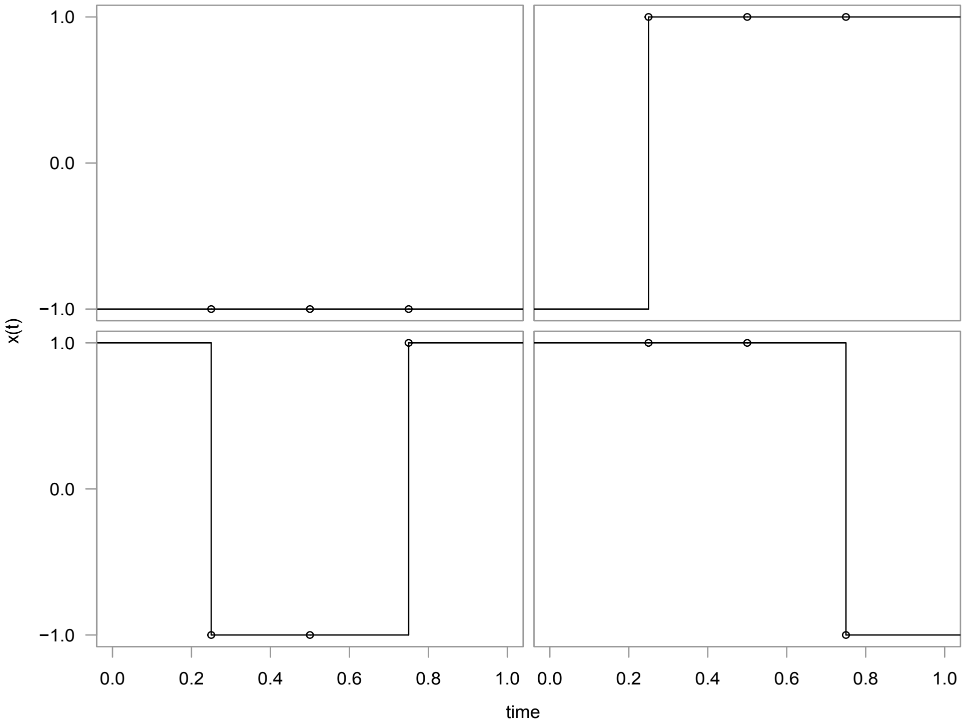


FIGURE 6.8: Four run D-optimal design for $n_{x,1} = 4$, quadratic basis for $\beta_2(t)$ and step function basis for $x_{\cdot 1}(t)$ for the FLM.

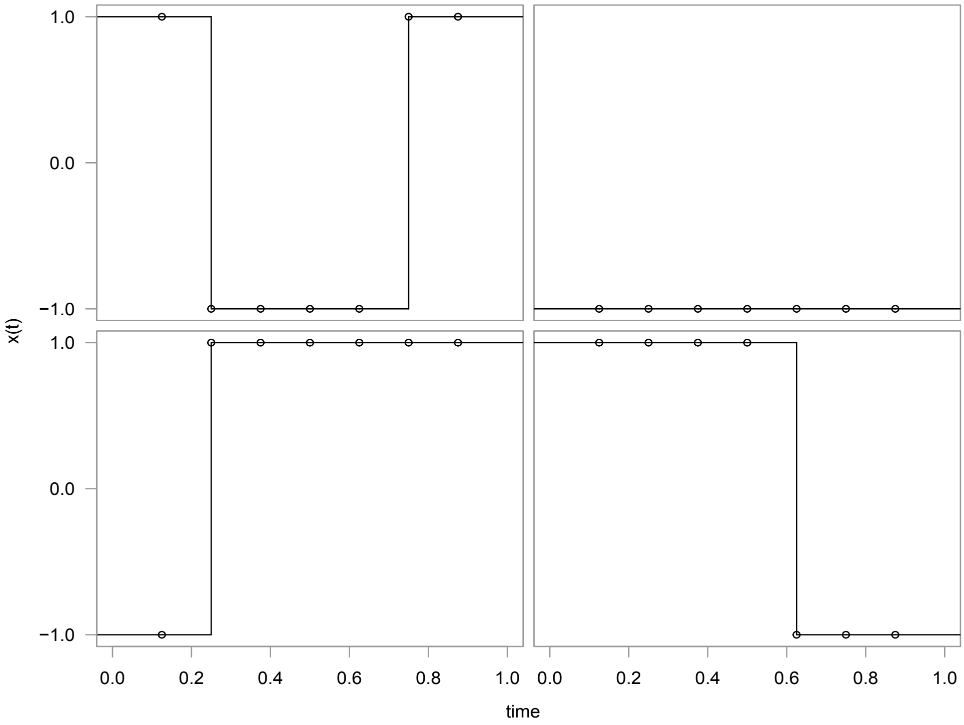


FIGURE 6.9: Four run D-optimal design for $n_{x,1} = 8$, quadratic basis for $\beta_2(t)$ and step function basis for $x_{\cdot 1}(t)$ for the FLM.

6.2 FLM involving one profile and three scalar factors

In this section, the aim is to identify A-optimal designs for a FLM that involves the intercept, a single profile factor, and three additional scalar factors. Scalar factors are a special case of profile factors, for which a single constant basis function is used; see Section 3.1. The case of main effects as well as the case for main and quadratic effects of the scalar factors is considered. For only main effects, the functional of the profile factors $\boldsymbol{f}$ is,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \begin{pmatrix} 1 & x_{i1}(t) & x_{i2} & x_{i3} & x_{i4} \end{pmatrix}, \tag{6.4}$$

with $x_{i1}(t)$ the single profile factor at the i^{th} run of the experiment, and x_{i2}, x_{i3} and x_{i4} the scalar factors at the i^{th} run of the experiment. For the addition of the quadratic effects, the functional of the profile factors $\boldsymbol{f}$ is,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \begin{pmatrix} 1 & x_{i1}(t) & x_{i2} & x_{i3} & x_{i4} & x_{i2}^2 & x_{i3}^2 & x_{i4}^2 \end{pmatrix}, \tag{6.5}$$

with x_{i2}^2, x_{i3}^2 and x_{i4}^2 the quadratic effects of the scalar factors at the i^{th} run of the experiment. Under this set up, the FLM takes the form,

$$y_i = \beta_1 + \int_0^{\mathcal{T}} \beta_2(t)x_{i1}(t)\,dt + \boldsymbol{g}^T(\boldsymbol{x}_i)\boldsymbol{v} + \epsilon_i, \quad i = 1, 2, \ldots, n,\ t \in [0, 1],\ -1 \leq \boldsymbol{x}_i(t) \leq 1, \tag{6.6}$$

$$\boldsymbol{g}^T(\boldsymbol{x}_i)\boldsymbol{v} = \begin{cases} \beta_3 x_{i2} + \beta_4 x_{i3} + \beta_5 x_{i4} & \text{Case 1} \\ \beta_3 x_{i2} + \beta_4 x_{i3} + \beta_5 x_{i4} + \beta_6 x_{i2}^2 + \beta_7 x_{i3}^2 + \beta_8 x_{i4}^2 & \text{Case 2.} \end{cases}$$

Control of the single profile factor is assumed to be represented by a step function basis, i.e, BS basis of degree $d = 0$. The functional parameter $\beta_2(t)$ is assumed to be represented by the linear basis. The number of basis functions for the profile factor is $n_{x,1} = 4$. The number of runs is $n = 12$. There are four unique functions of the profile factor; see Figure 6.10. The rest are repeated functions, with functions (a) and (b) repeated five times, and functions (c) and (d) once; see Table 6.5.

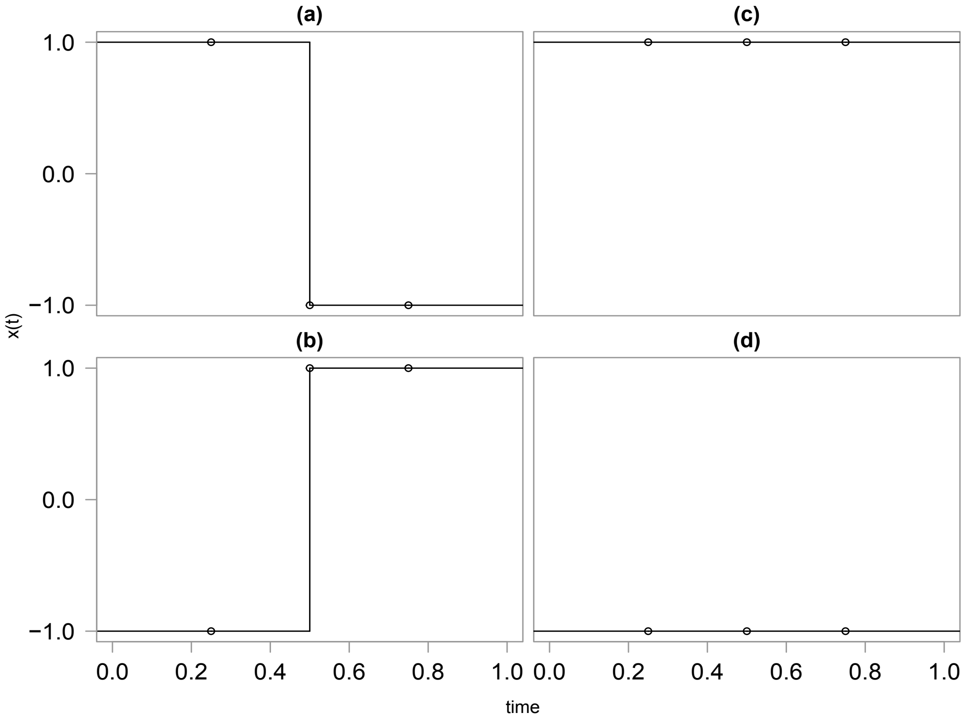


FIGURE 6.10: Four unique functions of the profile factor for the A-optimal design, with $n_{x,1} = 4$, linear basis for $\beta_2(t)$, and step function basis for $x_{\cdot 1}(t)$ for the FLM with one profile factor and three scalar factors with main effects.

What is interesting to investigate is how the scalar factors behave from main to main and quadratic effects. For the Case 1 model, i.e., the model with main effects, the optimal choices of the scalar factors are those at the boundaries. This is similar to the profile factor behaviour with linear basis for the parameters, where the function of the profile factor changes at most once. For the Case 2 model, i.e., the model with both main and quadratic effects, the optimal design for the scalar factors include boundary points and centre points in order to be able to estimate the curvature. This is similar to the profile factor behaviour with quadratic basis for the parameters, where the function of the profile factor changes at most twice. For the Case 1 model, the columns for the scalar factors are orthogonal, but the columns for the profile factor are not orthogonal. The A-optimal designs are given in Table 6.5.

Case 1

i	$x_{i1}(t)$	x_{i2}	x_{i3}	x_{i4}
1	(a)	-1.000	1.000	-1.000
2	(b)	-1.000	1.000	1.000
3	(a)	1.000	1.000	-1.000
4	(a)	-1.000	-1.000	1.000
5	(a)	-1.000	-1.000	-1.000
6	(b)	-1.000	1.000	1.000
7	(c)	1.000	-1.000	1.000
8	(b)	1.000	-1.000	-1.000
9	(a)	1.000	1.000	1.000
10	(d)	1.000	-1.000	1.000
11	(b)	1.000	1.000	-1.000
12	(b)	-1.000	-1.000	-1.000

Case 2

i	$x_{i1}(t)$	x_{i2}	x_{i3}	x_{i4}
1	(b)	-1.000	-1.000	-1.000
2	(a)	-1.000	1.000	1.000
3	(a)	0.000	0.000	1.000
4	(b)	0.000	-1.000	1.000
5	(a)	0.000	0.000	-1.000
6	(a)	0.000	-1.000	0.000
7	(c)	-1.000	0.000	0.000
8	(a)	1.000	-1.000	0.000
9	(b)	0.000	0.000	0.000
10	(b)	1.000	0.000	1.000
11	(d)	-1.000	0.000	0.000
12	(b)	0.000	1.000	0.000

TABLE 6.5: A-optimal designs for the single profile factor and main effects (Case 1) and main & quadratic effects (Case 2) of the three scalar factors for the FLM. The profile factor represented by a step function basis with $n_{x,1} = 4$ basis functions and the functional parameter by a linear basis. The column of the profile is labelled (a)-(d) for the optimal functions demonstrated on Figure 6.10.

6.3 FLM involving one profile factor and linear B-spline basis

In this section, the FLM considered is identical to model with one profile factor in (6.2). The only modification in the settings of the profile factor is the choice a BS basis of degree $d = 1$, instead of $d = 0$. For the functional parameters, only the linear basis is considered. For consistency purposes, the number of runs considered is $n \in \{4, 8, 12\}$ and the number of basis functions considered is $n_{x,1} \in \{3, 4, 8, 16, 100\}$. Knots are equally spaced and the number of basis functions depends on the length of the knot vector and the degree of the spline, see (4.3). The objective is to identify A-optimal designs and study their performance to changes in the settings of the model.

Results are given in Table 6.6, and A-optimal functions are demonstrated in Figures 6.11 and 6.12. Similar to the example in 6.1, the A-optimal designs perform better as the number of basis functions increases. However, the A-optimal designs for $n_{x,1} = 4$ are at least 90.2% efficient, compared to the the A-optimal design for $n_{x,1} = 100$. For instance, the A-optimal design with $n_{x,1} = 16$ is 99.7% A-efficient. Meaning that, there is no potential gain by increasing the number of basis functions even further.

On the other hand, the functions for the design are less complicated simpler for larger values of $n_{x,1}$. For instance, for $n_{x,1} = 8$, the chosen functions commonly have four changes in slope; see Figure 6.11, whereas the functions for $n_{x,1} = 16$ have only two changes; see Figure 6.12. This difference demonstrates the importance of the location of the breakpoints. For higher values of $n_{x,1}$, there is more flexibility in the location of the changes in the functions. For instance, for high $n_{x,1}$, each chosen function consists of two constant parts, at the boundaries, joined by a linear component with finite slope. A comparison with the A-optimality values of the designs found assuming step functions on Table 6.1, shows that the added complexity of the linear spline basis did not result in better designs.

	$n=4$		$n=8$		$n=12$	
$n_{x,1}$	A-opt	A-eff	A-opt	A-eff	A-opt	A-eff
3	9.450	0.889	6.224	0.624	4.123	0.609
4	9.314	0.902	4.168	0.931	2.759	0.910
8	8.594	0.978	3.940	0.985	2.571	0.977
16	8.433	0.997	3.895	0.997	2.528	0.994
100	8.405	1.000	3.882	1.000	2.512	1.000

TABLE 6.6: A-optimality values and A-efficiency values with $n = 4, 8, 12$ for the linear basis for $\beta_2(t)$ and BS basis of degree $d = 1$ for $x_{\cdot 1}(t)$ for the FLM.

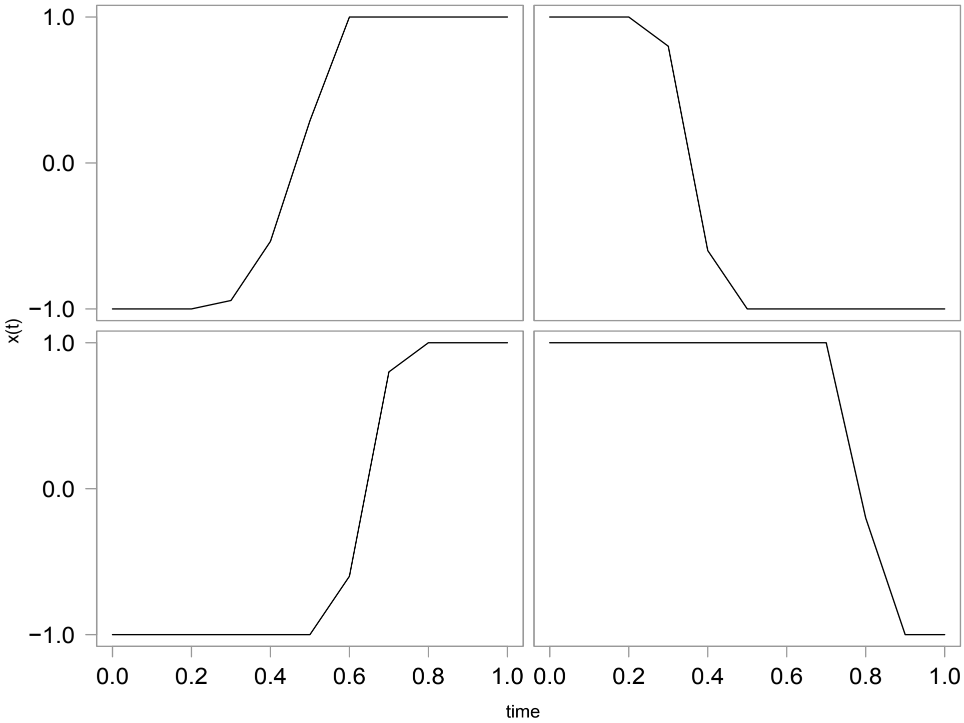


FIGURE 6.11: Four run A-optimal design for $n_{x,1} = 8$, linear basis for $\beta_2(t)$ and BS basis of degree $d = 1$ for $x_{\cdot 1}(t)$ for the FLM .

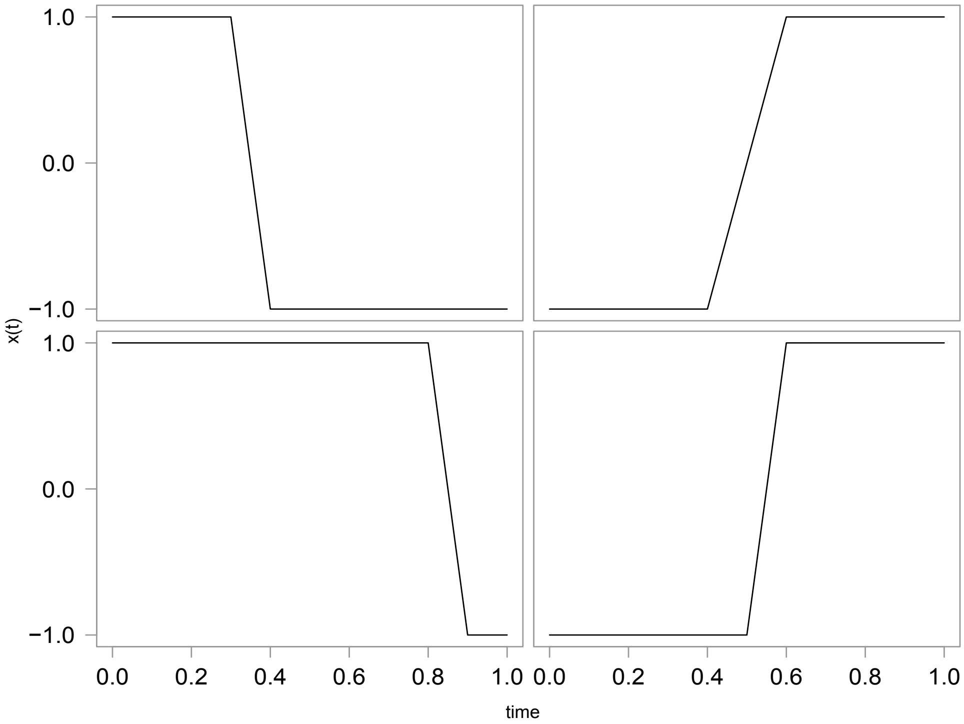


FIGURE 6.12: Four run A-optimal design for $n_{x,1} = 16$, linear basis for $\beta_2(t)$ and BS basis of degree $d = 1$ for $x_{\cdot 1}(t)$ for the FLM.

6.4 FLM involving two profile factors with and without their interaction

In this section, a FLM that involves two profile factors is considered. The aim is to identify D-optimal designs for two different scenarios and compare their performance. At first, the model that contains the intercept and the main effects of the profile factors is considered. Following that, the procedure is repeated with the addition of the interaction of the profile factors. For only main effects of the two profile factors, the functional of the profile factors $\boldsymbol{f}$ is,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \begin{pmatrix} 1 & x_{i1}(t) & x_{i2}(t) \end{pmatrix}, \tag{6.7}$$

with $x_{i1}(t)$ and $x_{i2}(t)$ the profile factors at the i^{th} run of the experiment. For the addition of the interaction of the profile factors, the functional of the profile factors $\boldsymbol{f}$ is,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \begin{pmatrix} 1 & x_{i1}(t) & x_{i2}(t) & x_{i1}(t)x_{i2}(t) \end{pmatrix}, \tag{6.8}$$

with $x_{i1}(t)x_{i2}(t)$ the interaction at the i^{th} run of the experiment.

The FLM involving the two profile factors and no interactions takes the form of the model discussed in Section 5.2 and it is given by,

$$\begin{gathered} y_i = \beta_1 + \int_0^{\mathcal{T}} x_{i1}(t)\beta_2(t)\ dt + \int_0^{\mathcal{T}} x_{i2}(t)\beta_3(t)\ dt + \epsilon_i, \\ i = 1, 2, \ldots, n,\ t \in [0,1],\ -1 \leq x_{ij}(t) \leq 1,\ j = 1, 2. \end{gathered} \tag{6.9}$$

The FLM involving the interaction of the two profile factors contains the extra interaction term and takes the form of the model in Section 5.3 and it is given by,

$$\begin{gathered} y_i = \beta_1 + \int_0^{\mathcal{T}} x_{i1}(t)\beta_2(t)\ dt + \int_0^{\mathcal{T}} x_{i2}(t)\beta_3(t)\ dt + \int_0^{\mathcal{T}} x_{i1}(t)x_{i2}(t)\beta_4(t)\ dt + \epsilon_i, \\ i = 1, 2, \ldots, n,\ t \in [0,1],\ -1 \leq x_{ij}(t) \leq 1,\ j = 1, 2, \end{gathered} \tag{6.10}$$

It is assumed that the control of the profile factors is represented by a BS basis of degree $d = 3$, i.e., cubic splines. The knot vector consists of four equally spaced interior knots, i.e., $\boldsymbol{\lambda} = (0.20.0.40, 0.60, 0.80)$. This means that the number of basis functions for the profile factors is $n_{x,1} = n_{x,2} = 3 + 4 + 1 = 8$. For the functional parameters, a BS basis of degree $d = 0$ and a single knot at $t = 0.5$ is considered, i.e., $n_{\beta,2} = n_{\beta,3} = n_{\beta,4} = 2$. The number of runs is $n = 12$.

For the model with just the main effects of the two profile factors, the objective value of a D-optimal design is 0.291. The objective value of a D-optimal design with the addition of the interaction term, is 0.335. The design found for no interaction, evaluated under the model with added interaction gives a D-optimality value of 0.361, which is 92.8% D-efficient. An evaluation of the design for the model which includes the interaction under the model without interaction, gives a D-optimality value of 0.305, which is 95.4% D-efficient. This is an indication that the D-optimal design of the FLM with the interaction is more robust to the choice of model, as it is efficient even under settings for no interaction.

The functions of the profile factors for the main effects model are either smooth functions or constant functions at the boundaries. For both profile factors, six out of the 12 runs runs are constant functions at the boundaries, and six runs are smooth upward or downward functions. The optimal functions of $x_{i1}(t)$ and $x_{i2}(t), i = 1, 2, \ldots 12$, for the model in (6.9) are shown in Figure 6.13 and Figure 6.14.

The addition of the interaction term in the model caused the functions to become slightly less complicated. Specifically, four out of the 12 runs are smooth upward or downward functions, and eight runs are constant functions at the boundaries. The optimal functions of $x_{i1}(t)$ and $x_{i2}(t), i = 1, 2, \dots 12$, for the model in (6.10) are shown in Figure 6.15 and Figure 6.16.

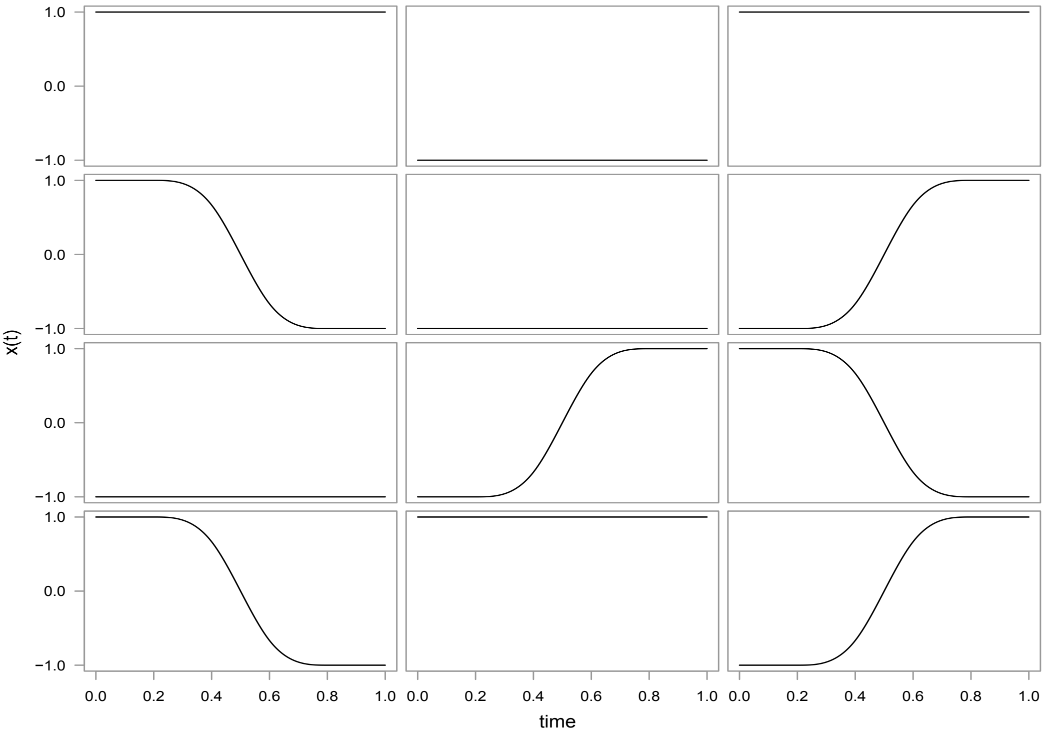


FIGURE 6.13: 12 run D-optimal design for $\boldsymbol{x}_{\cdot 1}(t)$ of a FLM with two profile factors of cubic BS basis and four equally spaced knots so that $n_{x,1} = 8$, degree $d = 0$ BS basis for the functional parameters and no interaction effect considered.

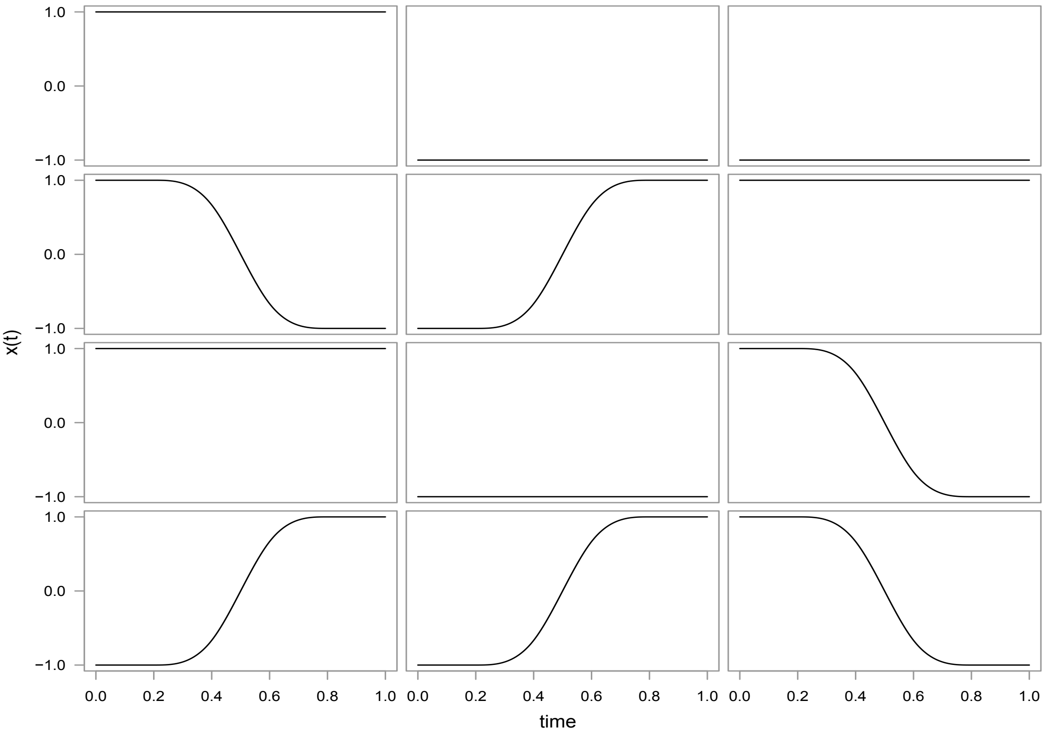


FIGURE 6.14: 12 run D-optimal design for $\boldsymbol{x}_{\cdot 2}(t)$ of a FLM with two profile factors of cubic BS basis and four equally spaced knots so that $n_{x,1} = 8$, degree $d = 0$ BS basis for the functional parameters and no interaction effect considered.

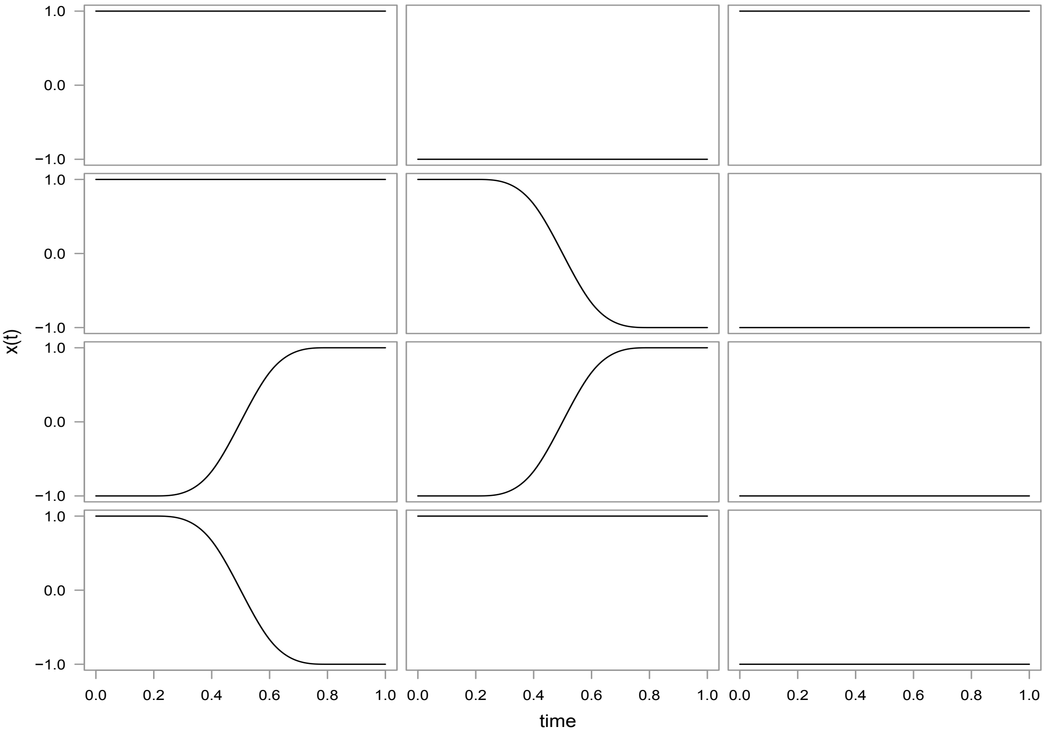


FIGURE 6.15: 12 run D-optimal design for $\boldsymbol{x}_{\cdot 1}(t)$ of a FLM with two profile factors of cubic BS basis and four equally spaced knots so that $n_{x,1} = 8$, degree $d = 0$ BS basis for the functional parameters and interaction effect considered.

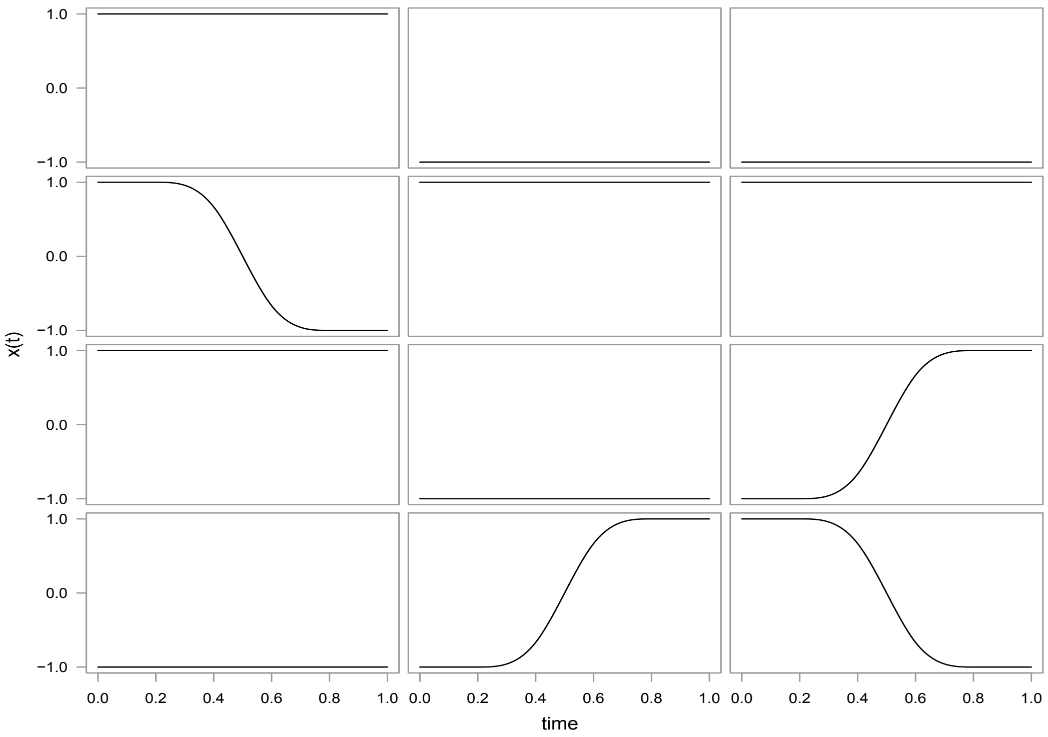


FIGURE 6.16: 12 run D-optimal design for $\boldsymbol{x}_{\cdot 2}(t)$ of a FLM with two profile factors of cubic BS basis and four equally spaced knots so that $n_{x,1} = 8$, degree $d = 0$ BS basis for the functional parameters and interaction effect considered.

To investigate further the effect of the interaction term, the settings of the basis of the functional parameters are modified. For instance, the parameters of the first profile factor and the interaction term are assumed to be represented by a BS basis of degree $d = 3$ with a single knot at $t = 0.5$, i.e., $n_{\beta,2} = n_{\beta,4} = 5$. The parameter of the second profile factor is still represented by a BS basis of degree $d = 0$ with a single knot at $t = 0.5$, i.e., $n_{\beta,3} = 2$. The number of runs remains unchanged at $n = 12$ and the objective is still to identify D-optimal designs.

For the model with just the main effects of the two profile factors, the objective value of a D-optimal design is 3.183. Thus, the use of a cubic spline for the parameter basis of the first profile factor caused an increase in the objective value of the D-optimal design. Moreover, the optimal functions of the profile factor are more complicated; see Figure 6.18. The functions of the second profile factor have not changed.

For the model that involves the interaction of the profile factors, the objective value of a D-optimal design is 125.833. In the first example in this section, the D-optimality values of the models with and without the interaction were close. However, the use of cubic spline for the parameters increase the objective value of the model with the interaction. In addition, the functions of the profile factor $x_{i1}(t), i = 1, 2, \ldots 12$, are more wiggly; see

Figure 6.17. On top of that, the functions of $x_{i1}(t), i = 1, 2, \ldots 12$, for the model without interaction are more complicated compared to the functions for the model with the interaction. This is noticeable from the values in red in Figure 6.18 and Figure 6.17 which represent the penalised complexity value of each function, using the squared differences penalty method from Section 5.10. For design functions for the model with no interaction the P-spline penalty is 14.667 on average, while for the model with an interaction the penalty is 8.548 on average. The higher the penalty, the more complicated the functions; see Section 5.10.

The design found for no interaction, evaluated under the model with added interaction gives a D-optimality value of 184.282, which is 68.3% D-efficient. An evaluation of the design for the model which includes the interaction under the model without interaction gives a D-optimality value of 4.049, which is 78.6% D-efficient. As in the first part of the section, the design with the interaction is more robust to the choice of model.

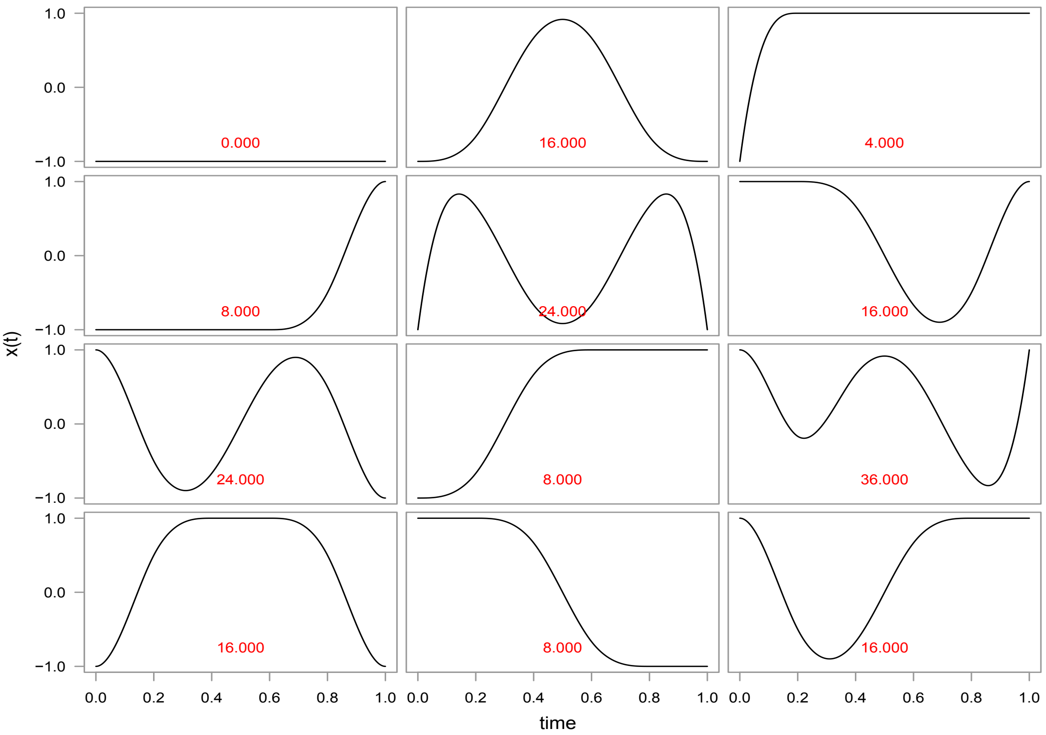


FIGURE 6.17: 12 run D-optimal design for $\boldsymbol{x}_{\cdot 1}(t)$ of a FLM with two profile factors of cubic BS basis and four equally spaced knots so that $n_x = 8$, degree $d = 3$ and $d = 0$ BS basis for the functional parameters and no interaction effect considered. The values in red correspond to the complexity value of each function.

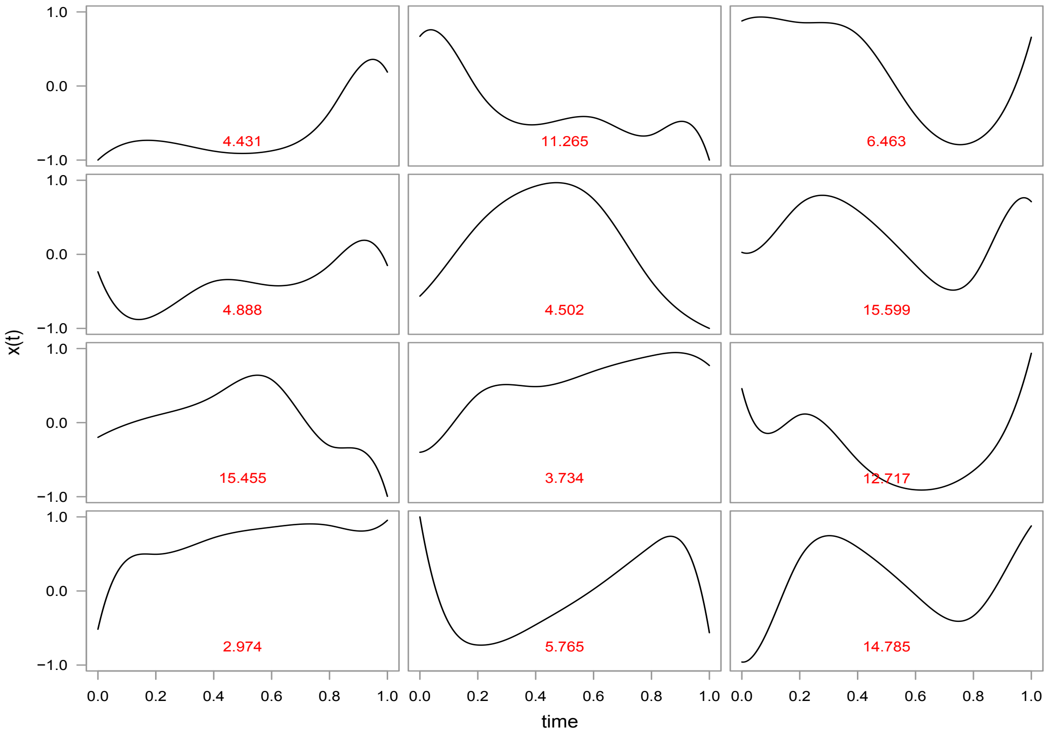


FIGURE 6.18: 12 run D-optimal design for $\boldsymbol{x}_{\cdot 1}(t)$ of a FLM with two profile factors of cubic BS basis and four equally spaced knots so that $n_x = 8$, degree $d = 3$ and $d = 0$ BS basis for the functional parameters and interaction effect considered. The values in red correspond to the complexity value of each function.

6.5 Bayesian approach to a FLM involving one profile factor

In this section, the FLM considered is identical to model with one profile factor from (6.2). The control of the profile factor is represented via step functions, through BS basis of degree $d = 0$. The uniqueness of the example in this section is that the Bayesian approach via a roughness penalty is followed. Thus, the aim is to identify Bayesian A- and D- optimal designs, maximising (5.41) and (5.44), respectively. After that, it is desired to investigate the effect of Λ on the performance of the optimal designs. The prior choice for variance is an inverse gamma distribution such that, $\sigma^2 \sim IG(2, 1)$. The latter prior distribution achieves small precision matrix $\boldsymbol{V}^{-1} = \Lambda \boldsymbol{R}_0$.

The Bayesian approach for fitting FLMs is connected to the use of roughness penalties to penalise the complexity of functions; see Section 5.6. In addition, the roughness penalties are defined as integrals over time of the second derivative of the functional parameters. This means that, if a linear basis for the functional parameters is considered, then the roughness matrix has all values being zero. This is because the second derivatives of linear terms are zero, thus, the problem is equivalent to the frequentist approach.

For the latter reason, the functional parameter is represented by a quadratic basis for which $D^2\boldsymbol{b}_2(t) = (0,0,2)$ and matrix $\boldsymbol{R}_p$ has a non zero entry, i.e., $r_{33} = 4$. The choice of smoothing values considered are $\Lambda \in \{0.01, 1, 10\}$ and the number of basis functions for the profile factor are $n_{x,1} \in \{3,4,8\}$. The choice of runs for the experiment are $n \in \{4,12\}$. The objective values for the Bayesian A- and D- optimal designs are shown in Table 6.7 and Table 6.8, respectively.

	$\Lambda = 0.01$		$\Lambda = 1$		$\Lambda = 10$	
$n_{x,1}$	$n = 4$	$n = 12$	$n = 4$	$n = 12$	$n = 4$	$n = 12$
3	58.183	39.846	9.343	3.298	8.880	2.830
4	57.772	36.233	9.257	3.083	8.801	2.622
8	55.827	34.933	9.002	3.054	8.544	2.591

TABLE 6.7: A-optimality values under the Bayesian approach for $n \in \{4,12\}$ and $\Lambda \in \{0.01, 1, 10\}$, for the quadratic basis of $\beta_2(t)$ for the FLM.

	$\Lambda = 0.01$		$\Lambda = 1$		$\Lambda = 10$	
$n_{x,1}$	$n = 4$	$n = 12$	$n = 4$	$n = 12$	$n = 4$	$n = 12$
3	2.329	0.995	0.740	0.320	0.416	0.180
4	2.236	0.981	0.707	0.310	0.398	0.174
8	2.236	0.981	0.707	0.310	0.398	0.174

TABLE 6.8: D-optimality values under the Bayesian approach for $n \in \{4,12\}$ and $\Lambda \in \{0.01, 1, 10\}$, for the quadratic basis of $\beta_2(t)$ for the FLM.

At first, as Λ increases, the designs found are less complicated, i.e., less changes in the step functions. For Bayesian A-optimal designs and small values of Λ, there are at most two changes in the step functions. Moreover, for small values of Λ, the A-optimal designs are similar to the A-optimal designs in the frequentist approach; see Figure 6.19. However, when Λ increases, there is at most one change in the step functions, and the A-optimal designs are similar to the A-optimal designs for the linear basis in the frequentist approach; see Figure 6.20. This is because, big values of Λ penalise wiggly functions and the only term to be penalised is the quadratic term.

For Bayesian D-optimal designs, even small value of penalty penalise the complexity of functions enough. For instance, the functions of the profile factor have at most one change, for every choice of $\Lambda \in \{0.01, 1, 10\}$ and $n \in \{4,12\}$. For an example of D-optimal functions for small penalty $\Lambda = 0.01$; see Figure 6.21.

To add a point, findings are in line with the frequentist approach conclusions in Section 6.1. The change in the A-optimality objective values from four to eight basis functions is not significant and there is no change in the D-optimality objective values.

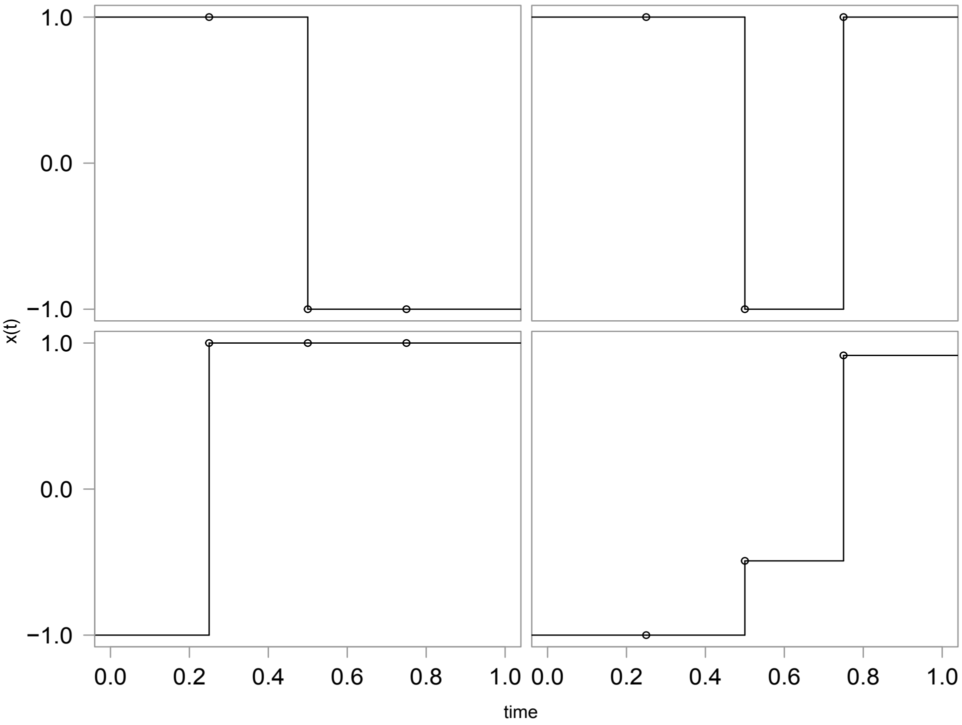


FIGURE 6.19: Four run Bayesian A-optimal design for $n_{x,1} = 4$, $\Lambda = 0.01$ and quadratic basis for $\beta_2(t)$ for the FLM.

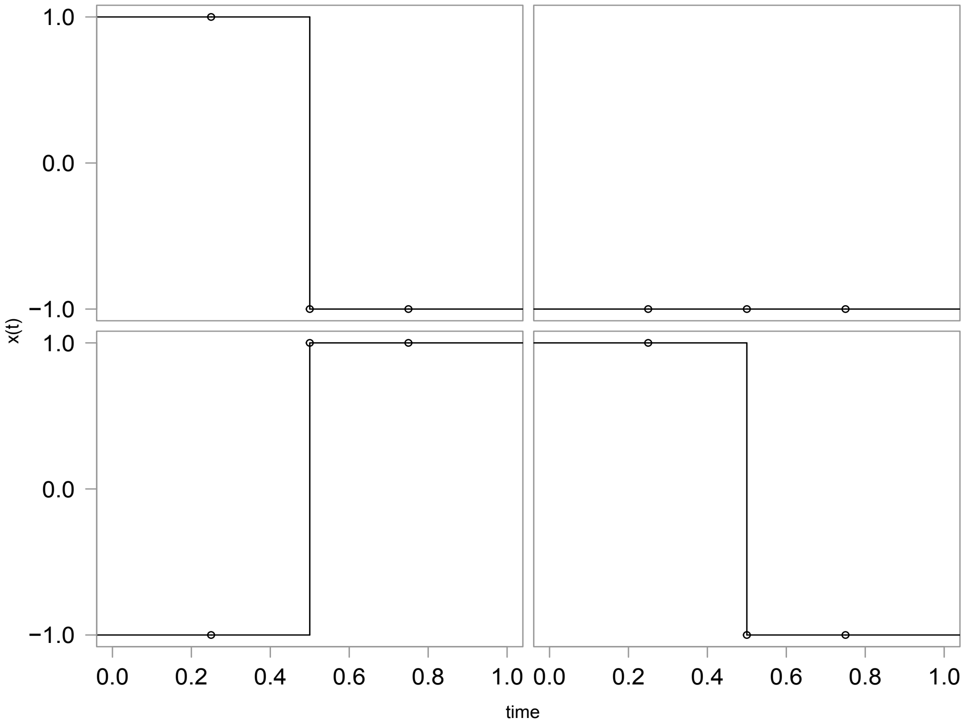


FIGURE 6.20: Four run Bayesian A-optimal design for $n_{x,1} = 4$, $\Lambda = 10$ and quadratic basis for $\beta_2(t)$ for the FLM.

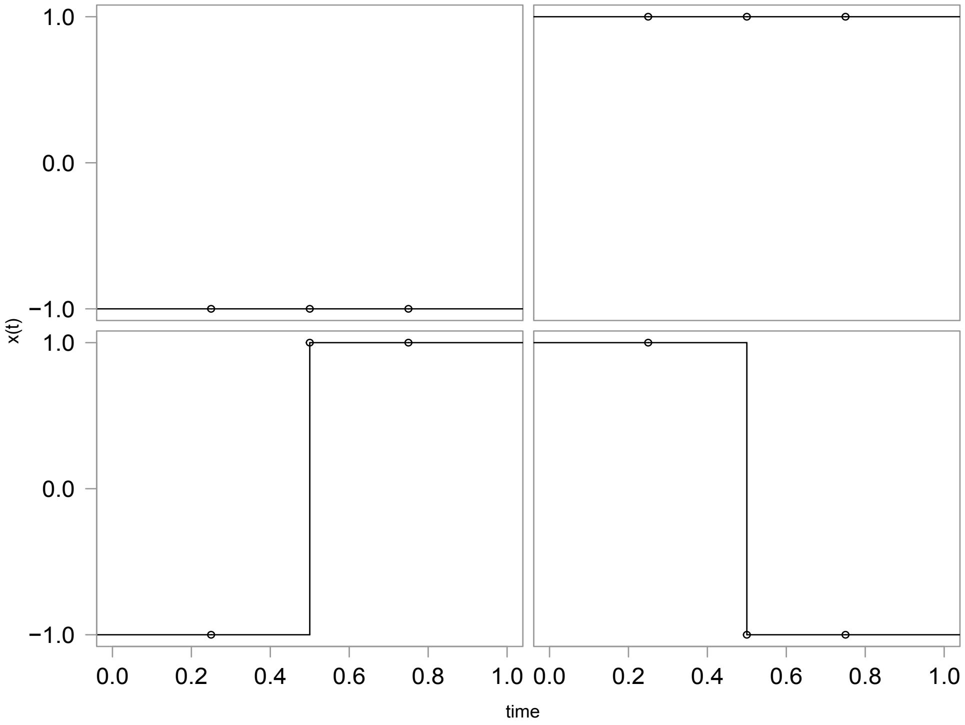


FIGURE 6.21: Four run Bayesian D-optimal design for $n_{x,1} = 4$, $\Lambda = 0.01$ and quadratic basis for $\beta_2(t)$ for the FLM.

6.6 Bayesian design criterion tailored for profile factors to a FLM with one profile and two scalar factors

In this section, a FLM with one profile and two scalar factors is considered. The aim of the example is to identify Bayesian A-optimal designs under the criterion in (5.41) and the Bayesian criterion tailored to profile factors in (5.46). The functional of the profile factors $\boldsymbol{f}$ is,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \begin{pmatrix} 1 & x_{i1}(t) & x_{i2} & x_{i3} \end{pmatrix}, \tag{6.11}$$

with $x_{i1}(t)$ the single profile factor at the i^{th} run of the experiment, and x_{i2} and x_{i3} the scalar factors at the i^{th} run of the experiment. Thus, the FLM takes the form,

$$y_i = \beta_1 + \int_0^{\mathcal{T}} \beta_2(t) x_{i1}(t)\, dt + \beta_3 x_{i2} + \beta_4 x_{i3} + \epsilon_i, \quad i = 1, ..., n,\ t \in [0, 1],\ -1 \leq \boldsymbol{x}_i(t) \leq 1. \tag{6.12}$$

For the profile factor, it is assumed that control is represented by a BS of degree $d = 1$ with a single knot at $t = 0.5$. For the functional parameter a linear basis is considered, and the number of runs of the experiment is $n = 12$.

The choice of Λ does not affect the designs, i.e., the designs are invariant to the value of Λ. This is because in the roughness matrix that includes the second derivatives, all the values are zero. The Bayesian design criterion that is tailored for profile factors requires the calculation of the matrix $\boldsymbol{A}$ which depends on the matrix $\boldsymbol{C}(t)$. The matrix $\boldsymbol{C}(t)$ includes the functions and parameters of interest. It is assumed that interest lies in the functional parameter and hence, the linear combination $\theta_1 + \theta_2 t$. Thus, the matrix $\boldsymbol{C}(t)$ is a 5×4 matrix such that,

$$\boldsymbol{C}(t) = \begin{pmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3^T & 1 \\ \boldsymbol{0}_3^T & t \end{pmatrix},$$

and

$$\boldsymbol{A} = \int_0^1 \boldsymbol{C}(t)\boldsymbol{C}(t)^T\, dt = \begin{pmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3^T & 1 & 1/2 \\ \boldsymbol{0}_3^T & 1/2 & 1/3 \end{pmatrix}.$$

To evaluate the performance of the Bayesian design criterion that is tailored for profile factors, the optimal designs are evaluated under the alternative criteria; see Table 6.9. The optimal designs under the two criteria are different. The efficiency values shown in brackets in Table 6.9, prove that the optimal designs are performing well under both criteria. However, the efficiency value of the optimal design under the Bayesian A-optimality design criterion is slightly higher. Thus, the optimal design identified under the Bayesian A-optimality design criterion, is more robust to the choice of design criterion.

	Tailored criterion	A-optimality
Tailored design	4.228 (1.000)	4.362 (0.988)
A-optimal design	4.254 (0.994)	4.310 (1.000)

TABLE 6.9: Objective values of the Bayesian A-optimality and Bayesian tailored to profile factors design criteria, with BS basis of degree $d = 1$ for $\boldsymbol{x}_{\cdot 1}(t)$ and single knot at $t = 0.5$, linear basis for $\beta_2(t)$ and $n = 12$. Efficiency values are shown in brackets after an evaluation of the optimal designs under the alternative criterion.

For the Bayesian A-optimality design criterion, there are five unique functions of the profile factor; see Figure 6.22. The rest are repeated functions. Moreover, the design values for the scalar factors are always at the boundaries. The combinations of the functions of the profile factor and the values of the scalar factors under the Bayesian A-optimality design criterion are given in Table 6.10.

i	$x_{i1}(t)$	x_{i2}	x_{i3}
1	(a)	-1.000	1.000
2	(b)	-1.000	1.000
3	(c)	1.000	-1.000
4	(a)	1.000	1.000
5	(b)	-1.000	1.000
6	(c)	1.000	1.000
7	(b)	1.000	-1.000
8	(e)	-1.000	-1.000
9	(a)	-1.000	-1.000
10	(d)	1.000	1.000
11	(c)	-1.000	-1.000
12	(d)	1.000	-1.000

TABLE 6.10: 12 run optimal design of the Bayesian A-optimality design criterion. For the functions of the profile factor labelled (a)-(e), refer to Figure 6.22

For the Bayesian design criterion tailored to profile factors, there are four unique functions of the profile factor; see Figure 6.23. The rest are repeated functions. The design values for the scalar factors are not always at the boundaries. The combinations of the functions of the profile factor and the values of the scalar factors under the Bayesian design criterion tailored to profile factors are given in Table 6.11.

i	$x_{i1}(t)$	x_{i2}	x_{i3}
1	(a)	1.000	1.000
2	(b)	-1.000	0.443
3	(a)	-1.000	-1.000
4	(c)	-1.000	-1.000
5	(d)	-1.000	-1.000
6	(c)	1.000	1.000
7	(d)	1.000	1.000
8	(b)	0.925	-1.000
9	(c)	1.000	-0.483
10	(c)	-1.000	1.000
11	(d)	-1.000	-1.000
12	(a)	1.000	1.000

TABLE 6.11: 12 run optimal design of the Bayesian design criterion tailored to profile factors. For the functions of the profile factor labelled (a)-(d), refer to Figure 6.23

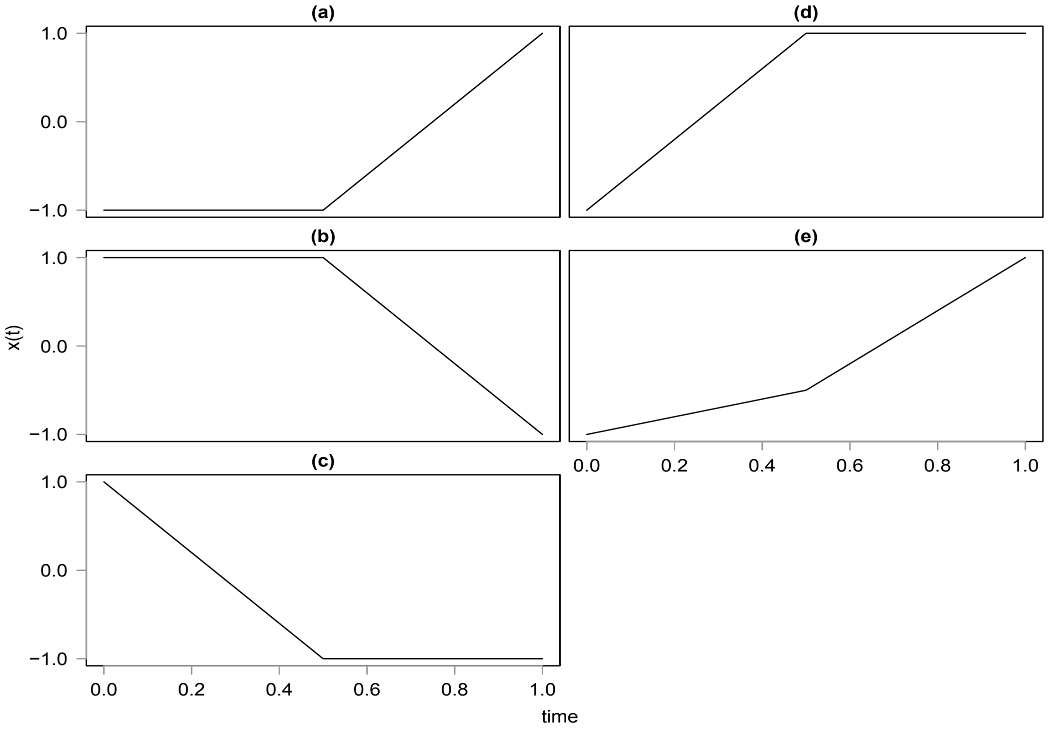


FIGURE 6.22: Five unique functions of the profile factor under the Bayesian A-optimality design criterion, with BS basis of degree $d = 1$ for $x_{\cdot 1}(t)$, single knot at $t = 0.5$ and linear basis for $\beta_2(t)$.

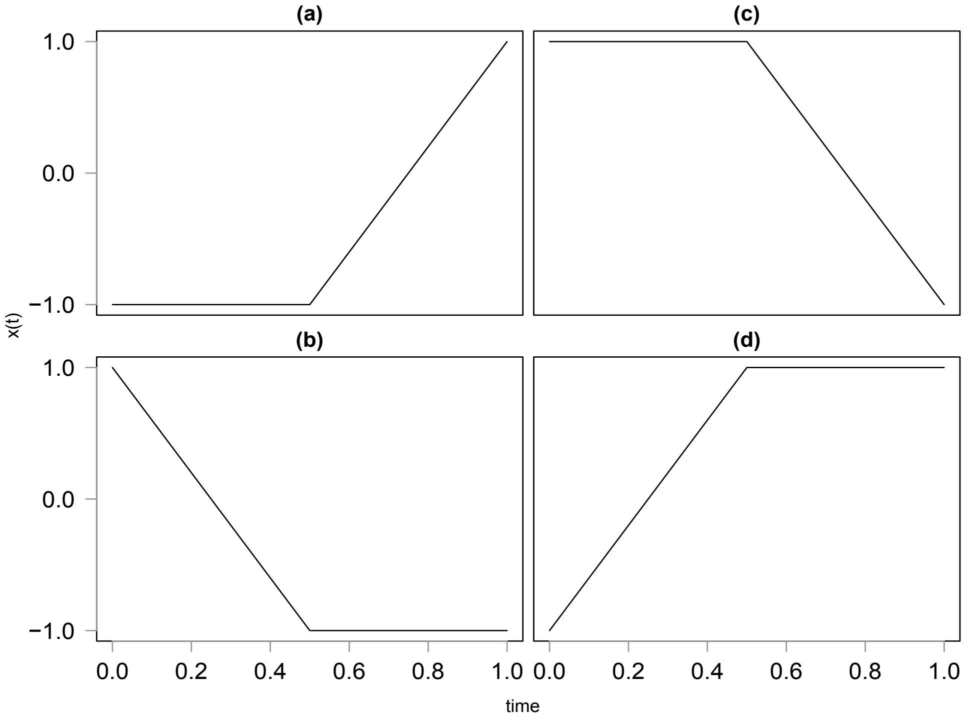


FIGURE 6.23: Four unique functions of the profile factor under the Bayesian design criterion tailored to profile factors, with BS basis of degree $d = 1$ for $x_{\cdot 1}(t)$, single knot at $t = 0.5$ and linear basis for $\beta_2(t)$.

6.7 BS basis for the functional parameters

In the previous examples in this chapter, focus has been given on how the performance of optimal designs is affected by changes in the complexity of settings of the profile factors. For this reason, the basis of the functional parameters are kept simple; for instance, linear and quadratic. In this section, the aim is to investigate the performance of the A-optimal designs under the assumption of more complicated basis for the functional parameters $\boldsymbol{\beta}(t)$. For the investigation, a FLM involving a single profile factor that takes the form of the model in (6.2) is considered. The number of runs of the experiment is $n = 12$.

For the profile factor, it is assumed that control is represented through a cubic BS basis, i.e., $d = 3$. The functional parameter is assumed to be represented by a linear BS basis, i.e., $d = 1$. For the basis of the profile factor, the knot vector consists of 19 equally spaced knots. Thus, the number of basis functions is $n_{x,1} = 23$; see (4.3). For the basis of the functional parameters, the number of basis functions is $n_{\beta,2} \in \{3, 4, 5\}$; defined from the choice of 1-3 equally spaced knots.

Due to the increase in complexity of the basis of the functional parameter, the functions of the profile factor are more complicated. For $n_{\beta,2} = 3$, the functions are similar to the functions found for a quadratic basis, but slightly smoother due to the use of a B-spline basis; see Figure 6.24. As the number of basis functions for the functional parameters increases, the functions are getting more complicated, i.e., more wiggly; see Figure 6.25 and Figure 6.26. The complexity of the functions is calculated using the squared differences penalisation method from Section 5.10, and the values are shown on the figures in red. The average wiggliness values of the functions when $n_{\beta,2} = 3, 4, 5$ are 10.667, 14.667 and 20.000, i.e., average wiggliness increases as $n_{\beta,2}$ increases.

The objective values of the A-optimal designs for the three scenarios are in Table 6.12. The objective values increase as the number of basis functions increases. This makes sense because the number of parameters increases. To investigate if the increase in the number of parameters is useful, the optimal designs are evaluated under the settings of $n_{\beta,2} = 3$ basis functions; see the last column of Table 6.12. From the investigation, the complicated designs have higher variance when evaluated under a simpler model. Hence, a large number of basis functions for the functional parameters is not needed, unless the response surface is or is thought to be more complicated. Hence, the number of basis functions needs to be chosen based on the flexibility needed to estimate the functions of the profile factors.

$n_{x,1}$	$n_{\beta,2}$	A-opt	A-opt evaluated for $n_{\beta,2} = 3$
23	3	5.386	5.386
23	4	13.062	7.662
23	5	26.120	11.281

TABLE 6.12: A-optimality objective values for cubic BS basis for the single profile factor, linear BS basis for the functional parameter, the number of basis functions for the profile factor constantly being $n_{x,1} = 23$, the basis functions for the parameter changing between $n_{\beta,2} = 3, 4, 5$ and $n = 12$ runs. The last column contains evaluations of the A-optimal designs under the setting of $n_{\beta,2} = 3$.

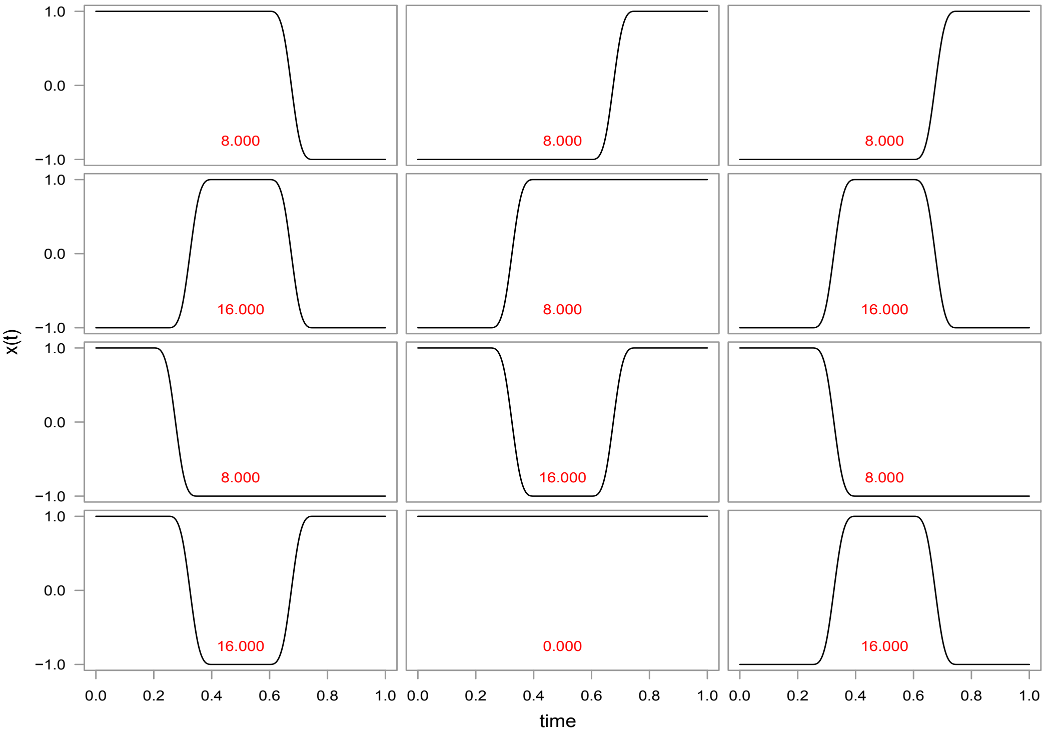


FIGURE 6.24: 12 run A-optimal design with BS basis for $\boldsymbol{x}_{\cdot 1}(t)$ and $\beta_2(t)$, with $n_{x,1} = 23$ and $n_{\beta,2} = 3$. The values in red correspond to the complexity value of each function.

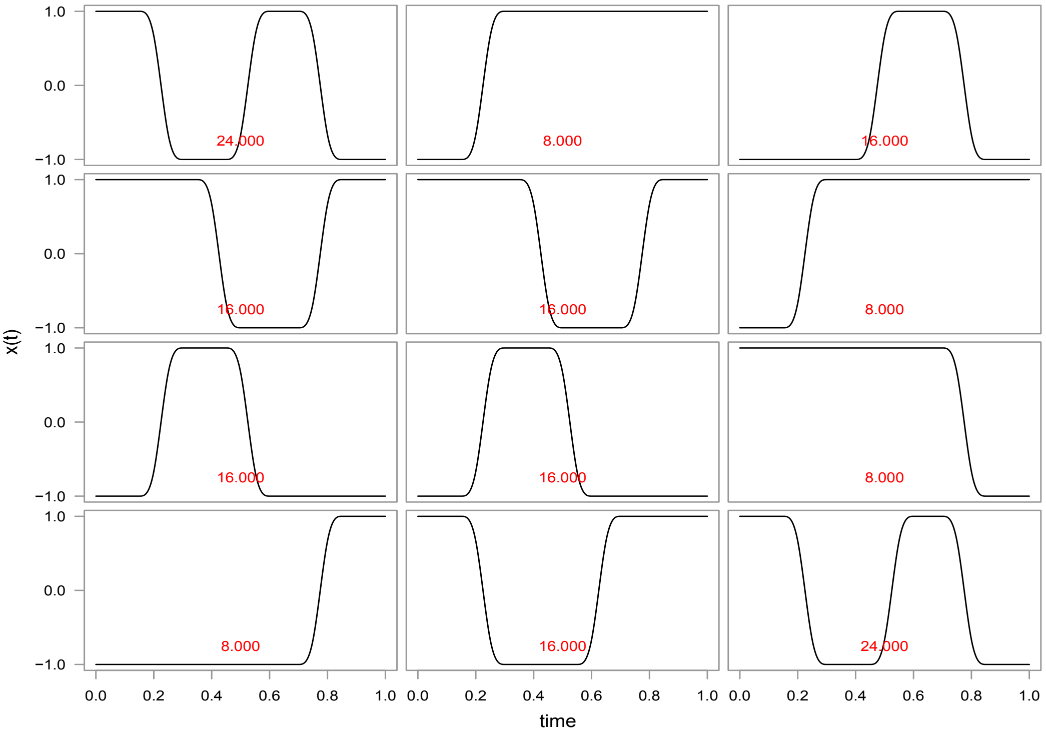


FIGURE 6.25: 12 run A-optimal design with BS basis for $\boldsymbol{x}_{\cdot 1}(t)$ and $\beta_2(t)$, with $n_{x,1} = 23$ and $n_{\beta,2} = 4$. The values in red correspond to the complexity value of each function.

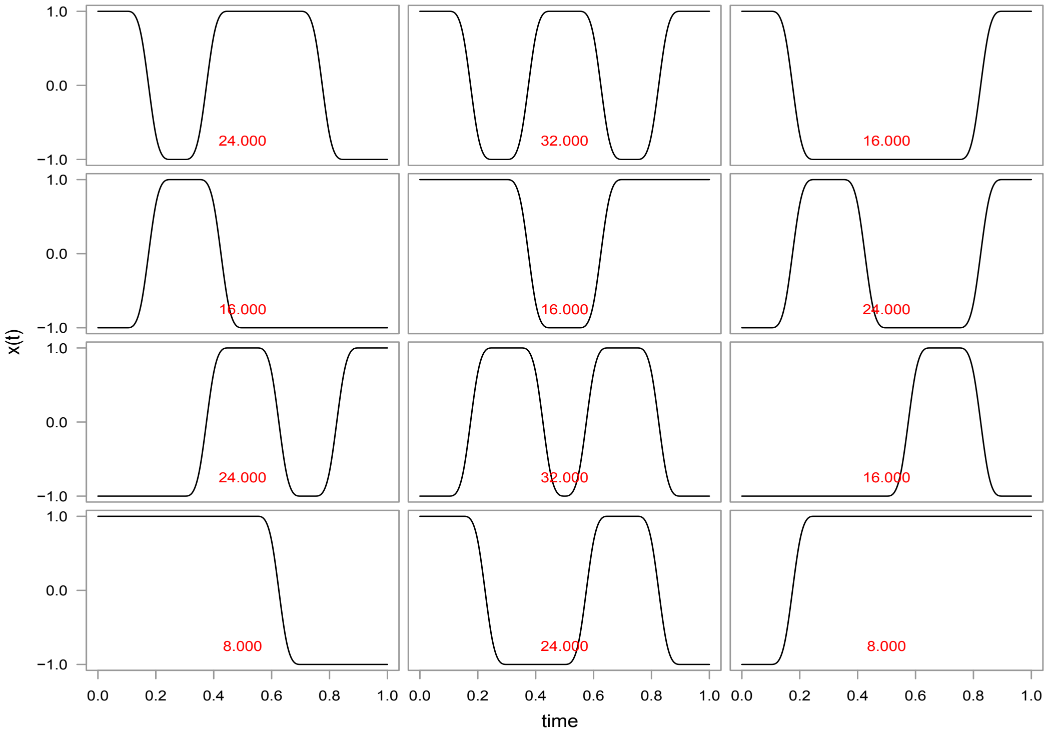


FIGURE 6.26: 12 run A-optimal design with BS basis for $\boldsymbol{x}_{\cdot 1}(t)$ and $\beta_2(t)$, with $n_{x,1} = 23$ and $n_{\beta,2} = 5$. The values in red correspond to the complexity value of each function.

6.8 Conclusion

In this chapter, functional linear models were investigated and functions of profile factors for A- and D-optimal designs have been identified. An investigation on how the performance of the optimal designs is affected by changes in the number of runs, the number of basis functions and the basis system for the parameters has been carried out. Control of the profile factors was assumed to be represented by B-spline basis of certain degree and knots. Moreover, the basis for the functional parameters was assumed to be either power series basis or B-spline basis.

It has been showed that increasing the number of runs, improves the performance of designs in terms of A- and D-optimality, but the functions have more repetitions. If the basis for the parameters is linear, increasing the number of basis functions for the profile factors does not improve the performance of the designs significantly. Thus, a large number of basis functions is not needed. However, if the basis for the parameters is quadratic, the improvement is more significant, but huge number of basis functions is similarly unnecessary. As the number of basis functions gets big, the improvement by further increase in basis functions is getting less. To add a point, the A- and D-optimal designs are more complicated as the basis for the parameters becomes more complicated, i.e., from linear to quadratic. A further increase in the complexity of the parameters using B-spline basis have been considered, and the functions of the profile factors were even more complicated.

Bayesian optimal designs have been identified using the roughness penalty approach. An investigation on the sensitiveness of the optimal designs in changes to the smoothing parameter have been carried out. The latter allowed a comparison between the frequentist and the Bayesian optimal designs. From the investigation, the conclusion was that for small values of penalty Λ, the Bayesian designs are similar to the frequentist designs and when the penalty Λ is big, the designs are similar to the designs with a linear basis. This happens because Λ penalises the quadratic term. Finally, optimal designs for the Bayesian criterion tailored to the design of the FLM that tackles to minimise the average posterior variance of the functional parameters $\boldsymbol{\beta}(t)$ directly averaged with respect to time have been identified.

Chapter 7

Design of Experiments for functional generalised linear models

Generalised linear models (GLMs) are an extension to the common linear model. As their name suggests, they generalise the idea of linear models, by using a link function to relate the linear model to response types that are not normally distributed; see Dobson and Barnett (2018). Following this concept, the aim of this chapter is to generalise and extend the functional linear model (FLM), to form functional generalised linear models (FGLM). The FGLM will then be able to model a response type that has a distribution other than normal, as a function of profile factors.

GLMs are described in Section 7.1. After that, FGLMs are discussed in Section 7.2, and the FLM methodology using basis functions in extended to FGLMs. The design problem is revised, and the challenge with optimality functions depending on the unknown parameters is addressed. For this reason, the pseudo-Bayesian approach, that assumes prior knowledge for the parameters, is followed. The pseudo-Bayesian approach is discussed in Section 7.3. However, identifying pseudo-Bayesian optimal designs is not straightforward, because they often result from analytically intractable, and usually high-dimensional integrals. The Monte Carlo and the quadrature approximation methods, used to integrate numerically the intractable integrals, are discussed in Section 7.4. The roughness penalty approach is extended to the FGLMs in Section 7.5.

Functional logistic and functional Poisson models depending on profile factors are considered in Sections 7.6 - 7.10. Sensitivity studies are raised, to investigate the performance of the pseudo-Bayesian optimal designs, in different settings of the profile factors and the functional parameters.

Moreover, FGLMs are computationally more expensive than FLMs. Thus, over-complicated settings for the profile factors and the functional parameters are avoided.

7.1 Introduction to GLMs

GLMs are a generalisation to the traditional linear models to consider response variables whose distribution is other than normal; see Dobson and Barnett (2018). They are widely used due to their flexibility in the response distribution and their variety of data types, i.e., binary or categorical responses; see Russell et al. (2009) and McGree and Eccleston (2012). Moreover, GLMs consist of three components which are defined below and discussed in detail in Agresti (2018, Chapter 3). For the rest of this section, the notation used is the notation from the linear model discussed in Chapters 1 and 2 and specifically equation (1.4).

1. Random component: The first component of the GLMs is the random component. It refers to the distribution of the response y_i, with i representing the i^{th} run of the experiment for $i = 1, 2, \ldots, n$. The distribution of the response variable is formed from a distribution that is a member of the exponential family of distributions (EFDs); see Wood (2017*a*, p. 103). The EFDs includes several distributions including the normal, Bernoulli, binomial, Poisson and gamma distributions. For instance, linear models are a special case of a GLM with normally distributed responses. In the examples later in this chapter, primary focus is given to the Bernoulli and Poisson distributions.

2. Systematic component: The second component of the GLMs is the systematic component. It refers to the linear predictor of the model, that relates a vector $\boldsymbol{\eta}$ to the controllable factors such that,

$$\boldsymbol{\eta} = \boldsymbol{F}\boldsymbol{\beta}.$$

The matrix $\boldsymbol{F}$ and the vector $\boldsymbol{\beta}$ are the model matrix and vector of parameters from (1.4), and $\boldsymbol{\eta}$ is a $n \times 1$ vector with the i^{th} entry being,

$$\eta_i = f(\boldsymbol{x}_i)^T \boldsymbol{\beta}, \quad i = 1, 2, \ldots, n.$$

3. Link function: The third and final component of the GLMs is the link function, that is is a one to one transformation $g(\mu_i)$. It provides a link between the random and the systematic components, with μ_i the mean of the i^{th} response, i.e., $\mathbb{E}(y_i) = \mu_i$. This means that, the mean response is linked to the linear predictor through the link function, such that,

$$g(\mu_i) = \eta_i = f(\boldsymbol{x}_i)^T \boldsymbol{\beta}, \quad i = 1, 2, \ldots, n.$$

> The choice of link function depends on the distribution of the responses. For example, the logit function is a link for the Bernoulli and binomial distributions, and the log function is a link for the Poisson distribution. In examples demonstrated later in this chapter, the choice of link function is logit for Bernoulli distributed responses, known as logistic model, and log for Poisson distributed responses, known as Poisson model. The logistic and the Poisson models are described in detail in Dobson and Barnett (2018, Chapter 7) and Dobson and Barnett (2018, Chapter 9), respectively.

The estimates of the parameters $\boldsymbol{\beta}$ can be found by solving $U(\boldsymbol{\beta}) = 0$, with $U(\boldsymbol{\beta})$ the score functions, derived by differentiating the log-likelihood with respect to the parameters; see Wood (2017*a*, p. 106). The score functions are given by,

$$U(\boldsymbol{\beta}, \boldsymbol{X}) = \boldsymbol{F}^T \boldsymbol{\Pi}(\boldsymbol{y} - \boldsymbol{\mu}) \tag{7.1}$$

which depends on $\boldsymbol{\beta}$ through the systematic component and the mean. The matrix $\boldsymbol{\Pi}$ is a $n \times n$ diagonal matrix with entries,

$$\pi_{ii} = \frac{1}{g'(\mu_i)} \frac{1}{\text{Var}(y_i)}, \quad i = 1, 2, \ldots, n,$$

and $\boldsymbol{X}$ is the design matrix, as discussed in Chapter 2. The Fisher information is a measure for the amount of information about parameters provided by experimental data. It is a well-established characteristic of an experimental design used to assess and optimize the design for maximizing the expected accuracy of parameter estimates. The Fisher information matrix is given by,

$$\mathcal{I}(\boldsymbol{\beta}, \boldsymbol{X}) = \boldsymbol{F}^T \boldsymbol{W} \boldsymbol{F} \tag{7.2}$$

where $\boldsymbol{W}$ is an $n \times n$ diagonal matrix with entries,

$$w_{ii} = \frac{1}{g'(\mu_i)^2} \frac{1}{\text{Var}(y_i)}, \quad i = 1, 2, \ldots, n.$$

Furthermore, since $\boldsymbol{y}$ follows a distribution which is a member of the exponential family of distributions, then using the log likelihood, the asymptotic variance-covariance matrix of the parameter estimators is given by the inverse of the Fisher information matrix,

$$\text{Var}(\hat{\boldsymbol{\beta}}) = (\boldsymbol{F}^T \boldsymbol{W} \boldsymbol{F})^{-1}, \tag{7.3}$$

discussed in Wood (2017*a*, p. 106-108).

Finally, the objective functions for A- and D-optimality, discussed in detail in Section 2.2 of Chapter 2, are updated to become,

$$\Psi_A(\boldsymbol{X}) = \text{tr}\big[\mathcal{I}(\boldsymbol{\beta}, \boldsymbol{X})^{-1}\big] = \text{tr}\big[(\boldsymbol{F}^T \boldsymbol{W} \boldsymbol{F})^{-1}\big] \tag{7.4}$$

$$\Psi_D(\boldsymbol{X}) = \text{det}\big[\mathcal{I}(\boldsymbol{\beta}, \boldsymbol{X})\big]^{-1/p} = \text{det}\big[\boldsymbol{F}^T \boldsymbol{W} \boldsymbol{F}\big]^{-1/p}. \tag{7.5}$$

The Fisher information matrix, thus, the designs, depend on the unknown model parameters. As a result, identifying optimal experimental designs is not as straightforward as for linear models. The difficulty of obtaining optimal designs, and an approach to incorporate prior information for the model parameters is discussed in Section 7.3.

7.2 FGLM development

The motivation in this section is to extent the FLM methodology, to FGLMs. FGLMs are models that model a response variable that belongs to the exponential family of distributions, as a function of profile factors; see Marx and Eilers (1999) and Morris (2015). A way of modelling the relationship between the response that follows an EFD and J profile factors, is a FGLM given by,

$$\begin{aligned} y_i &\sim \text{EFD}(\mu_i, \phi / w_i) \\ g(\mu_i) &= \int_0^{\mathcal{T}} \boldsymbol{f}^T(\boldsymbol{x}_i(t))\, \boldsymbol{\beta}(t)\, dt = \eta_i, \quad i = 1, 2, \ldots, n, \end{aligned} \tag{7.6}$$

with $\boldsymbol{f}^T(\boldsymbol{x}_i(t))$ and $\boldsymbol{\beta}(t)$ as in model (3.7), μ_i the mean of the response y_i, ϕ the dispersion parameter with w_i the weights of the dispersion parameter, and i the i^{th} run of the experiment. In matrix form, the additive FGLM is an extension to the model in (5.1) and it is given by,

$$\begin{aligned} \boldsymbol{y} &\sim \text{EFD}(\boldsymbol{\mu}, \phi \boldsymbol{A}) \\ g(\boldsymbol{\mu}) &= \int_0^{\mathcal{T}} \boldsymbol{f}^T(\boldsymbol{X}(t))\, \boldsymbol{\beta}(t)\, dt = \boldsymbol{\eta}, \end{aligned} \tag{7.7}$$

with $\boldsymbol{f}^T(\boldsymbol{X}(t))$ and $\boldsymbol{\beta}(t)$ as in model (5.1), $\boldsymbol{\mu}$ the $n \times 1$ vector containing the mean of the responses, ϕ the dispersion parameter, and $\boldsymbol{A}$ the $n \times 1$ vector of the weights of the dispersion parameter with the i^{th} entry being $1/w_i, i = 1, 2, \ldots, n$.

It is straightforward to notice that the linear predictor in models (7.6) and (7.7) is identical to the right-hand side of models (3.7) and (5.1). Thus, the functional parameters and the profile factors can be represented as a linear combination of basis functions, exactly as the basis expansions in (3.11) and (3.12). A substitution of the basis function expansion into the FGLM from (7.7) updates the linear predictor to,

$$g(\boldsymbol{\mu}) = \boldsymbol{Z\theta} \tag{7.8}$$

with $\boldsymbol{Z}$ the $n \times \sum_{q=1}^{Q} n_{\beta,q}$ model matrix and $\boldsymbol{\theta}$ the $\sum_{q=1}^{Q} n_{\beta,q} \times 1$ vector of unknown parameters, as defined in (5.10).

Remember from Chapter 5 that, the model matrix $\boldsymbol{Z}$ is the solution to an integral with respect to time, of the product of the function $\boldsymbol{f}^T(\boldsymbol{X}(t))$ and the vector of basis functions $\boldsymbol{b}^T(t)$,

$$\int_0^{\mathcal{T}} \boldsymbol{f}^T(\boldsymbol{X}(t))\, \boldsymbol{b}^T(t)\, dt.$$

As before, $\boldsymbol{Z}$ is partitioned in Q column blocks, with the q^{th} column block a $n \times n_{\beta,q}$ matrix $\boldsymbol{Z}_{\bullet q}$, which is the solution to an integral of the form,

$$\begin{aligned}
\boldsymbol{Z}_{\bullet q} &= \int_0^{\mathcal{T}} \boldsymbol{f}_q(\boldsymbol{X}(t))\, \boldsymbol{b}_q^T(t)\, dt \\
&= \int_0^{\mathcal{T}} \boldsymbol{f}_q\left(\boldsymbol{x}_{\bullet 1}(t) \quad \boldsymbol{x}_{\bullet 2}(t) \quad \cdots \quad \boldsymbol{x}_{\bullet J}(t)\right)\, \boldsymbol{b}_q^T(t)\, dt \\
&= \int_0^{\mathcal{T}} \boldsymbol{f}_q\left(\boldsymbol{\Gamma}_1\boldsymbol{c}_1(t) \quad \boldsymbol{\Gamma}_2\boldsymbol{c}_2(t) \quad \cdots \quad \boldsymbol{\Gamma}_J\boldsymbol{c}_J(t)\right)\, \boldsymbol{b}_q^T(t)\, dt, \\
&\quad q = 1, 2, \ldots, Q,
\end{aligned}$$

as in (5.12). The form of the integrals depends on the specification of the function of the profile factors $\boldsymbol{f}$, i.e., main effects, higher order polynomials and interactions. Finally, $\boldsymbol{b}_q(t)$ is the vector of basis functions from the basis expansion of the q^{th} functional parameter, as before. For details on the expansion of each partition of the model matrix refer to Chapter 5.

7.3 Pseudo-Bayesian A- and D- optimal designs

The FGLM is an extension to the GLM discussed in Section 7.1 with model matrix $\boldsymbol{Z}$ instead of $\boldsymbol{F}$ and unknown parameters $\boldsymbol{\theta}$ instead of $\boldsymbol{\beta}$. Thus, equations for the score functions, Fisher information matrix, variance of the parameter estimates, and hence the A- and D- optimal objective functions are extensions to the equations in Section 7.1.

Following the equations from the GLM case, the score functions for the FGLM become,

$$U(\boldsymbol{\theta}, \boldsymbol{\Gamma}) = \boldsymbol{Z}^T\boldsymbol{\Pi}(\boldsymbol{y} - \boldsymbol{\mu}) \tag{7.9}$$

which depends on $\boldsymbol{\theta}$ through the systematic component and the mean and with $\boldsymbol{\Pi}$ being an $n \times n$ diagonal matrix with entries,

$$\pi_{ii} = \frac{1}{g'(\mu_i)} \frac{1}{\text{Var}(y_i)}, \quad i = 1, 2, \dots, n.$$

Similarly, following the derivation from (7.2), the Fisher information matrix for the FGLM becomes,

$$\mathcal{I}(\boldsymbol{\theta}, \boldsymbol{\Gamma}) = \boldsymbol{Z}^T \boldsymbol{W} \boldsymbol{Z} \tag{7.10}$$

with $\boldsymbol{W}$ the $n \times n$ diagonal matrix with entries,

$$w_{ii} = \frac{1}{g'(\mu_i)^2} \frac{1}{\text{Var}(y_i)}, \quad i = 1, 2, \dots, n.$$

For GLMs, and thus for FGLMs, identifying optimal designs is not straightforward. This is because the Fisher information matrix is a function of the unknown parameters $\boldsymbol{\theta}$. As a consequence, the design also depends on the unknown parameters. For this reason, in order to identify optimal experimental designs for GLMs, FGLMs, or generally for models for which the information matrix depends on the unknown parameters, prior information of the model parameters are required.

This problem can be approached in different ways, including locally optimal designs, sequential designs, maximin designs, and pseudo-Bayesian optimal designs; see Section 2.5. In this chapter, the approach that is followed to incorporate the prior information into the model and define the optimality objective functions is the pseudo-Bayesian approach; see Chaloner and Verdinelli (1995), Overstall and Woods (2017), citetwoods2017, and Section 2.5. Following the results from Section 2.5, the pseudo-Bayesian A- and D- optimality objective functions are defined as,

$$\begin{aligned}\Psi_A(\boldsymbol{\Gamma}) &= \mathbb{E}_{\boldsymbol{\theta}}\Big\{\text{tr}\big[\mathcal{I}(\boldsymbol{\theta}, \boldsymbol{\Gamma})^{-1}\big]\Big\} \\ &= \int_{\boldsymbol{\Theta}} \text{tr}\big[\mathcal{I}(\boldsymbol{\theta}, \boldsymbol{\Gamma})^{-1}\big] \pi(\boldsymbol{\theta}) \, d\boldsymbol{\theta} \\ &= \int_{\boldsymbol{\Theta}} \text{tr}\big[(\boldsymbol{Z}^T \boldsymbol{W} \boldsymbol{Z})^{-1}\big] \pi(\boldsymbol{\theta}) \, d\boldsymbol{\theta},\end{aligned} \tag{7.11}$$

$$\begin{aligned}\Psi_D(\boldsymbol{\Gamma}) &= \mathbb{E}_{\boldsymbol{\theta}}\Big\{\text{det}\big[(\mathcal{I}(\boldsymbol{\theta}, \boldsymbol{\Gamma})\big]^{-1/p}\Big\} \\ &= \int_{\boldsymbol{\Theta}} \text{det}\big[(\mathcal{I}(\boldsymbol{\theta}, \boldsymbol{\Gamma})\big]^{-1/p} \pi(\boldsymbol{\theta}) \, d\boldsymbol{\theta} \\ &= \int_{\boldsymbol{\Theta}} \text{det}\big[\boldsymbol{Z}^T \boldsymbol{W} \boldsymbol{Z}\big]^{-1/p} \pi(\boldsymbol{\theta}) \, d\boldsymbol{\theta} \\ &= \exp\Big\{ \int_{\boldsymbol{\Theta}} -\frac{1}{p} \log\big(\text{det}\big[\boldsymbol{Z}^T \boldsymbol{W} \boldsymbol{Z}\big]\big) \pi(\boldsymbol{\theta}) \, d\boldsymbol{\theta}\Big\},\end{aligned} \tag{7.12}$$

with $p = \sum_{q=1}^{Q} n_{\beta,q}$ the total number of basis functions of the functional parameters. It is usually more convenient, and in general advisable, to integrate the logarithm of the determinant; see Atkinson et al. (2007, Chapter 11). Hence, designs $\boldsymbol{\Gamma}^* \in \mathcal{X}$ that minimise the pseudo-Bayesian A- and D- optimality objective functions,

$$\Psi_A(\boldsymbol{\Gamma}^*) = \min_{\boldsymbol{\Gamma}\in\mathcal{X}} \int_{\boldsymbol{\Theta}} \operatorname{tr}\left[(\boldsymbol{Z}^T\boldsymbol{W}\boldsymbol{Z})^{-1}\right]\pi(\boldsymbol{\theta})\, d\boldsymbol{\theta}, \tag{7.13}$$

$$\Psi_D(\boldsymbol{\Gamma}^*) = \min_{\boldsymbol{\Gamma}\in\mathcal{X}} \exp\left\{\int_{\boldsymbol{\Theta}} -\frac{1}{p}\log\left(\det\left[\boldsymbol{Z}^T\boldsymbol{W}\boldsymbol{Z}\right]\right)\pi(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right\}, \tag{7.14}$$

are known as pseudo-Bayesian A- and D- optimal designs, respectively; see Chaloner and Verdinelli (1995), Woods et al. (2006), and Van De Ven and Woods (2014).

The challenge in identifying pseudo-Bayesian optimal designs is that the expectation of the objective functions with respect to the prior distribution, results from analytically intractable, and usually high-dimensional integrals. To overcome this challenge, such objective functions must be approximated numerically. Suitable numerical integration methods to approximate the intractable integrals, are discussed in the next section.

7.4 Approximation methods for intractable integrals

Up to this point, optimal experimental designs were identified from analytically tractable objective functions. For pseudo-Bayesian optimal designs, the expectation of the objective functions with respect to the prior distribution often result from analytically intractable and high-dimensional integrals of the form in (2.29).

In this scenario, the integral of the expectation of the objective functions with respect to the prior distribution is approximated numerically. Numerical integration refers to methods that compute the value of the integrable function at a finite number of points. Thus, a higher number of points increases the approximation accuracy; see Gelman et al. (2013, Chapter 10). The methods discussed and applied in this chapter are: Monte Carlo approximation which is a stochastic method; and quadrature approximation which is a deterministic method.

7.4.1 Monte Carlo approximation

Monte Carlo approximation is a stochastic method that evaluates numerically the expectation of a general function, as in in (2.29). It uses random samples, drawn from the desired distribution; see Caflisch (1998), in this case the prior distribution of the parameters, $\pi(\boldsymbol{\theta})$. After generating a reasonable number of random samples B from the prior distribution, the expectation of the objective function is approximated as,

$$\Psi(\boldsymbol{\Gamma}) = \mathbb{E}_{\boldsymbol{\theta}}\Big\{\psi(\boldsymbol{\theta},\boldsymbol{\Gamma})\Big\} = \int_{\boldsymbol{\Theta}} \psi(\boldsymbol{\theta},\boldsymbol{\Gamma})\ \pi(\boldsymbol{\theta})\ d\boldsymbol{\theta} \approx \frac{1}{B}\sum_{b=1}^{B} \psi(\boldsymbol{\theta_b},\boldsymbol{\Gamma}), \tag{7.15}$$

with $\psi(\boldsymbol{\theta_b},\boldsymbol{\Gamma})$ the objective function evaluated at the parameter random sample $\boldsymbol{\theta_b}$ for $b = 1, 2, \ldots, B$. According to the law of large numbers that is discussed by Lapeyre (2007), as the number of random samples becomes larger, the approximation gets closer to the actual expectation,

$$\frac{1}{B}\sum_{b=1}^{B} \psi(\boldsymbol{\theta_b},\boldsymbol{\Gamma}) \rightarrow \mathbb{E}_{\boldsymbol{\theta}}\Big\{\psi(\boldsymbol{\theta},\boldsymbol{\Gamma})\Big\}, \quad \text{as } B \rightarrow \infty.$$

7.4.2 Quadrature approximation

Quadrature approximation is a weighted version of the approximation in (7.15), with the integral evaluated at selection points. The approximation is defined as,

$$\Psi(\boldsymbol{\Gamma}) = \mathbb{E}_{\boldsymbol{\theta}}\Big\{\psi(\boldsymbol{\theta},\boldsymbol{\Gamma})\Big\} = \int_{\boldsymbol{\Theta}} \psi(\boldsymbol{\theta},\boldsymbol{\Gamma})\ \pi(\boldsymbol{\theta})\ d\boldsymbol{\theta} \approx \sum_{b=1}^{B} \omega_b\ \psi(\boldsymbol{\theta_b},\boldsymbol{\Gamma}), \tag{7.16}$$

with $\boldsymbol{\theta_b}$ and ω_b for $b = 1, 2, \ldots, B$, being abscissas and weights, respectively; see Gotwalt et al. (2009). As discussed in Gelman et al. (2013, Chapter 10), each weight value ω_b represents the space volume that is represented by the corresponding parameter point θ_b.

In literature, different quadrature rules exist, that tackle both bounded and unbounded integrals. In addition, depending on the form and the characteristics of the integrand, there may be a specific quadrature rule that is more suitable and hence, more likely to provide a good approximation. For instance, when the prior distribution is assumed to follow a normal distribution, the Gauss-Hermite quadrature rule is followed, and when the prior is assumed to be uniformly distributed, the Gauss-Legendre quadrature rule is followed. Both the Gauss-Hermite and the Gauss-Legendre quadrature rules are for one-dimensional integrals. However, the rules can be extended to multivariate integrals using Cartesian product grids or sparse grids. These are formulated in the **mvQuad** package (Weiser, 2016), in the R software.

Gauss-Hermite is a quadrature rule with infinite domain, that is based on Hermite polynomials; see Salzer et al. (1952). For a general function $\tilde{\psi}(\theta,\boldsymbol{\Gamma})$, it approximates integrals of the form,

$$\int_{-\infty}^{\infty} \exp(-\theta^T\theta)\ \tilde{\psi}(\theta,\boldsymbol{\Gamma})\ d\theta.$$

By definition, the approximation of the integrals of the above form is discussed in Liu and Pierce (1994) and it is given by,

$$\int_{-\infty}^{\infty} \exp(-\theta^T\theta)\, \tilde{\psi}(\theta, \mathbf{\Gamma})\, d\theta \approx \sum_{b=1}^{B} \omega_b\, \tilde{\psi}(\theta_b, \mathbf{\Gamma}), \tag{7.17}$$

with B the number of quadrature points, θ_b the abscissas, and ω_b the weights. Thus, to approximate the required integral in (2.29), the approximation takes the form,

$$\begin{aligned} \int_{-\infty}^{\infty} \exp(-\theta^T\theta)\, \tilde{\psi}(\theta, \mathbf{\Gamma})\, d\theta &= \int_{-\infty}^{\infty} \exp(-\theta^T\theta)\, \{\exp(\theta^T\theta)\, \psi(\theta, \mathbf{\Gamma})\pi(\theta)\}\, d\theta \\ &\approx \sum_{b=1}^{B} \omega_b\, \exp(\theta_b^T\theta_b)\, \psi(\theta_b, \mathbf{\Gamma})\, \pi(\theta_b), \end{aligned} \tag{7.18}$$

with $\tilde{\psi}(\theta, \mathbf{\Gamma}) = \exp(\theta^T\theta)\, \psi(\theta, \mathbf{\Gamma})\, \pi(\theta)$, and a normal prior distribution with mean μ_0 and variance Σ_0, such that $\pi(\theta) \sim N(\mu_0, \Sigma_0)$. Depending on the choice of parameters for the prior, the abscissas and the weights are rescaled. The Gauss-Hermite quadrature rule is formulated in the **mvQuad** package (Weiser, 2016), in the R software.

Gauss-Legendre is a quadrature rule with finite domain, that is based on Legendre polynomials; see Abramowitz and Stegun (1964) and Lether (1978). According to Hale and Townsend (2013), Gauss-Legendre is the most commonly used rule. For a general function $\tilde{\psi}(\boldsymbol{\theta}, \mathbf{\Gamma})$, it approximates integrals over the interval $[\boldsymbol{l}, \boldsymbol{u}]$ of the form,

$$\int_{l}^{u} \tilde{\psi}(\theta, \mathbf{\Gamma})\, d\theta,$$

for values l and u representing the lower and upper intervals of the parameter. The approximation of integrals of the form above is given by Hale and Townsend (2013) such that,

$$\int_{l}^{u} \tilde{\psi}(\theta, \mathbf{\Gamma})\, d\theta \approx \sum_{b=1}^{B} \omega_b\, \tilde{\psi}(\theta_b, \mathbf{\Gamma}), \tag{7.19}$$

with B the number of quadrature points, θ_b the abscissas, and ω_b the weights, as before. Following the latter definition, the usually intractable integral in (2.29), is approximated as,

$$\begin{aligned} \int_{l}^{u} \tilde{\psi}(\theta, \mathbf{\Gamma})\, d\theta &= \int_{l}^{u} \psi(\theta, \mathbf{\Gamma})\pi(\theta)\, d\theta \\ &\approx \sum_{b=1}^{B} \omega_b\, \psi(\theta_b, \mathbf{\Gamma})\, \pi(\theta_b), \end{aligned} \tag{7.20}$$

with $\tilde{\psi}(\theta, \mathbf{\Gamma}) = \psi(\theta, \mathbf{\Gamma})\, \pi(\theta)$, and a uniform prior distribution with parameters l and u representing the lower and upper intervals, such that $\pi(\theta) \sim U(l, u)$. According to the interval values, the domain of the integral changes, and the abscissas and the weights are rescaled. The Gauss-Legendre quadrature rule is also formulated in the **mvQuad** package (Weiser, 2016).

In conclusion, high dimensional integrals require a large number of quadrature points to accurately approximate the solution. With that being said, the computational time increases and hence, finding optimal experimental designs may be computationally expensive.

7.5 FGLM roughness penalty approach

In this section, the aim is to expand the roughness penalty approach discussed in Section 5.5, to the FGLMs. The parameter estimator and an asymptotic approximation to the variance-covariance matrix of the parameter estimators are derived using the score equations and the Fisher information matrix; see Sections 7.1 and 7.3. To incorporate a roughness penalty, instead of maximising the log-likelihood, the penalised log-likelihood is maximised. The penalised log-likelihood $p(\boldsymbol{\theta})$ is defined as,

$$p(\boldsymbol{\theta}) = l(\boldsymbol{\theta}) - \Lambda \boldsymbol{\theta}^T \boldsymbol{R}_0 \boldsymbol{\theta}, \tag{7.21}$$

with $l(\boldsymbol{\theta})$ the log-likelihood, $\Lambda > 0$ the scalar smoothing parameter, and $\boldsymbol{R}_0$ the matrix that is representing the roughness penalties, which is a $\sum_{q=1}^{Q} n_{\beta,q} \times \sum_{q=1}^{Q} n_{\beta,q}$ block diagonal matrix; as in (5.32) in Section 5.5. Thus, the estimating equations are updated to,

$$U(\boldsymbol{\theta}, \boldsymbol{\Gamma}) = \boldsymbol{Z}^T \boldsymbol{\Pi}(\boldsymbol{y} - \boldsymbol{\mu}) - 2\Lambda \boldsymbol{R}_0 \boldsymbol{\theta}, \tag{7.22}$$

with $\boldsymbol{\Pi}$ as in (7.9). After that, the information matrix is updated to,

$$\mathcal{I}(\boldsymbol{\theta}, \boldsymbol{\Gamma}) = \boldsymbol{Z}^T \boldsymbol{W} \boldsymbol{Z} + \Lambda \boldsymbol{R}_0 \tag{7.23}$$

with $\boldsymbol{W}$ as in (7.10). The connection with the Bayesian approach is that $\boldsymbol{V}^{-1} = \Lambda \boldsymbol{R}_0$, which comes from maximising the estimating equation in (7.21) and from the posterior mode when the prior mean is centered around zero.

7.6 Functional logistic model involving one profile factor and step function basis

In this example, a functional logistic model depending on the intercept, and a single profile factor is examined. Thus, the functional of the profile factors $\boldsymbol{f}$ is,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \begin{pmatrix} 1 & x_{i1}(t) \end{pmatrix}, \tag{7.24}$$

with $x_{i1}(t)$ the single profile factor at the i^{th} run of the experiment. The model in (7.6) is simplified such that,

$$\begin{aligned} &y_i \sim \text{Bernoulli}(\mu_i) \\ &\eta_i = \beta_1 + \int_0^T \beta_2(t) x_{i1}(t)\, dt, \quad i = 1, 2, \ldots, n,\ t \in [0,1],\ -1 \le x_{i1}(t) \le 1, \end{aligned} \tag{7.25}$$

with link function $g(\mu_i) = \log\left(\frac{\mu_i}{1-\mu_i}\right)$.

It is assumed that control of the profile factor is represented via a step function basis, i.e., BS basis of degree $d = 0$. The basis for the functional parameters is assumed to be linear, i.e.,

$$\boldsymbol{\theta}^T = \begin{pmatrix} \theta_1 & \boldsymbol{\theta}_2 \end{pmatrix} = \begin{pmatrix} \theta_1 & \theta_{21} & \theta_{22} \end{pmatrix}, \tag{7.26}$$

and quadratic, i.e.,

$$\boldsymbol{\theta}^T = \begin{pmatrix} \theta_1 & \boldsymbol{\theta}_2 \end{pmatrix} = \begin{pmatrix} \theta_1 & \theta_{21} & \theta_{22} & \theta_{23} \end{pmatrix}. \tag{7.27}$$

The objective of the experiment is represented through the A-optimality objective function from (7.4). The method of approximation of the integral (7.11) is quadrature, for normal prior distribution with mean 0 and variance 1,

$$\theta_1 \sim N(0,1), \quad \theta_{21} \sim N(0,1), \quad \theta_{22} \sim N(0,1), \quad \theta_{23} \sim N(0,1). \tag{7.28}$$

As for the FLM, a sensitivity study is carried out to investigate how the number of runs, the number of basis functions of the profile factor, and the parameter basis affect the final designs. The choice of runs is $n \in \{4, 8, 12\}$, and the choice of the number of basis functions for the profile factor is $n_{x,1} \in \{2, 3, 4, 8, 16, 100\}$. The constraints from Section 5.8 still apply. In every case, 1000 random starts are investigated.

The study exhibits similar results to the equivalent FLM example in Section 6.1. Firstly, for the linear basis of the parameters results are available in Table 7.1. The optimal designs identified, have at most two changes in the step function. However, most designs have at most a single change in the step functions. To be precise, two changes in the step function occur only in four run final designs. For $n = 8, 12$, every optimal design found has a single change in the step function; for instance see Figure 7.1. Function patterns are similar when increasing the number of runs, i.e., same functions with more repetitions, or translations of the existing functions. A selection of nested knots is chosen except when $n_{x,1} = 3$. The objective value of the pseudo-Bayesian A-optimal design under $n_{x,1} = 3$ and $n = 4$, is lower than the objective values of $n_{x,1} = 2, 4$. This is the only case this happens; see Table 7.1. For $n = 8, 12$, the objective values under $n_{x,1} = 3$ are higher than $n_{x,1} = 2, 4$, which is the case for the FLM example also.

Additionally, as the number of basis functions of the profile factor increases, the optimality values drop. However, the drop in the objective values becomes insignificant from a relatively small $n_{x,1}$ onwards. Thus, it can be argued that high number of basis functions for the profile factor is not needed. For instance, comparing the designs for $n_{x,1} = 2, 3, 4, 8, 16$ to $n_{x,1} = 100$, since it achieves the minimum value for every n, it is clear to see that their performance does not differ significantly. This is also noticeable from the A-efficiencies, which are almost always higher than 90%. This means that designs found for small number of basis functions are efficient and comparable; see Table 7.1.

	$n = 4$		$n = 8$		$n = 12$	
$n_{x,1}$	A-opt	A-eff	A-opt	A-eff	A-opt	A-eff
2	50.026	0.919	21.884	0.980	14.447	0.966
3	48.050	0.957	23.641	0.991	15.438	0.904
4	49.512	0.929	21.805	0.984	14.234	0.981
8	46.450	0.990	21.645	0.991	14.050	0.994
16	46.156	0.996	21.566	0.995	13.974	0.999
100	45.983	1.000	21.457	1.000	13.960	1.000

TABLE 7.1: Pseudo-Bayesian A-optimality values and A-efficiency values with $n = 4, 8, 12$ for the linear basis for $\beta_2(t)$ for the functional logistic model.

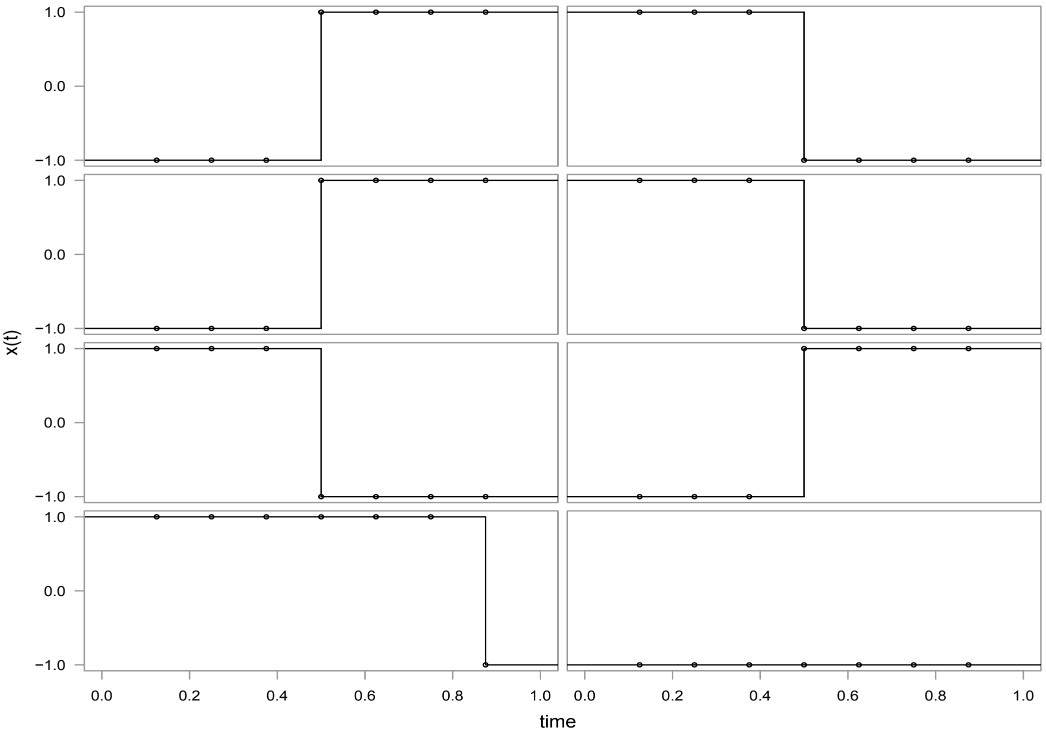


FIGURE 7.1: Eight run pseudo-Bayesian A-optimal design for $n_{x,1} = 8$ for $x_{\cdot 1}(t)$, and linear basis for $\beta_2(t)$ for the functional logistic model.

Results for the quadratic basis, are also similar to the results found for the FLM. The objective values of the optimal designs identified are in Table 7.2. The step functions achieve at most three changes, with most of the optimal functions changing twice. It is interesting to notice that in $n = 12$ run designs, the step functions change twice, irrespective the number of basis functions; see Figure 7.2. Moreover, the quadratic parameter basis is more sensitive to changes in the number of basis functions. In particular, increasing the number of basis functions causes a more significant change in the performance of the designs, than it does for the linear case; see Table 7.2.

	$n = 4$		$n = 8$		$n = 12$	
$n_{x,1}$	A-opt	A-eff	A-opt	A-eff	A-opt	A-eff
3	2105.674	0.527	1018.511	0.510	678.416	0.500
4	1323.035	0.839	555.925	0.934	365.898	0.926
8	1172.411	0.947	532.386	0.975	349.056	0.970
16	1121.446	0.990	522.327	0.994	340.889	0.993
100	1110.486	1.000	519.025	1.000	338.650	1.000

TABLE 7.2: Pseudo-Bayesian A-optimality values and A-efficiency values with $n = 4, 8, 12$ for the quadratic basis for $\beta_2(t)$ for the functional logistic model.

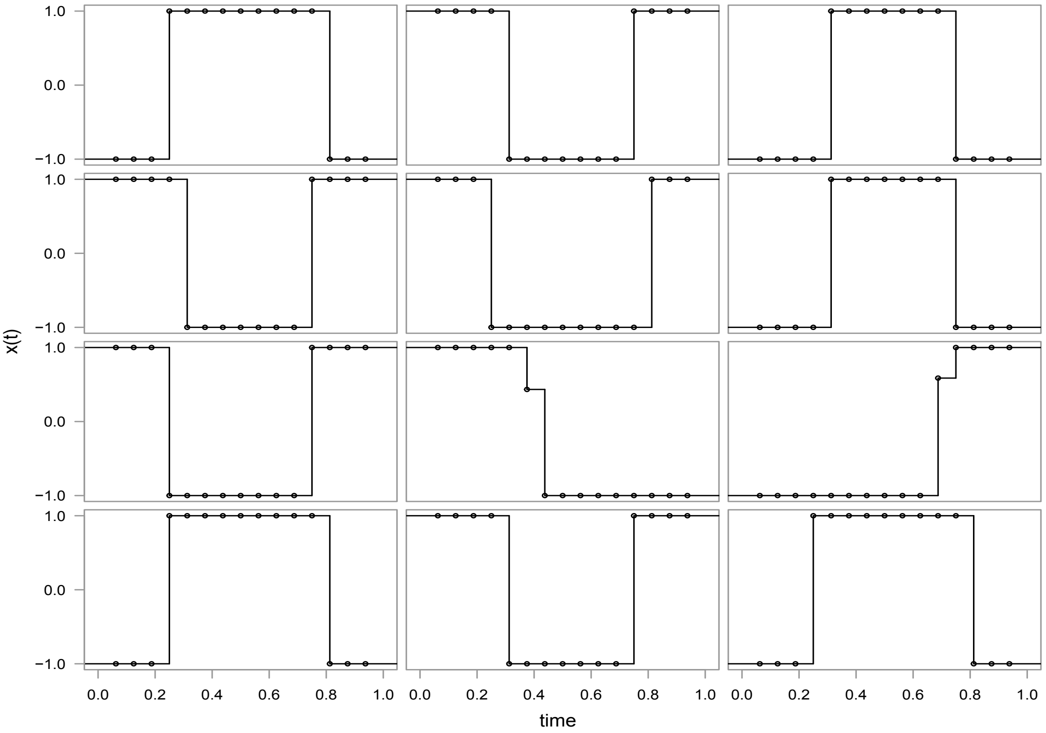


FIGURE 7.2: 12 run pseudo-Bayesian A-optimal design for $n_{x,1} = 16$ for $x_{\cdot 1}(t)$, and quadratic basis for $\beta_2(t)$ for the functional logistic model.

7.7 Functional logistic model involving one profile factor and two scalar factors

In this section, the aim is to expand the example from Section 7.6, by adding two scalar factors. For the scalar factors, main and main and quadratic effects are considered; as in the FLM example in Section 6.2. Scalar factors are a special case for which a single constant basis function is used; see Section 3.1. For only main effects, the functional of the profile factors $\boldsymbol{f}$ is,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \begin{pmatrix}1 & x_{i1}(t) & x_{i2} & x_{i3}\end{pmatrix}, \tag{7.29}$$

with $x_{i1}(t)$ the single profile factor at the i^{th} run of the experiment, and x_{i2} and x_{i3} the scalar factors at the i^{th} run of the experiment. For the addition of the quadratic effects, the functional of the profile factors $\boldsymbol{f}$ is,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \begin{pmatrix}1 & x_{i1}(t) & x_{i2} & x_{i3} & x_{i2}^2 & x_{i3}^2\end{pmatrix}, \tag{7.30}$$

with x_{i2}^2 and x_{i3}^2 the quadratic effects of the scalar factors at the i^{th} run of the experiment. Under this set up, the functional logistic model takes the form,

$$\begin{aligned}
&y_i \sim \text{Bernoulli}(\mu_i)\\
&\eta_i = \beta_1 + \int_0^{\mathcal{T}} \beta_2(t)x_{i1}(t)\,dt + \boldsymbol{g}^T(\boldsymbol{x}_i)\boldsymbol{\nu} + \epsilon_i,\\
&\qquad i = 1,2,\dots,n,\ t \in [0,1],\ -1 \le \boldsymbol{x}_i(t) \le 1,
\end{aligned} \tag{7.31}$$

with link function $g(\mu_i) = \log\left(\frac{\mu_i}{1-\mu_i}\right)$, and

$$\boldsymbol{g}^T(\boldsymbol{x}_i)\boldsymbol{\nu} = \begin{cases}\beta_3x_{i2} + \beta_4x_{i3} & \text{Case 1}\\ \beta_3x_{i2} + \beta_4x_{i3} + \beta_5x_{i2}^2 + \beta_6x_{i3}^2 & \text{Case 2.}\end{cases}$$

For the single functional parameter, the linear basis is considered, i.e., $n_{\beta,2} = 2$. Thus, for the Case 1 and Case 2 model, the coefficients are,

$$\boldsymbol{\theta}^T = \begin{pmatrix}\theta_1 & \boldsymbol{\theta}_2 & \theta_3 & \theta_4\end{pmatrix} = \begin{pmatrix}\theta_1 & \theta_{21} & \theta_{22} & \theta_3 & \theta_4\end{pmatrix},$$

$$\boldsymbol{\theta}^T = \begin{pmatrix}\theta_1 & \boldsymbol{\theta}_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6\end{pmatrix} = \begin{pmatrix}\theta_1 & \theta_{21} & \theta_{22} & \theta_3 & \theta_4 & \theta_5 & \theta_6\end{pmatrix},$$

respectively. Control of the profile factor is represented via a step function basis, i.e., BS basis of degree $d = 0$ and the number of basis functions is $n_{x,1} = 4$ from the choice of interior knots $\lambda = (0.25, 0.50, 0.75)$. The number of basis functions is kept low on purpose, to avoid high computational cost due to the complexity of the model. The objective of the experiment is represented through the pseudo-Bayesian A-optimality function from (7.11). Due to high dimensionality, quadrature approximation is inefficient to use. To achieve numerical stability, a large number of quadrature points is needed, causing a computational burden. Instead, a normal prior distribution is assumed, with mean equal to zero and variance equal to two,

$$\theta_1 \sim N(0,2), \quad \theta_{21} \sim N(0,2), \quad \theta_{22} \sim N(0,2), \quad \theta_3 \sim N(0,2),$$
$$\theta_4 \sim N(0,2), \quad \theta_5 \sim N(0,2), \quad \theta_6 \sim N(0,2),$$

from which 10000 points are generated, and the expectation is approximated using the Monte Carlo method. The number of runs is $n = 12$.

Although the settings for the single profile factor are identical to the example in Section 7.6, the final designs are not identical. In Sections 6.1 and 6.2, for the equivalent examples of the FLM, the final designs did not differ. This may be due to the use of the Monte Carlo approximation rather than a quadrature scheme. In Case 1, the optimal design found for the profile factor is at boundaries. In Case 2, the optimal design for the profile factor is at the boundaries with the exception of some points. Moreover, in Case 1, the step functions achieve at most one change, but for Case 2, the step functions achieve at most two changes. Meaning that, the functions of the profile factor are more complicated for Case 2 compared to Case 1; see Figure 7.3 and Figure 7.4, respectively.

The values of the scalar factors are not at the boundaries; see Table 7.3. Moreover, the columns of the scalar designs are not orthogonal, as it was the case for the columns of the scalar design in Section 6.2. The scalar values in Case 2 do not include center points equal to zero as in 6.2, but they include small values, close to zero, to allow the estimation of the curvature. The objective values for both cases, as well as evaluations of the designs under the opposite case, are given in Table 7.4. In Table 7.4, the values in brackets are the A-efficiency values. The optimal design found in Case 2 is more efficient than the design found in Case 1; see the A-efficiency values in Table 7.4. The full optimal designs are the combinations of the optimal functions of the profile factor in Figure 7.3 and Figure 7.4 with functions plotted by column, and the values of the scalar factors in Table 7.3.

Case 1			Case 2		
i	x_{i2}	x_{i3}	i	x_{i2}	x_{i3}
1	0.403	0.913	1	0.084	0.031
2	-0.919	0.527	2	0.113	0.094
3	0.438	-0.913	3	-0.435	1.000
4	-0.764	-0.754	4	-0.246	-0.067
5	0.218	0.982	5	0.398	1.000
6	0.946	0.104	6	-0.395	-1.000
7	0.541	-0.842	7	1.000	0.454
8	0.807	0.061	8	-0.229	-0.200
9	-0.861	0.587	9	-1.000	0.417
10	-0.393	-0.051	10	0.392	-1.000
11	-0.778	-0.747	11	-1.000	-0.458
12	0.050	0.006	12	1.000	-0.455

TABLE 7.3: Optimal designs for main effects (Case 1) and main & quadratic effects (Case 2) of the two scalar factors for the functional logistic model.

	A-opt	Evaluation of the final design under the opposite case
Case 1	25.316 (1.000)	25.894 (0.978)
Case 2	50.832 (1.000)	80.831 (0.629)

TABLE 7.4: A-optimality values of the Case 1 and Case 2 functional logistic model with one profile and two scalar factors, linear basis for $\beta_2(t)$, $n_{x,1} = 4$ and $n = 12$. Efficiency values in brackets of evaluations of the Case 1 and Case 2 optimal designs under both cases.

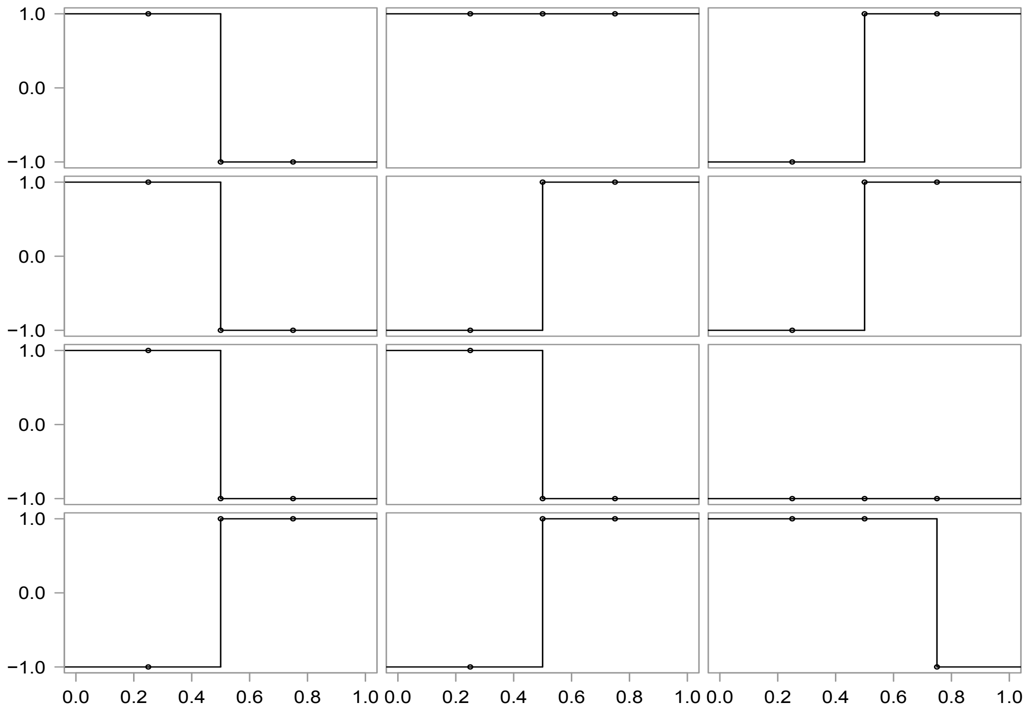


FIGURE 7.3: 12 run A-optimal design with $n_{x,1} = 4$ for $x_{\cdot 1}(t)$, linear basis for $\beta_2(t)$, and main effects for the two scalar factors, for the functional logistic model.

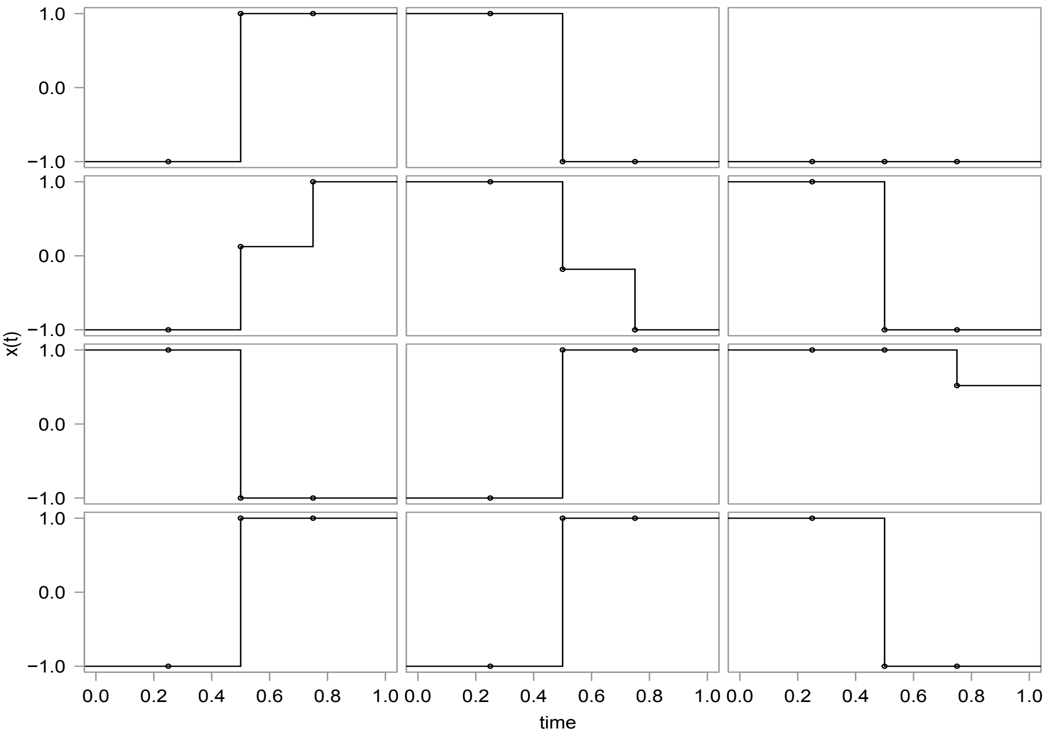


FIGURE 7.4: 12 run A-optimal design with $n_{x,1} = 4$ for $x_{\cdot 1}(t)$, linear basis for $\beta_2(t)$, and main and quadratic effects for the two scalar factors, for the functional logistic model.

7.8 Functional Poisson model involving a single profile factor with main and main and quadratic effect

In this section, the aim of the example developed, is to find D-optimal designs for a functional Poisson model, that depends on a single profile factor with main, and main and quadratic effect. The intercept is also included.

For the Poisson model with the main effect of the profile factor, the functional of the profile factors $\boldsymbol{f}$ is,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \begin{pmatrix} 1 & x_{i1}(t) \end{pmatrix}, \tag{7.32}$$

with $x_{i1}(t)$ the single profile factor at the i^{th} run of the experiment. Subsequently, the model from (7.6) takes the form,

$$\begin{aligned} & y_i \sim \text{Poisson}(\mu_i) \\ & \eta_i = \beta_1 + \int_0^{\mathcal{T}} x_{i1}(t)\beta_2(t)\, dt, \quad i = 1, 2, \ldots, n,\ t \in [0,1],\ -1 \le x_{i1}(t) \le 1, \end{aligned} \tag{7.33}$$

with link function $g(\mu_i) = \log(\mu_i)$. For the Poisson model with the main and the quadratic effect of the profile factor, the functional of the profile factors $\boldsymbol{f}$ is updated to,

$$\boldsymbol{f}^T(\boldsymbol{x}_i(t)) = \begin{pmatrix} 1 & x_{i1}(t) & x_{i1}^2(t) \end{pmatrix}, \tag{7.34}$$

and the model becomes,

$$\begin{aligned} & y_i \sim \text{Poisson}(\mu_i) \\ & \eta_i = \beta_1 + \int_0^{\mathcal{T}} x_{i1}(t)\beta_2(t)\, dt + \int_0^{\mathcal{T}} x_{i1}^2(t)\beta_3(t)\, dt, \\ & \qquad i = 1, 2, \ldots, n,\ t \in [0,1],\ -1 \le x_{i1}(t) \le 1. \end{aligned} \tag{7.35}$$

For the function of the profile factor, control is represented by a degree $d = 1$ BS basis with four equal interior knots $\boldsymbol{\lambda} = (0.20, 0.40, 0.60, 0.80)$. Thus, the total number of basis functions for the profile factor is $n_{x,1} = 6$. For the functional parameters, a BS basis of degree $d = 1$ and a single knot at $t = 0.5$ is considered, i.e., $n_{\beta,2} = n_{\beta,3} = 3$. Thus, for the model with the main effect the coefficients are,

$$\boldsymbol{\theta}^T = \begin{pmatrix} \theta_1 & \boldsymbol{\theta}_2 \end{pmatrix} = \begin{pmatrix} \theta_1 & \theta_{21} & \theta_{22} & \theta_{23} \end{pmatrix},$$

and for the model with the main and the quadratic effect the coefficients are,

$$\boldsymbol{\theta}^T = \begin{pmatrix} \theta_1 & \boldsymbol{\theta}_2 & \boldsymbol{\theta}_3 \end{pmatrix} = \begin{pmatrix} \theta_1 & \theta_{21} & \theta_{22} & \theta_{23} & \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix}.$$

The expectation is approximated using the Monte Carlo method, and normal prior with mean 0 and variance 2,

$$\theta_1 \sim N(0,2), \quad \theta_{21} \sim N(0,2), \quad \theta_{22} \sim N(0,2), \quad \theta_{23} \sim N(0,2),$$
$$\theta_{31} \sim N(0,2), \quad \theta_{32} \sim N(0,2), \quad \theta_{33} \sim N(0,2)$$

The number of runs is $n = 12$, and the objective is to identify pseudo-Bayesian D-optimal designs using the objective function in (7.12).

For the model with the main effect of the profile factor, the objective value of the final design is 1.924. The objective value of the final design after the addition of the quadratic term, is 7.283. The design found for the main effect, evaluated under the model with main and quadratic effects results in a huge objective value which is 100000. Thus, the D-efficiency value is close to zero. This is because the design with main effects is always at the boundaries, not allowing estimation of the curvature when quadratic effects are assumed. If the final design for the main effect is used as starting design on the model with the quadratic effect, the objective value is 7.710, which is very close to the optimal. An evaluation of the design for the model which includes the quadratic effect under the model with the main effect, gives a D-optimality value of 2.698, which is 71.3% D-efficient.

Moreover, the functions of the profile factors for the model with main effects, are always at the boundaries, with at most two ramp changes. Most of the functions are upward or downward one change ramp functions; see Figure 7.5. The values in red in the figure, represent the complexity value of each function. On average, the complexity value of the functions for the model with main effects is 8.000. The functions are less complicated after the addition of the quadratic effect, with the average complexity value being 3.858; see Figure 7.6. This is mostly because the functions do not get close to the boundaries.

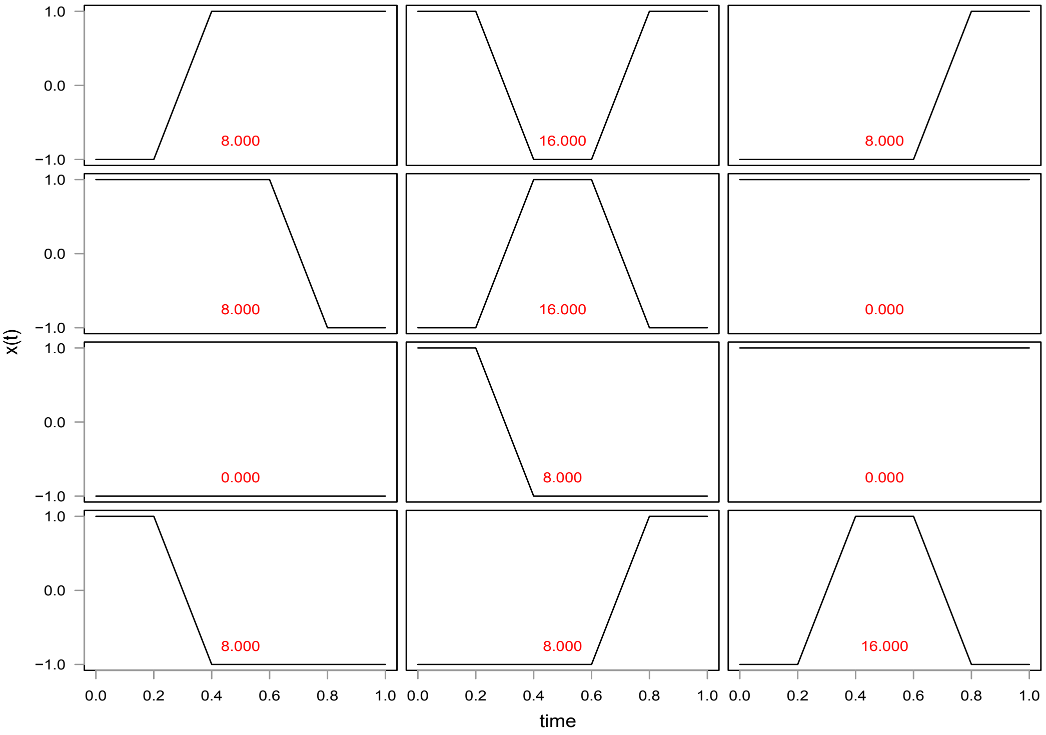


FIGURE 7.5: 12 run pseudo-Bayesian D-optimal design for one profile factor with BS degree $d = 1$ basis and $n_{x,1} = 6$ basis functions with only main effect considered, and BS basis of degree $d = 1$ for $\beta_2(t)$ for the functional Poisson model.

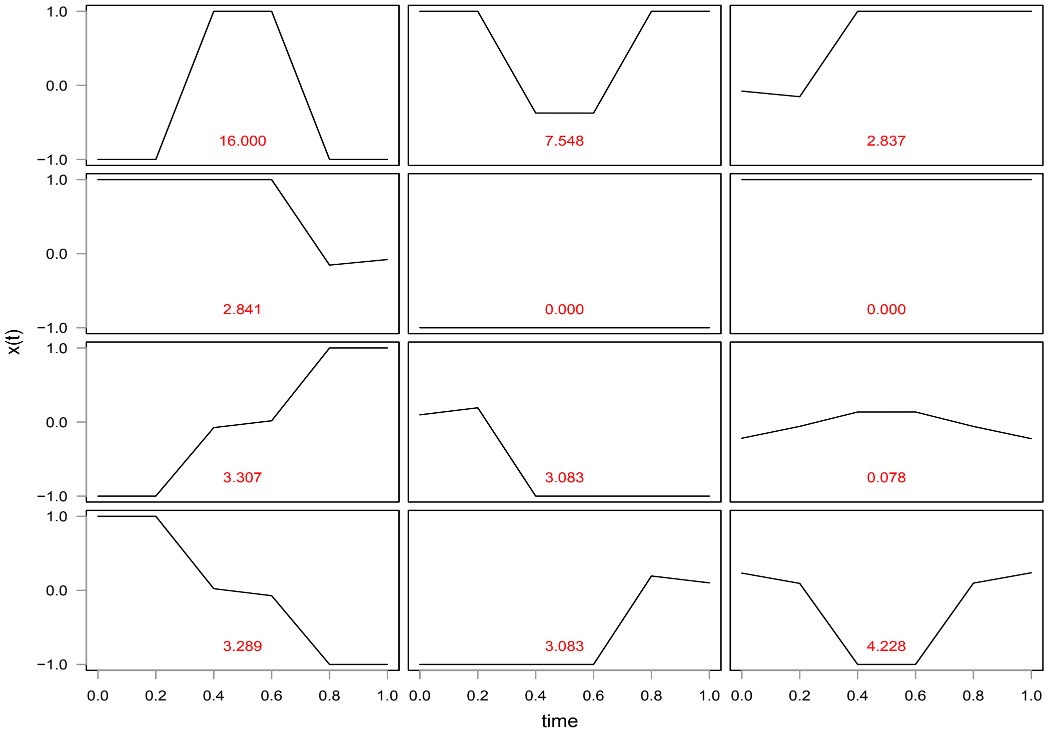


FIGURE 7.6: 12 run pseudo-Bayesian D-optimal design for one profile factor with BS degree $d = 1$ basis and $n_{x,1} = 6$ basis functions with main and main and quadratic effect considered, and BS basis of degree $d = 1$ for $\beta_2(t)$ and $\beta_3(t)$ for the functional Poisson model.

7.9 Bayesian approach on the functional logistic model involving one profile factor

The functional logistic model from 7.6, with exactly the same settings for the single profile factor is reconsidered. The addition in this example is that, the complexity of the functions is penalised, to identify Bayesian A-optimal designs, with the roughness penalties added into the information matrix; see Section 7.5. The prior choice for the variance is $\sigma^2 \sim IG(2,1)$. Remember that, a choice of linear basis for the parameters does not make sense, since the second derivatives in matrix $\boldsymbol{R}_p$ from Section 5.5 are zero. For this reason, quadratic parameters are considered. The choice of a quadratic basis for the functional parameters means that the matrix $\boldsymbol{R}_p$ has a non zero entry, i.e., $r_{33} = 4$.

The choice of smoothing parameter varies between $\Lambda \in \{0.01, 1, 10\}$. This is to investigate how the smoothing parameter affects the resulting Bayesian A-optimal designs, and also to compare with the designs derived under the frequentist approach in Section 7.6. The number of basis functions for the profile factor are $n_{x,1} \in \{3, 4, 6, 8, 15, 16\}$

and the choice of runs is $n \in \{4, 8, 12\}$. The Bayesian A-optimality values of the final designs, can be found in Table 7.5. It is clear to identify, that as the number of runs increases, the objective values drop.

The range of basis functions investigated, is wider, compared to the FLM example in Section 6.5. The reason is that, for $n = 4$ runs, the objective value when $n_{x,1} = 3$ is lower, compared to when $n_{x,1} = 4$. Based on earlier examples in this thesis, this is unexpected. To investigate this further, $n_{x,1} = 6, 15$ are added to the analysis, to form a sequence of nested knot selections for $t = 0.33$ and $t = 0.66$, rather than just for $t = 0.5$. In almost every case, the objective value is slightly lower for the choice of $n_{x,1} = 15$ compared to $n_{x,1} = 16$. Counter to other studies in this thesis, models nested with $t = 0.5$ as a knot do not always perform better. The differences in the objective values are mostly insignificant, but in some cases the functions of the profile factor achieve fewer changes in the step functions for $n_{x,1} = 15$; see Figure 7.7 and Figure 7.8.

In some cases, the number of runs affect significantly the complexity of the Bayesian A-optimal designs. This is noticeable in the number of changes in the step functions. For example, the step functions achieve at most two changes for $n = 8, \Lambda = 10$, and $n_{x,1} = 16$, but at most a single change for $n = 12, \Lambda = 10$, and $n_{x,1} = 16$; see Figure 7.7 and Figure 7.9.

The smoothing parameter Λ affects the resulting optimal designs in the same way as in Section 6.5. When the value of the smoothing parameter is close to zero, the designs are similar to the designs found under the frequentist approach with the quadratic parameter basis. However, as the value of Λ increases, the designs become similar to the designs found under the frequentist approach and the linear parameter basis. This is because the quadratic term is highly penalised.

$\Lambda = 0.01$			
n_x	$n = 4$	$n = 8$	$n = 12$
3	99.804	74.967	66.358
4	101.487	72.918	65.398
6	99.094	72.891	65.211
8	97.791	72.876	65.001
15	97.555	72.766	64.890
16	97.632	72.772	64.894
$\Lambda = 1$			
n_x	$n = 4$	$n = 8$	$n = 12$
3	49.084	24.435	16.061
4	50.687	22.643	14.964
6	48.594	22.490	14.811
8	47.488	22.483	14.782
15	47.180	22.304	14.717
16	47.202	22.411	14.717
$\Lambda = 10$			
n_x	$n = 4$	$n = 8$	$n = 12$
3	48.621	23.971	15.595
4	50.225	22.185	14.503
6	48.134	22.030	14.346
8	47.031	22.024	14.314
15	46.722	22.304	14.256
16	46.744	21.944	14.249

TABLE 7.5: Bayesian A-optimality values for $n \in \{4, 8, 12\}$, $n_{x,1} \in \{3, 4, 6, 8, 15, 16\}$, and smoothing parameters $\Lambda \in \{0.01, 1, 10\}$, for the quadratic basis of $\beta_2(t)$ for the functional logistic model.

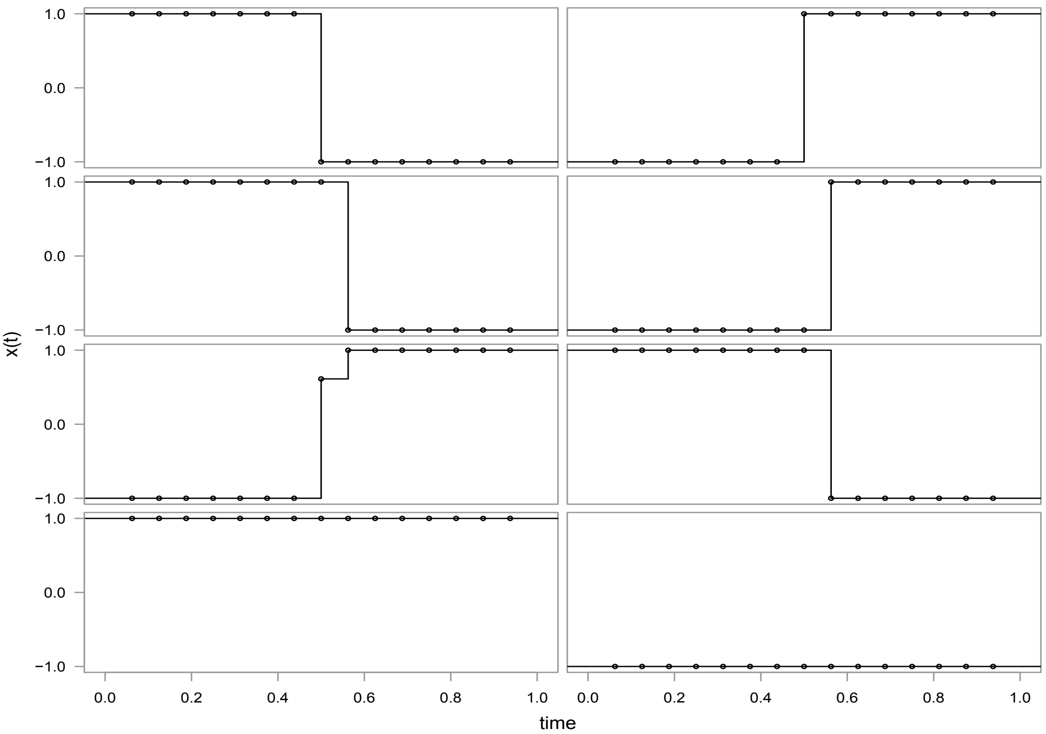


FIGURE 7.7: Eight run Bayesian A-optimal design for $n_{x,1} = 16, \Lambda = 10$ and quadratic basis for $\beta_2(t)$ for the functional logistic model.

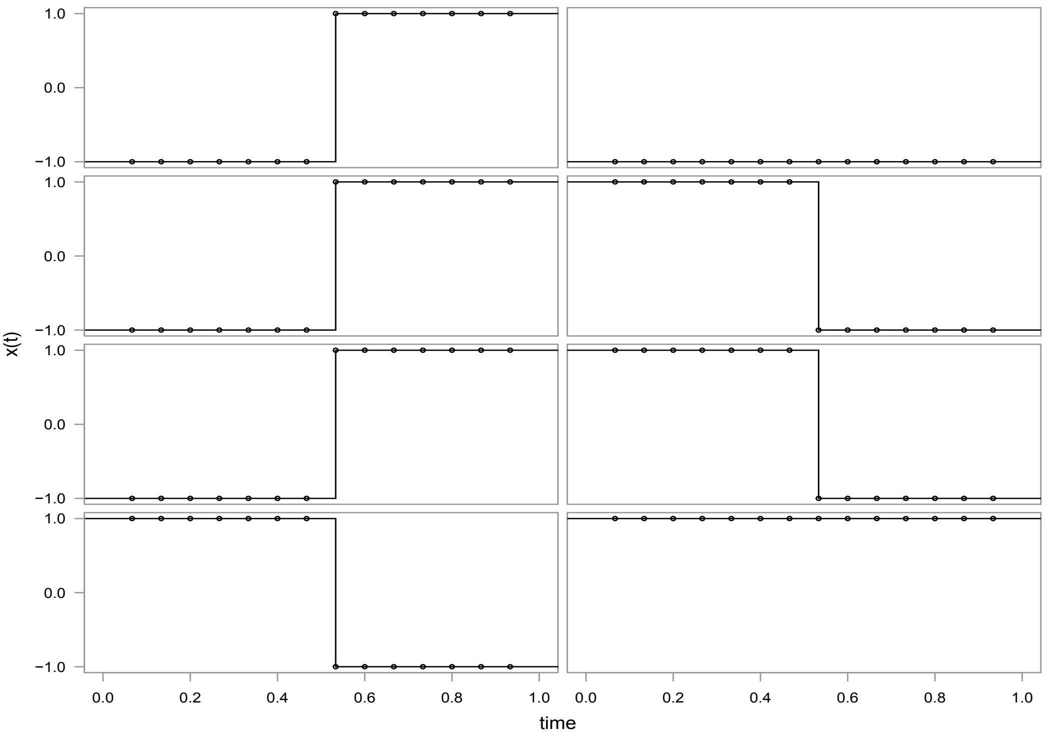


FIGURE 7.8: Eight run Bayesian A-optimal design for $n_{x,1} = 15, \Lambda = 10$ and quadratic basis for $\beta_2(t)$ for the functional logistic model.

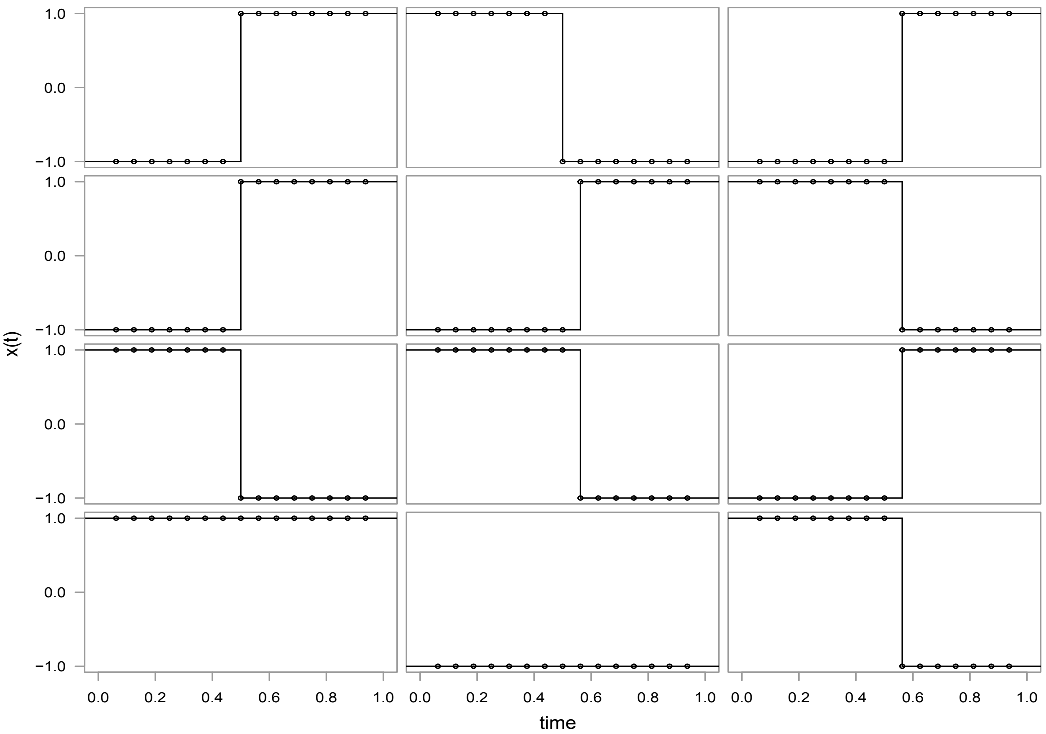


FIGURE 7.9: 12 run Bayesian A-optimal design for $n_{x,1} = 16, \Lambda = 10$ and quadratic basis for $\beta_2(t)$ for the functional logistic model.

7.10 Bayesian design criterion tailored for profile factors to a functional Poisson model with one profile factor

In this section the aim is to expand the investigation for the Bayesian design criterion tailored for profile factors, from Section 6.6, for a single profile factor Poisson model. The single profile factor choice is to avoid the excess computational expense involved in FGLMs, compared to the FLMs. The single profile factor Poisson model takes the form of the model in (7.33). The choice of the number of runs is $n = 12$.

It is assumed that control of the profile factor is represented by a BS of degree $d = 1$ with a single knot at $t = 0.5$. The choice for the basis of the functional parameters is linear, thus, the coefficients are identical to (7.26). The expectation of the objective function with respect to a prior distribution of the parameters is approximated using normal quadrature, mean zero variance 1, thus, the prior specification is identical to (7.28).

The matrix $\boldsymbol{C}(t)$ that includes the choice of functions and parameters of interest is a 3×2 matrix such that,

$$\boldsymbol{C}(t) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & t \end{pmatrix},$$

and subsequently the matrix $\boldsymbol{A}$ used in the design criteria for profile factors is a 3×3 matrix such that,

$$\boldsymbol{A} = \int_0^1 \boldsymbol{C}(t)\boldsymbol{C}(t)^T \, dt = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1/2 \\ 0 & 1/2 & 1/3 \end{pmatrix}.$$

The optimal designs found are evaluated under the alternative criterion; see Table 7.6, to provide a comparison amongst the designs. The efficiency values in brackets in Table 7.6, show that the Bayesian design tailored for profile factors is slighly more robust to the choice of design criterion. However, the efficiency values in the FLM in Section 6.6 are higher, compared to the functional Poisson model.

The functions of the profile factor have at most one change. For the pseudo A-optimality design criterion, out of the 12 runs there are four unique functions for the profile factor; see Figure 7.11, and the rest are repeated functions. For the Bayesian design criterion tailored for profile factors, out of the 12 runs there are six unique functions for the profile factor; see Figure 7.10, and the rest are repeated functions. Also, the Bayesian tailored to profile factors optimal design has two constant functions at the boundaries.

	Tailored criterion	Pseudo A-optimality
Tailored design	1.001 (1.000)	7.106 (0.872)
Pseudo A-optimal design	1.206 (0.830)	6.296 (1.000)

TABLE 7.6: Objective values of the pseudo A-optimality and Bayesian tailored to profile factors design criteria, with BS basis of degree $d = 1$ for $\boldsymbol{x}_{\cdot 1}(t)$ and single knot at $t = 0.5$, linear basis for $\beta_2(t)$ and $n = 12$, for the functional Poisson model. Efficiency values in brackets of evaluations of the optimal designs under the alternative criterion.

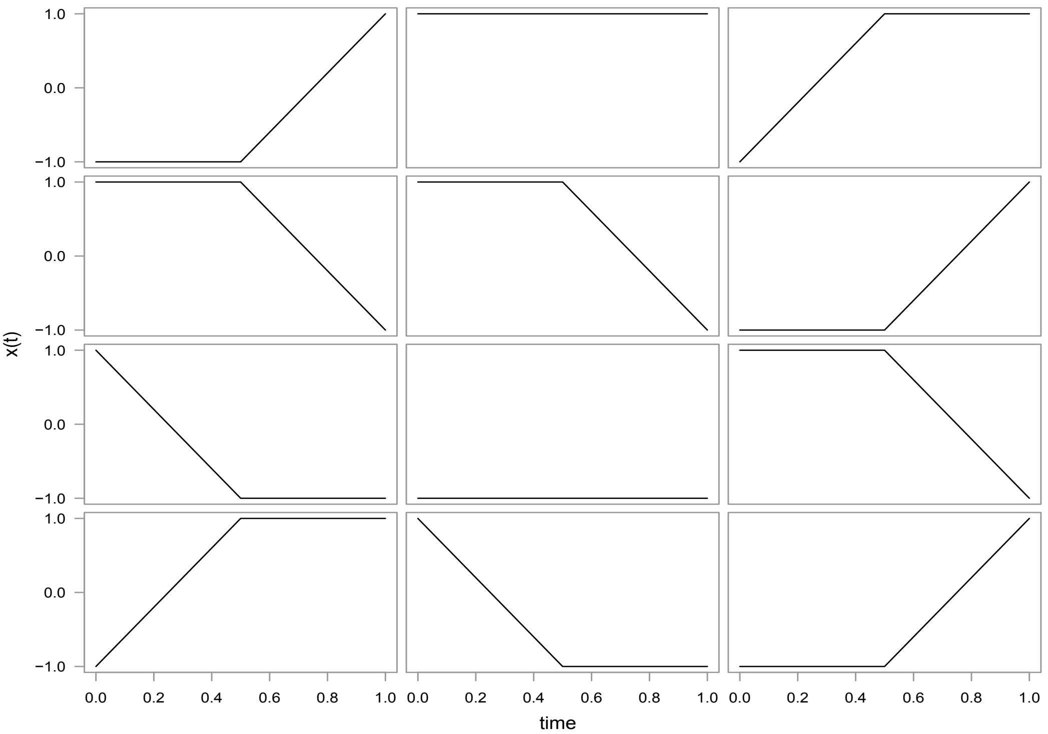


FIGURE 7.10: 12 run optimal design under the Bayesian design criterion tailored for profile factors, with BS basis of degree $d = 1$ for $x_{\cdot 1}(t)$, single knot at $t = 0.5$ and linear basis for $\beta_2(t)$ for the functional Poisson model.

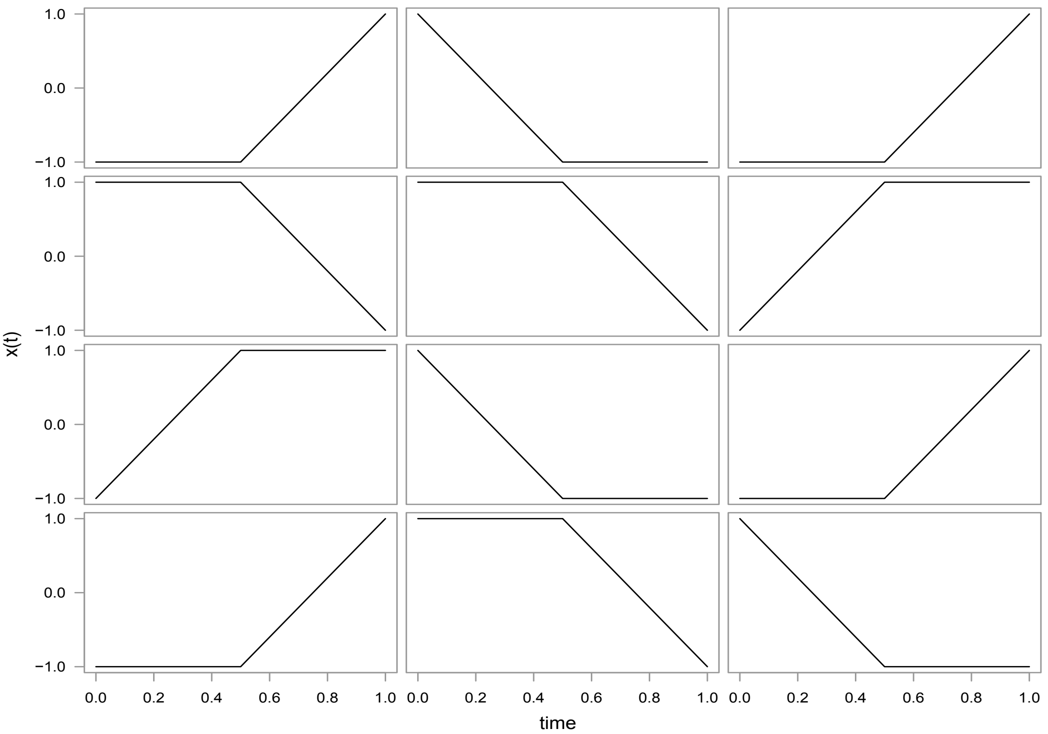


FIGURE 7.11: 12 run optimal design under the pseudo-Bayesian A-optimality criterion, with BS basis of degree $d = 1$ for $x_{\cdot 1}(t)$, single knot at $t = 0.5$ and linear basis for $\beta_2(t)$ for the functional Poisson model.

7.11 Conclusion

In this chapter, FGLMs have been tackled. It has been showed that such models, are an extension to the FLMs, as the linear predictor is identical to the right hand side of the FLMs. Thus, the methodology with the functions of the profile factors and the functional parameters represented by linear combinations of basis functions, still applies. The design problem has been described, and the objective functions for A- and D- optimality have been revised. Due to the dependence of the designs on the parameters, pseudo-Bayesian designs, that minimise the expectation of the objective functions with respect to a prior distribution of the parameters, have been discussed. This is to pass the available knowledge for the parameters. The expectation is usually the solution of an analytically intractable, and most likely, high-dimensional, integral. Approximation methods, including the Monte Carlo and quadrature methods, have been discussed, to numerically approximate the solution.

Functional logistic and functional Poisson models involving profile factors have been considered. Also, both the frequentist and the Bayesian approaches have been tackled. The investigation led to similar conclusions to the investigation of the FLM. For instance, it has been showed that increasing the number of basis functions does not improve the performance of the designs significantly, thus, a large number of basis functions is not needed. However, a larger number of basis functions is needed compared to FLM. Moreover, as the basis for the parameters is getting more complicated, the functions of the profile factors are getting more complicated as well.

An example with and without a quadratic effect of a profile factor has been considered. The addition of a higher order polynomial increases the computational expense, but the functions of the profile factors are less complicated, even not always at the boundaries. The design criterion tailored for profile factors has been used to find optimal designs for a functional Poisson model.

In general, the settings of the profile factors and the functional parameters have been less complicated compared to studies for FLMs. This is to avoid the excess computational burden. For models with high dimensionality, the Monte Carlo approximation has been preferred. The reason is that, in high dimensionality, many quadrature points are needed for an accurate approximation. This increases the computational expense significantly.

Chapter 8

R package fdesigns: optimal designs for functional models

8.1 Introduction to the fdesigns package

In this chapter, the aim is to describe the `R` package **fdesigns** that stands for functional designs, and demonstrate its use to find optimal experimental designs for functional models. The package includes two main functions, named `pflm()` and `pfglm()`. Both functions implement the methodology developed in this thesis, using basis functions to restrict the function space. In addition to the main functions, the package includes a support function, named `P()`. Furthermore, the package includes printing and plotting functions of the resulting objects.

The function `pflm()` stands for parallel functional linear models, and it is used to find optimal experimental designs for functional linear models using the coordinate exchange algorithm from Section 2.6. Similarly, the function `pfglm()` stands for parallel functional generalised linear models, and it is used to find optimal experimental designs for functional generalised linear models using the coordinate exchange algorithm. The use of the word parallel in the names of the functions suggests that the functions repeat the process for several starting designs. Further details of the two main functions of the package, including the implementation, the arguments and the output, are discussed in Sections 8.2 and 8.3.

Moreover, the support function `P()` stands for polynomials, and it is used to compute profile factor polynomials of a specific degree. The support function is further discussed in Section 8.4 to define its use and explain its difference to built-in functions in `R` that compute polynomial factors.

A big part of the **fdesigns** code is written in C++, through the packages **Rcpp** and **RcppArmadillo**; see Eddelbuettel et al. (2011) and Eddelbuettel and Sanderson (2014). **Rcpp** and **RcppArmadillo** offer seamless integration of R and C++. This increases the computational efficiency of the package and results in faster evaluation of the functions in **fdesigns**.

The use of the functions in the package are demonstrated through several examples of FLMs and FGLMs in Section 8.5. In every example, the set up of the model and the settings of the profile factors and the functional parameters are identical to examples from Chapter 6 and Chapter 7. In this way, the use of the package is exposed, and it enables interested users to repeat the examples and reproduce the optimal experimental designs.

The package is available on GitHub, and can be installed on your machine using the following R code,

```
devtools::install_github("damianosmichaelides/fdesigns")
```

or via the link: `https://github.com/damianosmichaelides/fdesigns`.

8.2 FLM implementation in fdesigns

Usage of `pflm()`

Functional linear models depending on profile factors are handled through the function `pflm()`. The function consists of several mandatory and optional arguments. Mainly, the mandatory arguments allow the user to specify the structure of the model and identify the settings of the experiment, the profile factors and the functional parameters. On the other hand, the optional arguments give the opportunity to the user to add a smoothing parameter or to change default bounds of the time and the profile factors, but mostly they remain unchanged in later examples. The usage of the function `pflm()` is such that:

```
pflm(formula, nsd = 1, mc.cores = 1, npf, tbounds, nruns, startd = NULL,
      dx, knotsx, pars, db, knotsb = NULL, lambda = 0,
      criterion = c("A", "D"), dlbound = -1, dubound = 1,
      tol = 0.0001, progress = FALSE)
```

with every argument explained in the next paragraphs and in the documentation of the function, available using the code,

```
?pflm
```

in the `R` console.

Arguments of `pflm()`

The FLM structure is specified through the `formula` argument. The `formula` argument is a built-in function in the **stats** package that is maintained by the `R` core team. The use of `formula` gives to the user complete flexibility to define the main effects, interactions and polynomials involved in the model. The number of starting designs is passed to the functions using the argument `nsd`, that defaults to one. The number of cores in parallel computing is set through the argument `mc.cores`, that also defaults to one. The time boundaries are defined as a vector of length two, i.e., the minimum and maximum time, in the argument `tbounds`. The total number of runs of the experiment is represented by the argument `nruns`. Furthermore, the choice of optimality criterion is made-through the argument `criterion`. Available optimality criteria include `criterion=`"A" and `criterion=`"D".

The number of profile factors in the model is specified through the argument `npf` and needs to match the number of factors specified in the `formula`. Scalar factors are treated as profile factors and expanded as a BS of degree zero and no interior knots. Thus, scalar factors are included in the argument `npf`, but their settings in the arguments that follow make them differ. The degree of the BS basis for each profile factor is indicated in the argument `dx`. The choice of `dx` must be a vector with its length identical to the value `npf` and the minimum entry value is zero. The interior knots for each profile factor basis is passed to the function through the argument `knotsx`. The argument `knotsx` must be a list with its length identical to the value `npf`. Each component of the list must be a vector, specifying the choice of interior knots of the profile factors. If a profile factor has no interior knots, for example a scalar factor, then the corresponding component in `knotsx` must be an empty vector.

The settings of the model parameters are controlled through the arguments `pars`, `db`, and `knotsb`. The argument `pars` is a character vector with accepted entries being "power" or "bspline", representing what basis system to use for basis expansion of each functional parameter. The argument `db` is a vector of the choice of degrees for the expansion of the parameters. If the basis choice is power series, then the degree one represents a linear basis, the degree two represents a quadratic basis, etc. The argument `knotsb` is a list, with each component being a vector of the interior knots of each parameter. For no interior knots, the component must be an empty vector. For instance, a scalar parameter is defined by the choice of "power", degree zero, and no interior knots. The default entry is NULL, meaning no interior knots for all parameters. The length of the vectors `pars` and `db`, and the list `knotsb` must be identical to the number of terms in the model, as specified by the `formula`. At last, penalisation of the complexity of the parameters is controlled through the argument `lambda`. The argument `lambda` is the smoothing parameter from Section 5.5. Thus, as the value of `lambda` increases, the penalisation on the wiggly functions increases as well.

The starting designs are passed to the functions through the argument `startd`. The argument `startd` must be a list of length equal to the number of different starting designs `nsd`. Each component of the list represents a single starting design, and it must be also a list of length equal to the number of profile factors `npf`. Every component of a single starting design must be a matrix named as per the terms in the formula argument. The number of rows of the matrix must be equal to `nruns` and the number of columns of the matrix must be equal to the number of basis functions of the corresponding profile factor. However, if the argument `startd` is not specified, random starting designs are generated. The lower and upper bounds of the design are specified by the arguments `dlbound` and `dubound`. Both arguments must be scalar values, representing common bounds for all profile factors. Due to properties of the BS basis, as discussed in Section 4.4, the design bounds are equivalent bounds to the profile factors. The tolerance value for the coordinate exchange algorithm is given by the argument `tol`, and the default value is 0.0001. Finally, the argument `progress` is a logical argument indicating whether the iteration progress of the algorithm must be printed. The default is FALSE.

A list of the arguments of the function `pflm()`, and a short description for each, is given in Table E.1 in Appendix E.1. The arguments with an asterisk are the mandatory arguments in the function. The function is demonstrated in Section 8.5 and the inputs are varied to assist the understanding of the arguments.

Output of `pflm()`

The function `pflm()` returns an object of class "flm". An object of class "flm" is a list of multiple components, but the most important are the final design (`design`) and the final objective value (`objval`). Additional components include: the number of iterations to identify the final design (`nits`), the starting design that led to the final design (`startd`), and several input arguments. Moreover, if the procedure was repeated for several starting designs, the output includes the value of the repetition that led to the final design (`bestrep`), and lists of all the starting designs (`allstartd`), the final designs (`alldesigns`), and the final objective values (`allobjvals`). A list of the output components of the function `pflm()`, and a short description for each, is given in Table E.2 in

Appendix E.2.

A print of the resulting "flm" object provides a summary of information. The summary of an "flm" object can be viewed using one of the following lines of code,

```
print()
summary()
```

in the `R` console. The summary takes the form:

```
The number of profile factors is: ()
The number of runs is: ()
The objective criterion is: ()
The objective value is: ()
The number of iterations is: ()
The computing elapsed time is: ()
```

Finally, a plot of the resulting "flm" object return the question "Which profile factor to plot?". The user needs to provide a scalar value between one and `npf`, indicating which profile factor functions to plot. After indicating the profile factor of interest, `nruns` optimal functions of the profile factor are plotted. The functions of the profile of an "flm" object are plotted using the code,

```
plot()
```

and then entering the profile factor of interest.

8.3 FGLM implementation in fdesigns

Usage of `pfglm()`

Functional generalised linear models depending on profile factors are handled through the function `pfglm()`. A large proportion of the arguments of the function `pfglm()` are identical to the arguments of the function `pflm()`, which is described in the previous section. The reason for this, is that the structure of the model and the settings of the experiment, the profile factors and functional parameters are identified identically. The additional arguments; mandatory and optional; are used to specify the family of the responses, the prior information, and the approximation method of the expectation of the objective function with respect to the prior distribution. The usage of the function `pfglm()` is such that:

```
pfglm(formula, nsd = 1, mc.cores = 1, npf, tbounds, nruns,
      startd = NULL, dx, knotsx, pars, db, knotsb = NULL,
      lambda = 0, criterion = c("A","D"), family,
      method = c("quadrature", "MC"), level = NULL, B = NULL,
      prior, dlbound = -1, dubound = 1, tol = 0.0001,
      progress = FALSE)
```

with every argument explained in the next paragraphs and in the documentation of the function, available using the code,

```
?pfglm
```

in the R console.

Arguments of pfglm()

Most of the arguments of the function pfglm() are identical, in name and usage, to the arguments of the function pflm(). Specifically, the identical arguments are: formula, nsd, mc.cores, npf, tbounds, nruns, startd, dx, knotsx, pars, db, knotsb, lambda, criterion, dlbound, dubound, tol and progress. They are described in detail in Section 8.2 and in Table E.1 in Appendix E.1. The additional arguments of the function pfglm() are: family, method, level, B, and prior.

The family argument specifies the distribution of the responses and the link function. It must specify the name of the family of interest. The available families are the Binomial and the Poisson, for the logit and log links, respectively. Specifically, they must be specified using the the family R function, or using a character string. Moreover, the method argument is used to specify the method of approximation of the expectation of the objective function with respect to a prior distribution of the parameters. The available approximation methods are method="quadrature", and method="MC", for deterministic quadrature approximation, and stochastic Monte Carlo approximation, respectively.

The arguments level and B are optional arguments. The level represents the accuracy level. As the value of level increases, the number of quadrature points in each dimension increases as well. If the method of approximation is method="quadrature", and level=NULL, then it defaults to 5. A high value of level may increase the computation time; especially for complicated models. If the model is complicated, i.e., several profile factors or interactions and polynomials, prefer to use method="MC". The argument B is a scalar value that represents the Monte Carlo sample size. If method = "MC", and `B`=NULL, it defaults to 10000.

Furthermore, the argument `prior` is used to specify the prior distribution. If `method` = "MC", the `prior` must be a function of two scalar arguments, `B` and `Q`. The value of `B` is the argument `B` from above, and `Q` is the total number of basis functions from the expansion of the functional parameters. The prior function must generate a `B` × `Q` matrix of random sample from the prior distribution.

On the other hand, if `method` = "quadrature", the argument `prior` must be a list. Currently, normal and uniform prior distributions are available. For normal distribution, the list must contain two components, named "mu" for the prior mean and "sigma2" for the prior variance-covariance matrix. For uniform distribution, the list must contain a single component named "unifbound", for the lower and upper bounds of the prior distribution. Further details on the possible ways to pass "mu", "sigma2", and "unifbound", to the `prior` argument, as well as further descriptions of all arguments, are given on Table E.3 in Appendix E.3, or in the documentation of the function `pfglm()` in `R`.

Output of `pfglm()`

The function `pfglm()` returns an object of class "fglm". The object of class "fglm" is a list of the components mentioned in the previous section for `pflm()`, but with additional information for the choice of family, the prior specification, and the method of approximation. A list of the additional output components of the function `pfglm()`, and a short description for each, is given in Table E.4 in Appendix E.4.

A print of the resulting "fglm" object provides a summary of information. The summary of an "fglm" object can be viewed using one of the following lines of code,

```r
print()
summary()
```

in the `R` console. The summary takes the form:

```
The number of profile factors is: ()
The number of runs is: ()
The objective criterion is: ()
The objective value is: ()
The number of iterations is: ()
The method of approximation is: ()
The family distribution and the link function are: () and ()
The computing elapsed time is: ()
```

The final functions of the profile factors, resulting from an object of class "fglm, are plotted exactly as described for objects of class "flm".

8.4 Profile polynomials in fdesigns

In the R software, polynomial effects of static factors are computed using the functions I() and poly(), from the R build-in packages **base** and **stats**, respectively. The function P() in the package **fdesigns** stands for polynomials, and it is used to compute profile factor polynomials. In other words, it is an extension to the base functions, to handle profile factors rather than static factors.

The reason of the extension to the build-in functions, is that profile factors are linear combinations of basis functions. Thus, the coefficients are a matrix; see Chapter 5, in contrast to a static factor that is a vector of scalar values. As a result, higher order polynomials of profile factors are calculated using the Hadamard product of the basis function combinations of the interaction factors.

The function P() is a support function in the package. Its main use is to specify the polynomial effects of interest in the formula argument of the main functions. The usage of the function P() is such that:

```
P(x, deg)
```

with the arguments explained in the next paragraph, as well as in the documentation of the function, available using the code,

```
?P
```

in the R console.

Both arguments of the function are mandatory. The argument x is a matrix, that corresponds to the coefficient matrix from the basis expansion of a profile factor. When the function is used in the formula in the main functions, the name of the profile factor have to match the name of the profile factor in startd in the main function pflm() or `pfglm()`. The argument `deg` is a scalar value that corresponds to the degree of the polynomial of the profile factor. A description of the two arguments is given on Table E.5 in Appendix E.5, or in the documentation of the function `P()` in `R`.

Finally, the function returns an attributes list. The list contains the polynomial coefficient matrix of the profile factor, the argument `x`, and the argument `deg`.

8.5 Examples using the functions in fdesigns

In this section, the **fdesigns** package is demonstrated through several examples. Attention is given on the main functions of the package, meaning the functions `pflm()` and `pfglm()`. The support function `P()` for computing profile factor polynomials, is used in the `formula` argument of the main functions to specify polynomials in the functional models. The examples cover various scenarios of functional linear and functional generalised linear models, including models with main effects, interactions, polynomials, and smoothing to penalise complex functions.

Even though the functions `pflm()` and `pfglm()` are able to find efficient designs for models depending on multiple profile factors, as the models become more complicated, the computational expense increases, and significantly more computational resources are required. For that reason, most of the examples tackled in this section are kept simple on purpose. A few examples, mainly for generalised models, are slightly more expensive to run, but the reader is informed. All examples are referenced to source chapters, in order to demonstrate to interested users how to reproduce the optimal designs.

8.5.1 FLM with one profile factor

In this example, the FLM with a single profile factor from Section 6.1 is considered. It is assumed that control of the profile factor is represented by a BS basis of degree zero and for this example, only the choice of three equally spaced interior knots is considered, with time boundaries being 0 and 1. Thus, there are four basis functions for the profile factor; see (4.3). Moreover, the choice of the number of runs is four.

```
tbounds <- c(0, 1)
nruns <- 4
npf <- 1
dx <- c(0)
knotsx <- list(c(0.25, 0.50, 0.75))
```

```
nx <- rep(0, npf)
for (j in 1:npf) {
  nx[j] <- dx[j] + length(knotsx[[j]]) + 1
}
```

One thousand starting designs are considered, which are generated and passed to the function `pflm()` manually. Thus, the starting design must to be a list of 1000 components, and each component must be also a list with one matrix component for the single profile factor names as in the `formula`. The bounds of the profile factor are set to the defaults. Alternatively, one could leave the `startd` argument to the default entry which is NULL, and the function would randomly generate the starting designs. In this case, a set seed for reproducibility has to be set.

```
indd <- list()
startd <- list()
dlbound <- -1
dubound <- 1
nsd <- 1000
for (c in 1:nsd) {
  set.seed(c)
  for (i in 1:npf) {
    indd[[i]] <- matrix(runif(nruns * nx[i], dlbound, dubound),
                  nrow = nruns, ncol = nx[i])
    names(indd)[i] <- paste0("x", i, sep="")
  }
  startd[[c]] <- indd
}
```

The functional parameter is assumed to be a linear power series and the objective criterion is A-optimality. All other arguments are kept to their default values.

```
example1a <- pflm(formula = ~ x1, nsd = nsd, mc.cores = 1, npf = npf,
                  tbounds = tbounds, nruns = nruns, startd = startd,
                  dx = dx, knotsx = knotsx, pars = c("power"), db = c(1),
                  knotsb = list(c()), criterion = "A", lambda = 0,
                  dlbound = dlbound, dubound = dubound, tol = 0.0001,
                  progress = TRUE)
```

Printing the resulting "flm" object using the code,

```
print(example1a)
```

provides the summary of the outcome,

```
The number of profile factors is: 1
The number of runs is: 4
The objective criterion is: A-optimality
The objective value is: 8.75
The number of iterations is: 5
The computing elapsed time is: 00:00:00
```

with the objective value being an exact match to the corresponding value in Table 6.1. Additionally, the final design is extracted using the code,

```
example1a$design
```

and the outcome is a 4×4 design matrix.

```
$x1
     [,1] [,2] [,3] [,4]
[1,]    1    1    1    1
[2,]    1    1   -1   -1
[3,]   -1   -1    1    1
[4,]   -1   -1    1    1
```

The optimal functions of the profile factor are plotted using the code,

```
plot(example1a)
```

The function returns the question,

```
Which profile factor to plot?
```

in which the user needs to enter the scalar value of the profile factor of interest; in this case, 1. After that, the functions of the profile factor are plotted; see Figure 8.1, and match the functions from Figure 6.2. To add a point, out of the `nsd` starts, multiple final designs may attained the same final objective value, especially in simple models. In such case, the final design and so the functions of the profile factor may be slightly different or in different order of runs.

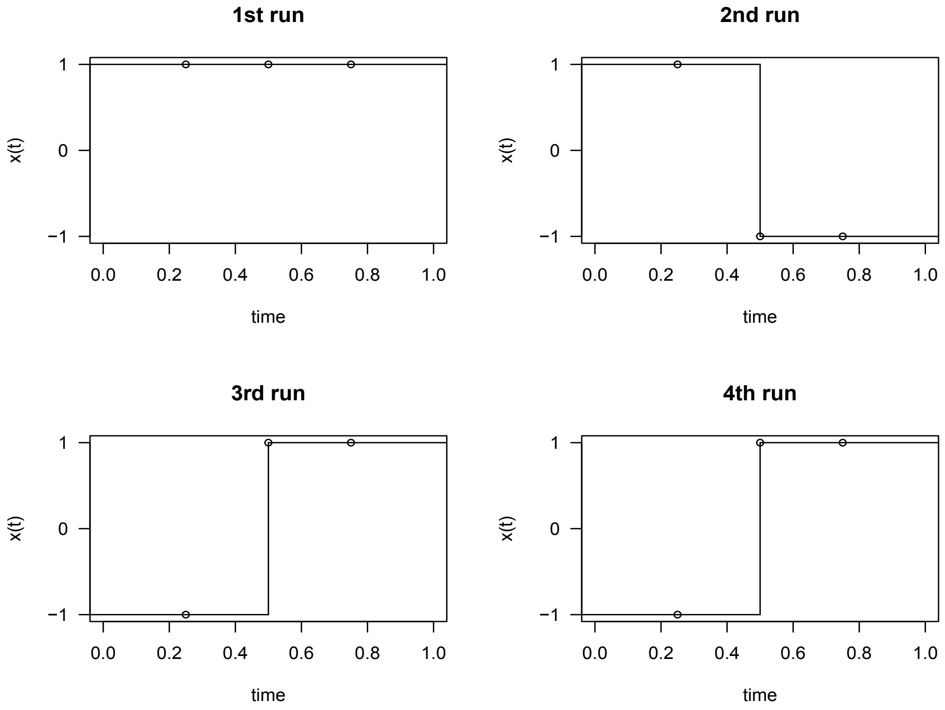


FIGURE 8.1: Four run A-optimal design for $n_{x,1} = 4$, linear basis for $\beta_2(t)$ and step function basis for $\boldsymbol{x}_{.1}(t)$ for the FLM, plotted using the "flm" object in package **fdesigns**.

Alternatively, suppose that the objective criterion becomes D-optimality and the functional parameter is represented by a quadratic power basis, with everything else remain unchanged, then the code is,

```
example1b <- pflm(formula = ~ x1, nsd = nsd, mc.cores = 1, npf = npf,
                  tbounds = tbounds, nruns = nruns, startd = startd,
                  dx = dx, knotsx = knotsx, pars = c("power"), db = c(2),
                  knotsb = list(c()), criterion = "D", lambda = 0,
                  dlbound = dlbound, dubound = dubound, tol = 0.0001,
                  progress = TRUE)

print(example1b)
```

and the summary of the outcome is,

```
The number of profile factors is: 1
The number of runs is: 4
The objective criterion is: D-optimality
The objective value is: 4.583135
The number of iterations is: 6
The computing elapsed time is: 00:00:00
```

with the objective value being an exact match to the corresponding value in Table 6.4.

8.5.2 FLM with one profile factor and roughness penalty

In this example, the model with one profile factor as in the example from Section 8.5.1 is considered. The addition to the previous example is that the complexity of the parameter is penalised through a smoothing parameter $\Lambda = 10$. This scenario has been investigated before, in the example from Section 6.5 to find Bayesian optimal designs. The penalty is passed to the function through the argument `lambda=10`. The objective function is A-optimality and the parameter basis is a quadratic power basis. The Bayesian A-optimal design can be found using the code,

```
example2 <- pflm(formula = ~ x1, nsd = nsd, mc.cores = 1, npf = npf,
                 tbounds = tbounds, nruns = nruns, startd = startd,
                 dx = dx, knotsx = knotsx, pars = c("power"), db = c(2),
                 knotsb = list(c()), criterion = "A", lambda = 10,
                 dlbound = dlbound, dubound = dubound, tol = 0.0001,
                 progress = TRUE)

print(example2)
```

and the summary of the outcome is,

```
The number of profile factors is: 1
The number of runs is: 4
The objective criterion is: A-optimality
The objective value is: 8.800694
The number of iterations is: 5
The computing elapsed time is: 00:00:00
```

with the objective value being an exact match to the corresponding value in Table 6.7. The final designs and the optimal functions of the profile factor are extracted in the same way as in the example in the previous section.

8.5.3 FLM with one profile factor and three scalar factors

In this example, the FLM with a single profile factor and three scalar factors from Section 6.2 is considered. It is assumed that control of the profile factor is represented by a BS basis of degree zero with three equally spaced interior knots and time boundaries again being 0 and 1. As before, there are four basis functions for the profile factor.

Firstly, the case 1 model with main effects of the scalar factors from (7.25) is considered. Moreover, the choice of the number of runs is 12. The scalar factors are passed as profile factors to the function `pflm()`, with degree zero and no interior knots to specify that they are scalar factors. Thus, the argument `npf` is equal to 4.

```
tbounds <- c(0, 1)
nruns <- 12
npf <- 4
dx <- c(0, 0, 0, 0)
knotsx <- list(c(0.25, 0.50, 0.75), c(), c(), c())
nx <- rep(0, npf)
for (j in 1:npf) {
  nx[j] <- dx[j] + length(knotsx[[j]]) + 1
}
```

The starting designs are generated and passed to the function as in the previous example, with the bounds of the factors set to the defaults. The factors are named $x1$, $x2$, $x3$, $x4$ and must match the factors in the `formula` argument.

```
indd <- list()
startd <- list()
dlbound <- -1
dubound <- 1
nsd <- 1000
for (c in 1:nsd) {
  set.seed(c)
  for (i in 1:npf) {
    indd[[i]] <- matrix(runif(nruns * nx[i], dlbound, dubound),
                        nrow = nruns, ncol = nx[i])
    names(indd)[i] <- paste0("x", i, sep="")
  }
  startd[[c]] <- indd
}
```

The functional parameter is assumed to be represented by a linear power series and the scalar parameters are represented through a power series of degree zero, as discussed in the description of the function `pflm()`. Moreover, the objective criterion is A-optimality. All other arguments are kept to their default values.

```
example3a <- pflm(formula = ~ x1 + x2 + x3 + x4, nsd = nsd, mc.cores = 1,
                  npf = npf, tbounds = tbounds, nruns = nruns,
                  startd = startd, dx = dx, knotsx = knotsx,
                  pars = c("power", "power", "power", "power"),
                  db = c(1, 0, 0, 0), knotsb = list(c(), c(), c(), c()),
                  criterion = "A", lambda = 0, dlbound = dlbound,
                  dubound = dubound, tol = 0.0001, progress = FALSE)
```

Printing the resulting "flm" object using the code,

```
print(example3a)
```

provides the summary of the outcome,

```
The number of profile factors is: 4
The number of runs is: 12
The objective criterion is: A-optimality
The objective value is: 2.833333
The number of iterations is: 10
The computing elapsed time is: 00:00:04
```

with the objective value being an exact match to the corresponding value found in Section 6.2. The A-optimal design for the scalar factors is extracted using the code,

```
final_design_a <- example3a$design
scalar_design_a <- matrix(0, nrow = nruns, ncol = (npf - 1))
for (k in 2:(npf)) {
  scalar_design_a[,k-1] <- final_design_a[[k]]
}
scalar_design_a
```

which gives a design at the boundaries that is identical to the case 1 design from Table 6.5.

```
      [,1] [,2] [,3]
 [1,]   -1   -1   -1
 [2,]    1    1    1
 [3,]   -1   -1   -1
 [4,]    1   -1   -1
 [5,]   -1    1    1
 [6,]   -1    1   -1
 [7,]    1    1   -1
 [8,]   -1   -1    1
 [9,]    1    1   -1
[10,]    1   -1    1
[11,]   -1    1    1
[12,]    1   -1    1
```

The order of the design runs may differ due to several designs attaining the same final objective value. For instance, `R` finds that the best repetition is repetition 36, using the component `bestrep` of the "flm" object.

```
example3a$bestrep
```

A print of the first 65 objective values, shows that the evaluation of several designs gives the objective value which is 2.833333.

```
[1] 2.968759 2.964321 2.854940 2.949824 2.970566 2.953171 2.852684
[8] 2.853202 2.961305 2.885000 2.961604 2.900983 2.985471 2.852041
[15] 2.965235 2.834491 2.917935 2.941454 2.833333 2.834491 2.852251
[22] 2.917314 2.910675 2.870574 2.927504 2.894165 2.946153 2.968504
[29] 2.917842 2.916820 2.965131 2.966247 2.852476 2.852476 2.893333
[36] 2.833333 3.111223 2.853209 2.973834 2.833333 2.904321 2.893324
[43] 2.893324 3.012996 2.852341 2.876926 2.893324 2.928718 2.983806
[50] 2.918199 2.940833 2.935810 2.894522 2.961303 2.916820 2.907548
[57] 2.974558 2.919542 2.852041 2.833333 3.177417 2.833333 2.969806
[64] 2.893325 2.925782
```

For this reason, the designs may be slightly different or the runs may be in different order.

In addition to the main effects of the scalar factors, the case 2 model in (7.25) contains the quadratic effects of the scalar factors. In this case, the quadratic effects are passed using the function `P()` that computes the polynomials. However, the FLM becomes much more complicated, and the computational expense is significantly higher with each repetition taking more than 30 minutes. For this reason, the code provided below is for one starting design (computation time is expected to be about 32 minutes). The starting design is for the repetition 130, which is the repetition that led to the final design. The use of set seed is recommended, to achieve reproducibility.

```
indd <- list()
startd <- list()
dlbound <- -1
dubound <- 1
nsd <- 1
set.seed(130)
for (c in 1:nsd) {
  for (i in 1:npf) {
    indd[[i]] <- matrix(runif(nruns * nx[i], dlbound, dubound),
                        nrow = nruns, ncol = nx[i])
    names(indd)[i] <- paste0("x", i, sep="")
  }
  startd[[c]] <- indd
}
```

```
example3b <- pflm(formula = ~ x1 + x2 + x3 + x4 + P(x2, 2) + P(x3, 2) +
                P(x4, 2), nsd = nsd, mc.cores = 1, npf = npf,
                tbounds = tbounds, nruns = nruns, startd = startd,
                dx = dx, knotsx = knotsx, pars = c("power", "power",
                "power", "power", "power", "power", "power"),
                db = c(1, 0, 0, 0, 0, 0, 0),
                knotsb = list(c(), c(), c(), c(), c(), c(), c()),
                criterion = "A", lambda = 0, dlbound = dlbound,
                dubound = dubound, tol = 0.0001, progress = FALSE)
```

and printing the resulting "flm" object using the code,

```
print(example3b)
```

provides the summary of the outcome,

```
The number of profile factors is: 4
The number of runs is: 12
The objective criterion is: A-optimality
The objective value is: 4.50883
The number of iterations is: 18
The computing elapsed time is: 00:32:30
```

with the objective value being an exact match to the corresponding value found in Section 6.2.

8.5.4 FLM with two profile factors and their interaction

In this example, the FLM to consider depends on two profile factors and their interaction. This model has been tackled in Section 6.4 to compare the final designs of the FLM with and without the interaction of the profile factors. The profile factors are represented by cubic BS basis and four equally spaced interior knots, with time boundaries at 0 and 1. The addition of the interaction term and the use of cubic splines, increases the computational expense, but not as much as the example with the quadratic effects of the scalar factors. On average, starting designs iterate to their final designs in 10 seconds.

```
tbounds <- c(0, 1)
nruns <- 12
npf <- 2
dx <- c(3, 3)
knotsx <- list(c(0.20, 0.40, 0.60, 0.80), c(0.20, 0.40, 0.60, 0.80))
nx <- rep(0, npf)
for (j in 1:npf) {
  nx[j] <- dx[j] + length(knotsx[[j]]) + 1
}

indd <- list()
startd <- list()
dlbound <- -1
dubound <- 1
nsd <- 1000
for (c in 1:nsd) {
  set.seed(c)
  for (i in 1:npf) {
    indd[[i]] <- matrix(runif(nruns * nx[i], dlbound, dubound),
                        nrow = nruns, ncol = nx[i])
    names(indd)[i] <- paste0("x", i, sep="")
  }
  startd[[c]] <- indd
}
```

The model includes three functional parameters, two for the main effects and one for the interaction, which are represented by BS basis of degree zero and single knots at $t = 0.5$. Moreover, the objective criterion is D-optimality. All other arguments are kept to their default values.

```
example4 <- pflm(formula = ~ x1 + x2 + x1:x2, nsd = nsd, mc.cores = 1,
                npf = npf, tbounds = tbounds, nruns = nruns,
                startd = startd,  dx = dx, knotsx = knotsx,
                pars = c("bspline", "bspline", "bspline"),
                db = c(0, 0, 0), knotsb = list(c(0.5), c(0.5), c(0.5)),
                criterion = "D", lambda = 0, dlbound = dlbound,
                dubound = dubound, tol = 0.0001, progress = FALSE)
```

The summary of the "flm" object resulting from,

```
print(example4)
```

shows that the objective value of the D-optimal design is 0.335, which is an exact match to the objective value found in the identical example developed in Section 6.4.

```
The number of profile factors is: 2
The number of runs is: 12
The objective criterion is: D-optimality
The objective value is: 0.3348156
The number of iterations is: 4
The computing elapsed time is: 00:00:06
```

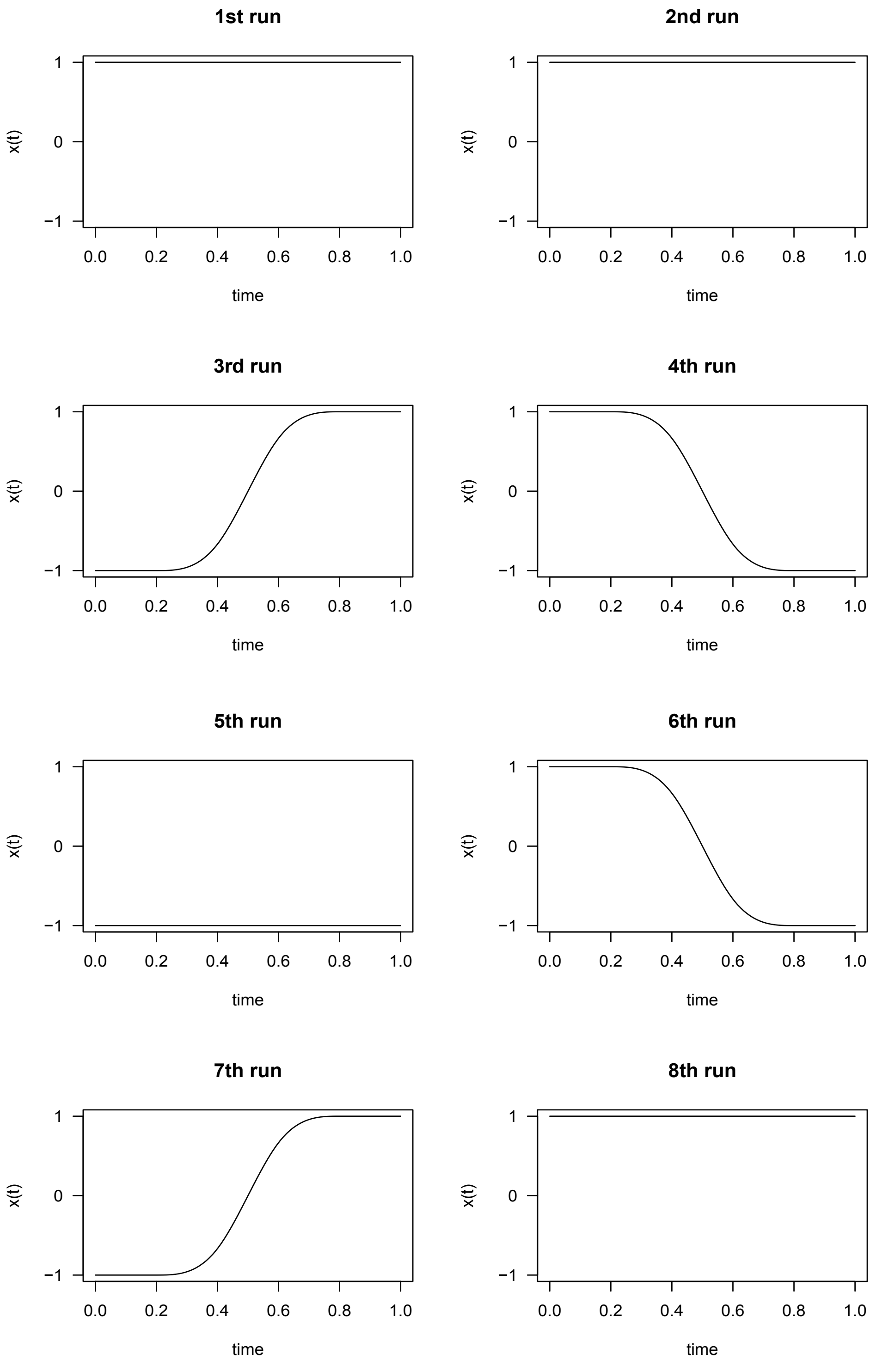
Plots of the functions of a profile factor can be generated using the code,

```
plot(example4)
```

that returns the question

```
Which profile factor to plot?
```

and the user has to enter the scalar value of the profile factor of interest. In this example, there are two profile factors, thus, either enter 1 or 2. For this example, the functions of the first profile factor are plotted; see Figure 8.2, and match the functions from Figure 6.15.


1st run
2nd run
3rd run
4th run
5th run
6th run
7th run
8th run
x(t)
1
0
−1
0.0
0.2
0.4
0.6
0.8
1.0
time

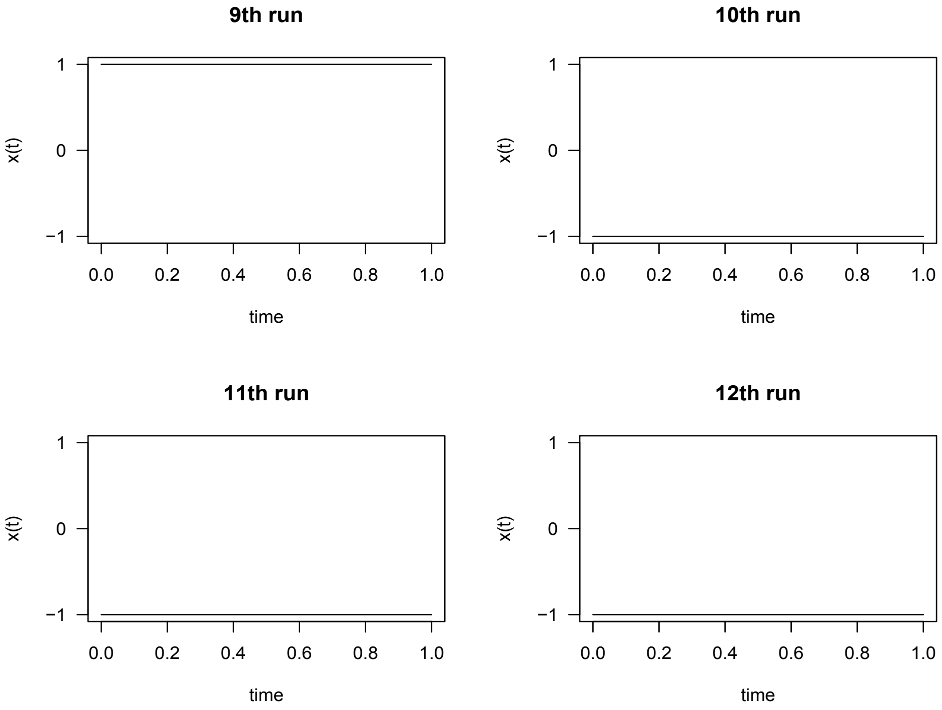


FIGURE 8.2: 12 run D-optimal design for the first profile factor of a FLM with two profile factors of cubic BS basis and four equally spaced knots, degree $d = 0$ BS basis for the functional parameters and interaction effect considered. The functions are plotted using the "flm" object in package **fdesigns**.

8.5.5 FGLM with one profile factor and quadrature approximation

In this example, the functional logistic model depending on a single profile factor, from Section 7.6 is considered. Thus, the `family` choice in `R` is "binomial". It is assumed that control of the profile factor is represented by a BS basis of degree zero and for this example, only the choice of seven equally spaced interior knots is considered, with time boundaries being 0 and 1. Thus, there are eight basis functions for the profile factor; see (4.3). The number of runs is eight.

```
tbounds <- c(0, 1)
nruns <- 8
npf <- 1
dx <- c(0)
knotsx <- list(c(0.125, 0.250, 0.375, 0.500, 0.625, 0.750, 0.875))
nx <- rep(0, npf)
for (j in 1:npf) {
  nx[j] <- dx[j] + length(knotsx[[j]]) + 1
}
```

One thousand starting designs are considered, which are generated and passed to the function `pfglm()` manually, as in previous examples. Also, the bounds of the profile factor are set to the defaults.

```
indd <- list()
startd <- list()
dlbound <- -1
dubound <- 1
nsd <- 1000
for (c in 1:nsd) {
  set.seed(c)
  for (i in 1:npf) {
    indd[[i]] <- matrix(runif(nruns * nx[i], dlbound, dubound),
                        nrow = nruns, ncol = nx[i])
    names(indd)[i] <- paste0("x", i, sep="")
  }
  startd[[c]] <- indd
}
```

The functional parameter is assumed to be a linear power series. The objective is to identify pseudo-Bayesian A-optimal designs. The prior of the parameters is assumed to be normal, with mean zero and variance one. To approximate the expectation with respect to the prior, a normal quadrature scheme with abscissas and weights is applied. All other arguments are kept to their default values.

```
example5 <- pfglm(formula = ~ x1, nsd = nsd, mc.cores = 1, npf = npf,
                  tbounds = tbounds, nruns = nruns, startd = startd,
                  dx = dx, knotsx = knotsx, pars = c("power"), db = c(1),
                  knotsb = list(c()), lambda = 0, criterion = "A",
                  family = binomial, method = c("quadrature"),
                  level = NULL, B = NULL, prior = list(mu = c(0),
                  sigma2 = c(1)), dlbound = -1, dubound = 1, tol = 0.0001,
                  progress = TRUE)
```

Printing the resulting "fglm" object using the code,

```
print(example5)
```

provides the summary of the outcome,

```
The number of profile factors is: 1
The number of runs is: 8
The objective criterion is: A-optimality
The objective value is: 21.64537
The number of iterations is: 6
The method of approximation is: quadrature
The family distribution and the link function are: binomial and logit
The computing elapsed time is: 00:00:03
```

with the objective value being an exact match to the corresponding value in Table 7.1. As for the FLM examples, the final design is extracted using the code,

```
example5$design
```

and the outcome is a 8 × 8 design matrix, always at the boundaries.

```
$x1
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]   -1   -1   -1   -1    1    1    1    1
[2,]   -1   -1   -1   -1    1    1    1    1
[3,]    1    1    1    1   -1   -1   -1   -1
[4,]    1    1    1    1    1    1    1   -1
[5,]   -1   -1   -1   -1   -1   -1   -1   -1
[6,]    1    1    1    1   -1   -1   -1   -1
[7,]    1    1    1    1   -1   -1   -1   -1
[8,]   -1   -1   -1   -1    1    1    1    1
```

The optimal functions of the profile factor are plotted using the code,

```
plot(example5)
```

and the value 1 is required, to plot the functions of profile factor 1. If more profile factors are involved in the model, the input should be the profile factor of interest. The final functions of the profile factor are available in Figure 8.3, and match the functions from Figure 7.1.

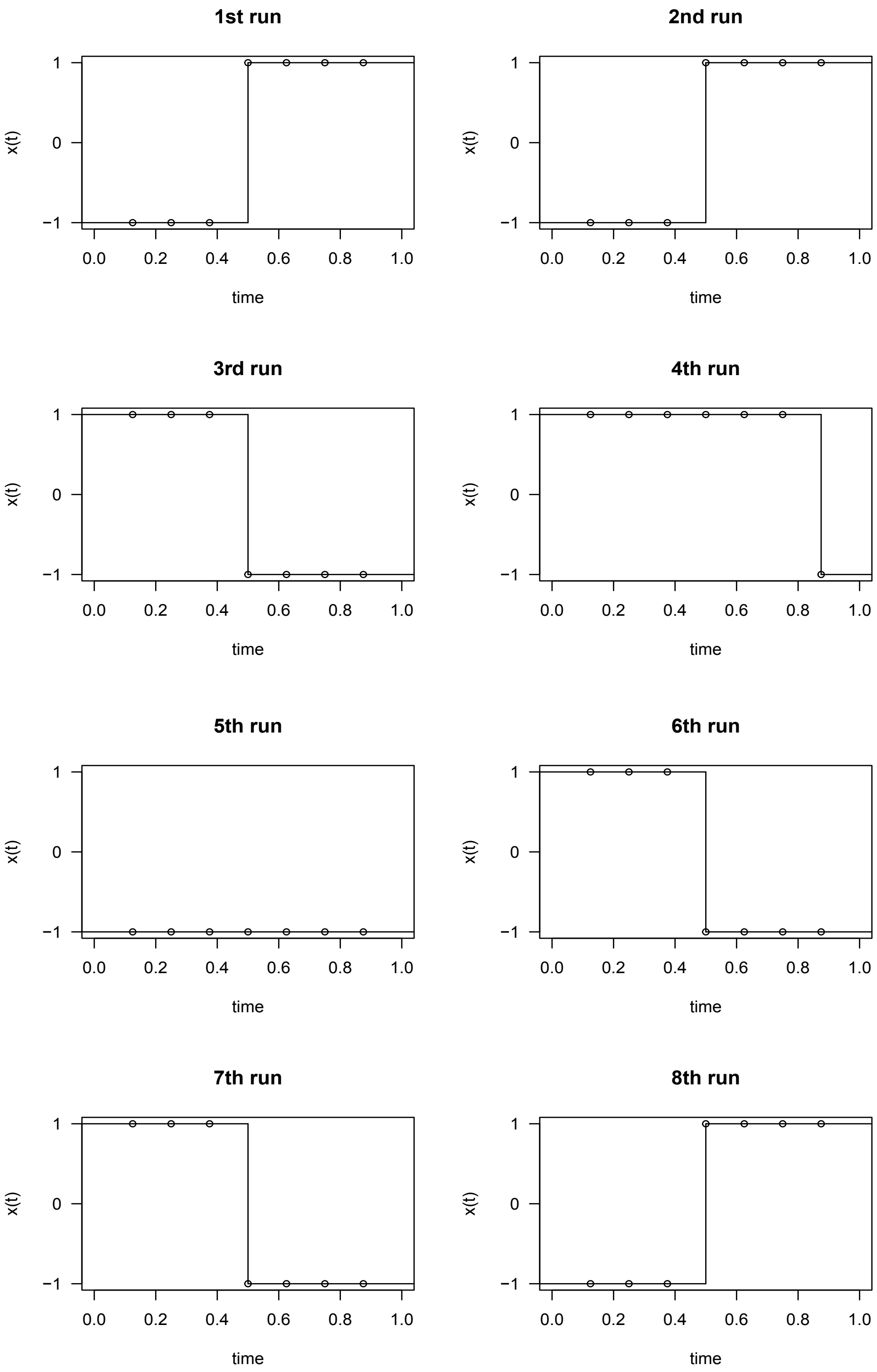


FIGURE 8.3: Eight run pseudo-Bayesian A-optimal design for $n_{x,1} = 4$, linear basis for $\beta_2(t)$, and step function basis for $x_{\cdot 1}(t)$ for the functional logistic model, plotted using the "fglm" object in package **fdesigns**.

8.5.6 FGLM with one profile factor depending on main or main and quadratic effect and MC approximation

In this example, the functional Poisson model depending on a single profile factor, from Section 7.8 is considered. Thus, the `family` choice in `R` is "poisson". It is assumed that control of the profile factor is represented by a BS basis of degree one, with the choice of four equally spaced interior knots. Thus, there are six basis functions for the profile factor. The time boundaries are 0 and 1, and the number of runs is 12. As in Section 7.8, the model with main effect and then, main and quadratic effect of the profile factor is tackled. The parameters are assumed to be represented by a BS basis of degree one and single knot at $t = 0.5$.

```
tbounds <- c(0, 1)
nruns <- 12
npf <- 1
dx <- c(1)
knotsx <- list(c(0.20, 0.40, 0.60, 0.80))
nx <- rep(0, npf)
for (j in 1:npf) {
  nx[j] <- dx[j] + length(knotsx[[j]]) + 1
}
```

The use of 12 runs and BS basis for the parameters, increases the complexity, and hence, the computational expense. For this reason, the Monte Carlo approximation, with a normal prior, mean zero and variance two, is preferred. The `prior` argument, when `method` is "MC", must be a function. The function used is,

```
set.seed(100)
prmc <- function(B, Q){
  matrix(rnorm(B * Q, mean = 0, sd = sqrt(2)), nrow = B, ncol = Q)
}
```

with set seed used for reproducibility of the results.

For the main effect case, the computing elapsed time for identifying one final design is around 30 minutes, and the final design was identified from the 151th repetition. Thus, the starting design is,

```
indd <- list()
startd <- list()
dlbound <- -1
```

```
dubound <- 1
nsd <- 1
set.seed(151)
for (c in 1:nsd) {
  for (i in 1:npf) {
    indd[[i]] <- matrix(runif(nruns * nx[i], dlbound, dubound),
                        nrow = nruns, ncol = nx[i])
    names(indd)[i] <- paste0("x", i, sep="")
  }
  startd[[c]] <- indd
}
```

The objective is to identify the pseudo-Bayesian D-optimal design for the 151th repetition. All other arguments are kept to their default values.

```
example6a <- pfglm(formula = ~ 1 + x1, nsd = nsd, mc.cores = 1, npf = 1,
                   tbounds = tbounds, nruns = nruns, startd = startd,
                   dx = dx, knotsx = knotsx, pars = c("bspline"),
                   db = c(1), knotsb = list(c(0.5)), lambda = 0,
                   criterion = "D", family = poisson, method = c("MC"),
                   level = NULL, B = 10000, prior = prmc,
                   dlbound = dlbound, dubound = dubound, tol = 0.0001,
                   progress = TRUE)
```

Printing the resulting "fglm" object using the code,

```
print(example6a)
```

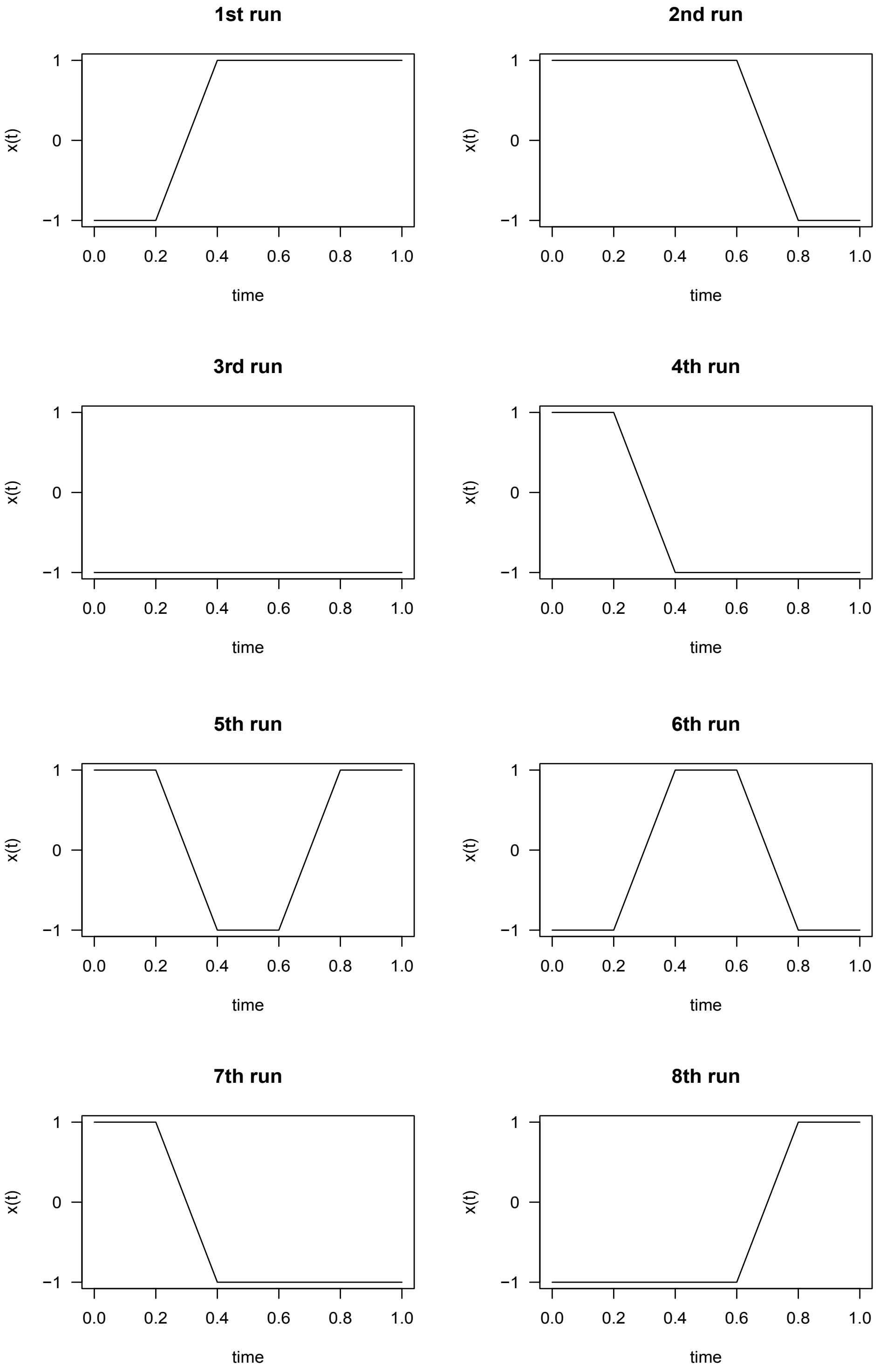
provides the summary of the outcome,

```
The number of profile factors is: 1
The number of runs is: 12
The objective criterion is: D-optimality
The objective value is: 1.92431
The number of iterations is: 4
The method of approximation is: MC
The family distribution and the link function are: poisson and log
The computing elapsed time is: 00:37:20
```

with the objective value being an exact match to the corresponding value in Section 7.8. The optimal functions of the profile factor are plotted using the code,

```
plot(example6a)
```

and the plots in Figure 8.4 are the exact match to the functions in Figure 7.5.



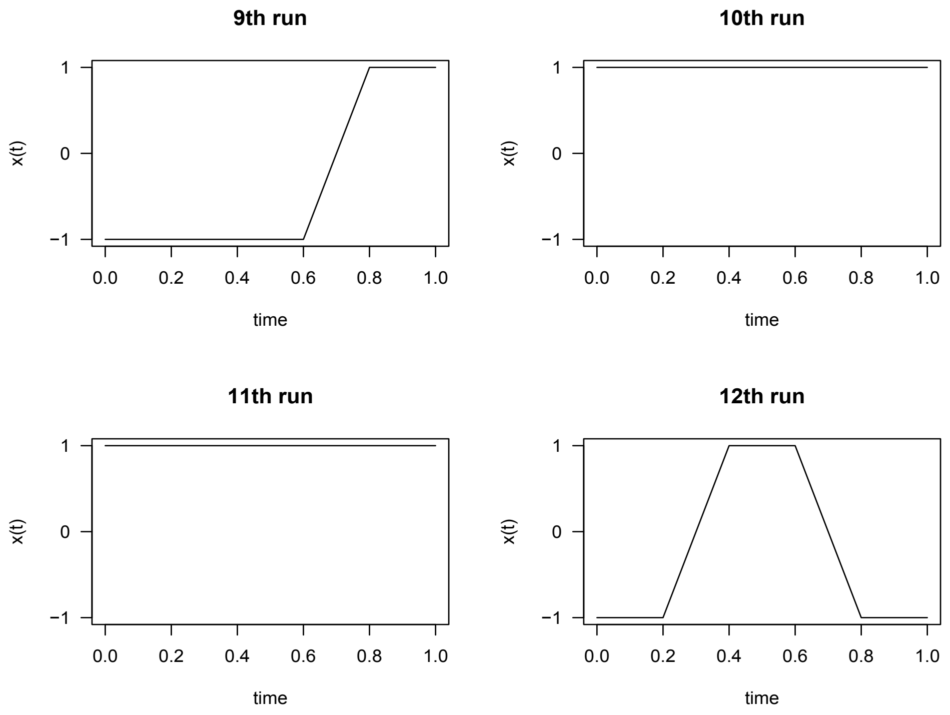


FIGURE 8.4: 12 run pseudo-Bayesian D-optimal design for one profile factor with BS degree $d = 1$ basis and $n_{x,1} = 6$ basis functions with the main effect considered, and BS basis of degree $d = 1$ for $\beta_2(t)$ for the functional Poisson model. The functions are plotted using the "fglm" object in package **fdesigns**.

For identifying the optimal design for the Poisson model with main and quadratic effects, the code below can be used. However, the computational expense is massive. For this reason, avoid running the lines below, unless using a supercomputer.

```
example6b <- pfglm(formula = ~ 1 + x1 + P(x1, 2), nsd = nsd, mc.cores = 1,
                  npf = 1, tbounds = tbounds, nruns = nruns,
                  startd = startd, dx = dx, knotsx = knotsx,
                  pars = c("bspline", "bspline"), db = c(1, 1),
                  knotsb = list(c(0.5), c(0.5)), lambda = 0,
                  criterion = "D", family = poisson, method = c("MC"),
                  level = NULL, B = 10000, prior = prmc,
                  dlbound = dlbound, dubound = dubound, tol = 0.0001,
                  progress = TRUE)
```

with printed output,

```
The number of profile factors is: 1
The number of runs is: 12
The objective criterion is: D-optimality
The objective value is: 7.28274
The number of iterations is: 7
The method of approximation is: MC
The family distribution and the link function are: poisson and log
The computing elapsed time is: 07:42:29
```

for which the objective value is also a match to its corresponding value in Section 7.8.

Chapter 9

Future work

9.1 Full higher order functional linear model

In the future, an aim is to extend the methodology developed in the thesis, to full higher order FLMs. In the previous chapters, polynomials and interactions are incorporated into the model through a functional of the profile factors, with a univariate functional parameter, integrated on single index. For the single index integration and the use of B-spline basis functions, closed form expressions of the integrals were derived. According to Morris (2015), the generalisation to the full model is more flexible. This is because it does not omit some interaction terms, thus, it gives more information of the interaction. The full second, and k^{th} order FLMs, with multivariate parameter functions and multiple integrals are discussed in Chapter 3, with reference to Yao and Müller (2010) and Usset et al. (2016). The functional quadratic model, and the k^{th} order FLMs polynomials and interactions are given in (3.3), (3.4), and (3.5), respectively.

The form of the full model to consider is a combination of the models in (3.4), and (3.5). The structure of the model, including the main effects of the profile factors, polynomials and interactions, is specified through functionals of the functions of the profile factors on multivariate indexing. The full k^{th} order FLM, defines polynomials and interactions up to and including the k^{th} order, such that,

$$\begin{aligned} y_i = \int_0^{\mathcal{T}} \boldsymbol{f}_1^T(\boldsymbol{x}_i(t_1))\, \boldsymbol{\beta}_1(t_1)\, dt_1 + \int_0^{\mathcal{T}}\int_0^{\mathcal{T}} \boldsymbol{f}_2^T(\boldsymbol{x}_i(t_1), \boldsymbol{x}_i(t_2))\, \boldsymbol{\beta}_2(t_1, t_2) \\ dt_1 dt_2 + \cdots + \int_0^{\mathcal{T}}\int_0^{\mathcal{T}} \cdots \int_0^{\mathcal{T}} \boldsymbol{f}_k^T(\boldsymbol{x}_i(t_1), \boldsymbol{x}_i(t_2), \ldots, \boldsymbol{x}_i(t_k)) \\ \boldsymbol{\beta}_k(t_1, t_2, \ldots, t_k)\, dt_1 dt_2 \cdots dt_k + \epsilon_i, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{9.1}$$

The $J \times 1$ vector $\boldsymbol{x}_i(t)$, that represents the functions of the profile factors at the i^{th} run of the experiment, is defined in (3.2). The functionals of the functions of the profile factors $\boldsymbol{f}_r^T(\boldsymbol{x}_i(t_1), \boldsymbol{x}_i(t_2), \ldots, \boldsymbol{x}_i(t_r))$, for $r = 1, 2, \ldots, k$, are functions that has to be specified. The specification of the functions $\boldsymbol{f}_r^T(\boldsymbol{x}_i(t_1), \boldsymbol{x}_i(t_2), \ldots, \boldsymbol{x}_i(t_r))$ forms the linear predictor equation with the functions of interest. In contrast to the univariate surface model in (3.1) for which a single functional is specified; for the model in (9.1) different functionals of the functions of the profile factors are specified in every dimension.

The function $\boldsymbol{f}_r(\boldsymbol{x}_i(t_1), \boldsymbol{x}_i(t_2), \ldots, \boldsymbol{x}_i(t_r))$ is a $Q_r \times 1$ vector, with Q_r the total number of terms in the r^{th} dimension integral in the model, for $r = 1, 2, \ldots, k$. Specifically, the function $\boldsymbol{f}_r(\boldsymbol{x}_i(t_1), \boldsymbol{x}_i(t_2), \ldots, \boldsymbol{x}_i(t_r))$ is the $Q_r \times 1$ vector defined as,

$$\boldsymbol{f}_r(\boldsymbol{x}_i(t_1), \boldsymbol{x}_i(t_2), \ldots, \boldsymbol{x}_i(t_r)) = \begin{pmatrix} f_{r1}(\boldsymbol{x}_i(t_1), \boldsymbol{x}_i(t_2), \ldots, \boldsymbol{x}_i(t_r)) \\ f_{r2}(\boldsymbol{x}_i(t_1), \boldsymbol{x}_i(t_2), \ldots, \boldsymbol{x}_i(t_r)) \\ \vdots \\ f_{rQ_r}(\boldsymbol{x}_i(t_1), \boldsymbol{x}_i(t_2), \ldots, \boldsymbol{x}_i(t_r)) \end{pmatrix} \tag{9.2}$$
$$i = 1, 2, \ldots, n, \; r = 1, 2, \ldots, k.$$

The total number of terms in the model is the addition of the number of terms in each dimension,

$$Q = \sum_{r=1}^{k} Q_r, \tag{9.3}$$

with $Q_1 = J$ if the single integral contains the main effects of the functions of the J profile factors. Additionally, an intercept is incorporated in the model through the function $\boldsymbol{f}_1(\boldsymbol{x}_i(t_1))$. If the intercept is included in the model, the first component of the vector of the function $\boldsymbol{f}_1(\boldsymbol{x}_i(t_1))$ is 1, i.e., $f_1(\boldsymbol{x}_i(t_1)) = 1$. The basis expansions for the profile factors, used to restrict the function space, are the same as in (3.12).

The functional parameters are represented through multivariate parameter functions $\boldsymbol{\beta}_r(t_1, t_2, \ldots t_r)$ for $r = 1, 2, \ldots, k$, that are $Q_r \times 1$ vectors. The number of indexing variables in each parameter function depends on the order of the polynomial or interaction it corresponds to. Specifically, the r^{th} parameter function is defined as,

$$\boldsymbol{\beta}_r(t_1, t_2, \ldots, t_r) = \begin{pmatrix} \beta_{r1}(t_1, t_2, \ldots, t_r) \\ \beta_{r2}(t_1, t_2, \ldots, t_r) \\ \vdots \\ \beta_{rQ_r}(t_1, t_2, \ldots, t_r) \end{pmatrix}. \tag{9.4}$$

For simplicity and without loss of generality, focus is given on the full 2^{nd} order FLM, as a special case of the model in (9.1). The full 2^{nd} order FLM is defined as,

$$\begin{aligned} y_i = \int_0^{\mathcal{T}} \boldsymbol{f}_1^T(\boldsymbol{x}_i(t_1))\, \boldsymbol{\beta}_1(t_1)\, dt_1 + \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} \boldsymbol{f}_2^T(\boldsymbol{x}_i(t_1), \boldsymbol{x}_i(t_2))\, \boldsymbol{\beta}_2(t_1, t_2) \\ dt_1 dt_2 + \epsilon_i, \quad i = 1, 2, \dots, n. \end{aligned} \tag{9.5}$$

To assist the understanding, assume a model with two profile factors, i.e., $J = 2$. In addition to the main effects of the profile factors, the model depends on the intercept, the interaction of the two profile factors, and the quadratic effect of the first profile factor. Thus, the linear predictor involves $1-$ and $2-$ dimensional integrals, i.e., $k = 2$. The total number of terms is $Q = Q_1 + Q_2 = 3 + 2 = 5$. The intercept and main effects of the profile factors are specified through,

$$\boldsymbol{f}_1^T(\boldsymbol{x}_i(t_1)) = \begin{pmatrix} 1 & x_{i1}(t_1) & x_{i2}(t_1) \end{pmatrix}, \quad i = 1, 2, \dots, n, \tag{9.6}$$

and the quadratic term and the $2-$way interaction are specified through,

$$\boldsymbol{f}_2^T(\boldsymbol{x}_i(t_1), \boldsymbol{x}_i(t_2)) = \begin{pmatrix} x_{i1}(t_1) x_{i1}(t_2) & x_{i1}(t_1) x_{i2}(t_2) \end{pmatrix}, \quad i = 1, 2, \dots, n. \tag{9.7}$$

For the example discussed above, the single and double parameter functions are defined as,

$$\begin{aligned} \boldsymbol{\beta}_1^T(t_1) &= \begin{pmatrix} \beta_{11} & \beta_{12}(t_1) & \beta_{13}(t_1) \end{pmatrix}, \\ \boldsymbol{\beta}_2^T(t_1, t_2) &= \begin{pmatrix} \beta_{21}(t_1, t_2) & \beta_{22}(t_1, t_2) \end{pmatrix}. \end{aligned}$$

The basis expansion of the univariate functional parameter is the same as in (3.11). A way to represent bivariate (and multivariate) functions is through the full tensor product of univariate basis; see De Boor (1978, p. 293). The bivariate tensor product basis for t_1, t_2 is defined over the 2-dimensional region $[0, \mathcal{T}] \times [0, \mathcal{T}]$. Each univariate basis is allowed a different choice of degree and a different choice of knots. However, for the bivariate parameter function below, identical basis are assumed. After that, the bivariate function can be represented as a linear combination of the basis functions of two basis, as in Fuchs et al. (2015),

$$\begin{aligned} \beta_{2q}(t_1, t_2) &= \sum_{l=1}^{n_{\beta,2q}} \sum_{m=1}^{n_{\beta,2q}} \theta_{lm} b_{2ql}(t_1) b_{2qm}(t_2) \\ &= \left[\boldsymbol{b}_{2q}(t_1) \otimes \boldsymbol{b}_{2q}(t_2)\right]^T \mathrm{vec}(\boldsymbol{\theta}_{2q}), \quad q = 1, 2, \dots, Q_2, \end{aligned} \tag{9.8}$$

with $\boldsymbol{b}_{2q}(t)^T = [b_{2q1}(t), b_{q2q2}(t), \dots, b_{2qn_{\beta,2q}}(t)]$ a $n_{\beta,2q} \times 1$ vector of known basis functions, and $\mathrm{vec}(\boldsymbol{\theta}_{2q})$ the $n^2_{\beta,2q} \times 1$ vectorisation of the $n_{\beta,2q} \times n_{\beta,2q}$ matrix $\boldsymbol{\theta}_{2q}$. The basis expansion of the the k-variate parameter function can be represented through the same approach, by k individual basis.

9.1.1 Methodology expansion

The basis expansions of the functions of the profile factors and of the functional parameters are substituted into the linear predictor of the model in (9.5). The aim is to simplify the linear predictor to a familiar form, as in (5.10). After that, the methodology of Chapter 5 is expanded to the bivariate index integrals, in an effort to derive closed form expressions. In matrix form, the FLM from (9.5) is given by,

$$\begin{aligned} \boldsymbol{y} = \int_0^{\mathcal{T}} \boldsymbol{f}_1^T(\boldsymbol{X}(t_1))\ \boldsymbol{\beta}_1(t_1)\ dt_1 + \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} \boldsymbol{f}_2^T(\boldsymbol{X}(t_1), \boldsymbol{X}(t_2))\ \boldsymbol{\beta}_2(t_1, t_2) \\ dt_1 dt_2 + \boldsymbol{\epsilon} \end{aligned} \tag{9.9}$$

for $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ the $n \times 1$ vector of responses, and $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$ the $n \times 1$ vector of independent error terms with mean zero and variance-covariance $\sigma^2 \boldsymbol{I}_n$. The matrix $\boldsymbol{X}(t_r)$ is a $n \times J$ matrix as defined in (5.2), for $t_r = t, r = 1, 2$. Subsequently, the function $\boldsymbol{f}_r^T(\boldsymbol{X}(t_1), \boldsymbol{X}(t_2), \ldots, \boldsymbol{X}(t_r))$ is a $n \times Q_r$ matrix,

$$\boldsymbol{f}_r(\boldsymbol{X}(t_1), \boldsymbol{X}(t_2), \ldots, \boldsymbol{X}(t_r)) = \begin{pmatrix} f_{r1}(\boldsymbol{X}(t_1), \boldsymbol{X}(t_2), \ldots, \boldsymbol{X}(t_r)) \\ f_{r2}(\boldsymbol{X}(t_1), \boldsymbol{X}(t_2), \ldots, \boldsymbol{X}(t_r)) \\ \vdots \\ f_{rQ_r}(\boldsymbol{X}(t_1), \boldsymbol{X}(t_2), \ldots, \boldsymbol{X}(t_r)) \end{pmatrix}, \quad r = 1, 2, \tag{9.10}$$

expanded further as in (5.4).

The functional parameters are expanded using the basis function expansions defined in (3.11) for the univariate parameter function, and (9.8) for the bivariate parameter function. The parameter function on single index is expanded as in (5.7),

$$\boldsymbol{\beta}_1(t_1) = \begin{pmatrix} \boldsymbol{b}_{11}^T(t)\boldsymbol{\theta}_{11} \\ \boldsymbol{b}_{12}^T(t)\boldsymbol{\theta}_{12} \\ \vdots \\ \boldsymbol{b}_{1Q_1}^T(t)\boldsymbol{\theta}_{1Q_1} \end{pmatrix} = \boldsymbol{b}_1^T(t)\boldsymbol{\theta}_1. \tag{9.11}$$

The vector $\boldsymbol{b}_1(t_1)$ is the $\sum_{q=1}^{Q_1} n_{\beta,q} \times 1$ vector containing the known basis functions from the expansion of the single index parameter function,

$$\boldsymbol{b}_1^T(t_1) = \begin{pmatrix} \boldsymbol{b}_{11}^T(t_1) & \boldsymbol{b}_{12}^T(t_1) & \cdots & \boldsymbol{b}_{1Q_1}^T(t_1) \end{pmatrix}, \tag{9.12}$$

and $\boldsymbol{\theta}_1$ is the $\sum_{q=1}^{Q_1} n_{\beta,q} \times 1$ vector of coefficients from the expansion of the functional parameters,

$$\boldsymbol{\theta}_1^T = \begin{pmatrix} \boldsymbol{\theta}_{11}^T & \boldsymbol{\theta}_{12}^T & \cdots & \boldsymbol{\theta}_{1Q_1}^T \end{pmatrix}. \tag{9.13}$$

The basis function expansion of the bivariate parameter function is defined as,

$$\boldsymbol{\beta}_2(t_1,t_2) = \begin{pmatrix} \left[\boldsymbol{b}_{21}(t_1) \otimes \boldsymbol{b}_{21}(t_2)\right]^T \mathrm{vec}(\boldsymbol{\theta}_{21}) \\ \left[\boldsymbol{b}_{22}(t_1) \otimes \boldsymbol{b}_{22}(t_2)\right]^T \mathrm{vec}(\boldsymbol{\theta}_{22}) \\ \vdots \\ \left[\boldsymbol{b}_{2Q_2}(t_1) \otimes \boldsymbol{b}_{2Q_2}(t_2)\right]^T \mathrm{vec}(\boldsymbol{\theta}_{2Q_2}) \end{pmatrix} = \left[\boldsymbol{b}_2(t_1) \otimes \boldsymbol{b}_2(t_2)\right]^T \boldsymbol{\theta}_2. \tag{9.14}$$

The vector $\boldsymbol{b}_2(t_1) \otimes \boldsymbol{b}_2(t_2)$ is the $\sum_{q=1}^{Q_2} n_{\beta,q}^2 \times 1$ vector containing the known basis functions from the expansion of the bivariate parameter function,

$$\boldsymbol{b}_2(t_1) \otimes \boldsymbol{b}_2(t_2) = \begin{pmatrix} \boldsymbol{b}_{21}(t_1) \otimes \boldsymbol{b}_{21}(t_2) \\ \boldsymbol{b}_{22}(t_1) \otimes \boldsymbol{b}_{22}(t_2) \\ \vdots \\ \boldsymbol{b}_{2Q_2}(t_1) \otimes \boldsymbol{b}_{2Q_2}(t_2) \end{pmatrix}, \tag{9.15}$$

and $\boldsymbol{\theta}_2$ is the $\sum_{q=1}^{Q_2} n_{\beta,q}^2 \times 1$ vector of coefficients,

$$\boldsymbol{\theta}_2^T = \begin{pmatrix} \mathrm{vec}(\boldsymbol{\theta}_{21})^T & \mathrm{vec}(\boldsymbol{\theta}_{22})^T & \cdots & \mathrm{vec}(\boldsymbol{\theta}_{2Q_2})^T \end{pmatrix}. \tag{9.16}$$

The basis function expansion of the k-variate parameter function is defined by expanding the above result accordingly.

The substitution of the basis expansions defined above, updates the model from (9.9) to,

$$\begin{aligned} \boldsymbol{y} &= \int_0^{\mathcal{T}} \boldsymbol{f}_1^T(\boldsymbol{X}(t_1))\, \boldsymbol{\beta}_1(t_1)\, dt_1 + \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} \boldsymbol{f}_2^T(\boldsymbol{X}(t_1), \boldsymbol{X}(t_2))\, \boldsymbol{\beta}_2(t_1,t_2) \\ &\quad dt_1 dt_2 + \boldsymbol{\epsilon} \\ &= \int_0^{\mathcal{T}} \boldsymbol{f}_1^T(\boldsymbol{X}(t_1))\, \boldsymbol{b}_1^T(t)\, dt_1\, \boldsymbol{\theta}_1 + \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} \boldsymbol{f}_2^T(\boldsymbol{X}(t_1), \boldsymbol{X}(t_2)) \\ &\quad \left[\boldsymbol{b}_2(t_1) \otimes \boldsymbol{b}_2(t_2)\right]^T dt_1 dt_2\, \boldsymbol{\theta}_2 + \boldsymbol{\epsilon} \\ &= \boldsymbol{Z}_1\boldsymbol{\theta}_1 + \boldsymbol{Z}_2\boldsymbol{\theta}_2 + \epsilon \\ &= \boldsymbol{Z}\boldsymbol{\theta} + \epsilon, \end{aligned} \tag{9.17}$$

with $\boldsymbol{Z}$ the $n \times \sum_{r=1}^{2} \sum_{q=1}^{Q_r} n_{\beta,q}^r$ model matrix and $\boldsymbol{\theta}$ the $\sum_{r=1}^{2} \sum_{q=1}^{Q_r} n_{\beta,q}^r \times 1$ vector of unknown parameters. The model matrix $\boldsymbol{Z}$ is the column bind of the matrices $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$,

$$\boldsymbol{Z} = \begin{pmatrix} \boldsymbol{Z}_1 & \boldsymbol{Z}_2 \end{pmatrix}, \tag{9.18}$$

with $\boldsymbol{Z}_1$ a $n \times \sum_{q=1}^{Q_1} n_{\beta,q}^1$ matrix, and $\boldsymbol{Z}_2$ a $n \times \sum_{q=1}^{Q_2} n_{\beta,q}^2$ matrix. Subsequently, $\boldsymbol{Z}_1$ is partitioned in Q_1 column blocks, with the q^{th} column block a $n \times n_{\beta,q}^1$ matrix $\boldsymbol{Z}_{1_{\cdot q}}$, which is the solution to the single integrals that depend on the main effects of the profile factors, derived in (5.14). The matrix $\boldsymbol{Z}_2$ is partitioned in Q_2 column blocks, with the q^{th} column block a $n \times n^2_{\beta,q}$ matrix $\boldsymbol{Z}_{2._q}$, which is the solution to integrals of the form,

$$\begin{aligned}
\boldsymbol{Z}_{2._q} &= \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} f_{2q}(\boldsymbol{X}(t_1), \boldsymbol{X}(t_2)) \left[\boldsymbol{b}_{2q}(t_1) \otimes \boldsymbol{b}_{2q}(t_2)\right]^T dt_1 dt_2 \\
&= \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} f_{2q}\Big(\boldsymbol{x}_{\cdot 1}(t_1) \quad \boldsymbol{x}_{\cdot 2}(t_1) \quad \cdots \quad \boldsymbol{x}_{\cdot J}(t_1),\ \boldsymbol{x}_{\cdot 1}(t_2) \quad \boldsymbol{x}_{\cdot 2}(t_2) \quad \cdots \quad \boldsymbol{x}_{\cdot J}(t_2)\Big) \\
&\qquad \left[\boldsymbol{b}_{2q}(t_1) \otimes \boldsymbol{b}_{2q}(t_2)\right]^T dt_1 dt_2 \\
&= \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} f_{2q}\Big(\boldsymbol{\Gamma}_1\boldsymbol{c}_1(t_1) \quad \boldsymbol{\Gamma}_2\boldsymbol{c}_2(t_1) \quad \cdots \quad \boldsymbol{\Gamma}_J\boldsymbol{c}_J(t_1),\ \boldsymbol{\Gamma}_1\boldsymbol{c}_1(t_2) \quad \boldsymbol{\Gamma}_2\boldsymbol{c}_2(t_2) \quad \cdots \quad \boldsymbol{\Gamma}_J\boldsymbol{c}_J(t_2)\Big) \\
&\qquad\qquad \left[\boldsymbol{b}_{2q}(t_1) \otimes \boldsymbol{b}_{2q}(t_2)\right]^T dt_1 dt_2, \quad q = 1, 2, \ldots, Q_2.
\end{aligned} \tag{9.19}$$

Hence, the model matrix $\boldsymbol{Z}$ is partitioned in $Q = Q_1 + Q_2$ column blocks. Closed form expression for single integrals that depend on the main effects of the profile factors are derived in (5.14). The more complicate integrals for the bivariate parameter functions and double integrals are tackled in the next section, for which closed form expressions are also derived, given the use of B-spline basis functions.

The $\sum_{r=1}^{2} \sum_{q=1}^{Q_r} n^r_{\beta,q} \times 1$ vector of unknown parameters $\boldsymbol{\theta}$ is divided in 2 sub-vectors,

$$\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \end{pmatrix}, \tag{9.20}$$

with $\boldsymbol{\theta}_1$ a vector of length $\sum_{q=1}^{Q_1} n^1_{\beta,q}$ and $\boldsymbol{\theta}_2$ a vector of length $\sum_{q=1}^{Q_2} n^2_{\beta,q}$.

9.1.2 Integrate on bivariate indexing

For the 2^{nd} order FLM, the model matrix $\boldsymbol{Z}$ is a $n \times \sum_{r=1}^{2} \sum_{q=1}^{Q_r} n^r_{\beta,q}$ matrix, that is the column bind of the matrices $\boldsymbol{Z}_1, \boldsymbol{Z}_2$, as in (9.18). As discussed in the previous section, $\boldsymbol{Z}_r$ is a $n \times \sum_{q=1}^{Q_r} n^r_{\beta,q}$ matrix, partitioned in Q_r column blocks, with the q^{th} column block a $n \times n^r_{\beta,q}$ matrix $\boldsymbol{Z}_{r._q}$, for $r = 1, 2$. The Q_1 column blocks from the matrix $\boldsymbol{Z}_1$, result from single index integrals, exactly as in (5.14). Subsequently, the Q_2 column blocks from the matrix $\boldsymbol{Z}_2$ result from bivariate integrals, with every partition the solution to an integral of the form in (9.19).

A column block $\boldsymbol{Z}_{2 \cdot q}$ of a 2-way interaction of the profile factors $\boldsymbol{x}_{\cdot j_1}(t)$ and $\boldsymbol{x}_{\cdot j_2}(t)$, is the solution to an integral of the form,

$$\begin{aligned}
\boldsymbol{Z}_{2 \cdot q} &= \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} f_{2q}(\boldsymbol{X}(t_1), \boldsymbol{X}(t_2)) \left[\boldsymbol{b}_{2q}(t_1) \otimes \boldsymbol{b}_{2q}(t_2)\right]^T dt_1 dt_2 \\
&= \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} \boldsymbol{x}_{\cdot j_1}(t_1) \circ \boldsymbol{x}_{\cdot j_2}(t_2) \left[\boldsymbol{b}_{2q}(t_1) \otimes \boldsymbol{b}_{2q}(t_2)\right]^T dt_1 dt_2 \\
&= \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} \boldsymbol{\Gamma}_{j_1} \boldsymbol{c}_{j_1}(t_1) \circ \boldsymbol{\Gamma}_{j_2} \boldsymbol{c}_{j_2}(t_2) \left[\boldsymbol{b}_{2q}(t_1) \otimes \boldsymbol{b}_{2q}(t_2)\right]^T dt_1 dt_2 \\
&= \boldsymbol{\Gamma}_{j_1 j_2} \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} \boldsymbol{c}_{j_1}(t_1)\, \boldsymbol{b}_{2q}(t_1) \;\otimes\; \boldsymbol{c}_{j_2}(t_2)\, \boldsymbol{b}_{2q}(t_2)\, dt_1 dt_2 \\
&= \boldsymbol{\Gamma}_{j_1 j_2} \left\{ \int_0^{\mathcal{T}} \boldsymbol{c}_{j_1}(t_1)\, \boldsymbol{b}_{2q}(t_1)\, dt_1 \;\otimes\; \int_0^{\mathcal{T}} \boldsymbol{c}_{j_2}(t_2)\, \boldsymbol{b}_{2q}(t_2)\, dt_2 \right\}, \\
&\quad q = 1, 2, \ldots, Q_2,
\end{aligned} \tag{9.21}$$

using properties of the Kronecker product. The matrix $\boldsymbol{\Gamma}_{j_1 j_2}$ is a $n \times \prod_{j=j_1}^{j_2} n_{x,j}$ coefficient matrix for which each column is the Hadamard product of columns of the coefficients matrices $\boldsymbol{\Gamma}_{j_1}$ and $\boldsymbol{\Gamma}_{j_2}$ from the basis function expansions of the profile factors that are involved in the interaction; see Section 5.3 for a detailed description. The Hadamard product of the functions of the profile factors, $\boldsymbol{x}_{\cdot j_1}(t_1) \circ \boldsymbol{x}_{\cdot j_2}(t_2)$, represents a 2-way interaction of the functions of the profile factors $\boldsymbol{x}_{\cdot j_1}(t_1)$ and $\boldsymbol{x}_{\cdot j_2}(t_2)$ if $j_1 \neq j_2$, or a quadratic polynomial if $j_1 = j_2$. The resulting expression in (9.21) is the Kronecker product of two univariate integrals, depending on the product of the basis functions of the profile factor and the parameter. For the choice of B-spline basis, closed form expressions for integrals of the form in (9.21) are discussed in Section 5.9.

9.1.3 Further investigation on the full higher order FLM

Functional models involving bivariate parameter functions and double integrals have been preliminary investigated. The linear predictor is still an extension to the linear predictor of the linear model. The model matrix is made up of column block matrices, with the number of column blocks being the order of the model. Subsequently, each column block is further partitioned in more column blocks, depending on the number of terms for each order. To derive the column blocks, the linear predictor of each term is integrated with respect to time. The main effects are integrated on single indexing, quadratic effects and 2-way interactions on double indexing. The methodology using basis functions can be expanded and adjusted to the bivariate parameter functions. As a result, it has been demonstrated that for the choice of B-spline basis, there exist closed form expressions for the double integrals. Equivalently, this can be expanded to k-order polynomials or k-way interactions, integrated on k-variate indexing.

In the future, it is of interest to investigate the full 2^{nd} (and higher) order FLM further, develop examples and compare the optimal designs against the designs derived from single integrals. Full higher order FGLMs are also going to be investigated. Additionally, it is of future interest to expand the Bayesian approach using roughness penalties from Chapter 5, to the full model. This will allow penalisation of the complicated multivariate parameter functions, to identify Bayesian optimal designs and investigate the effect of the value of the smoothing parameter.

To conclude with, the `R` package **fdesigns**, discussed in Chapter 8, is used to identify optimal designs for FLMs and FGLMs with univariate parameter functions and single integrals. The `R` package **fdesigns** will be updated in the future, to include functions capable of identifying optimal designs for functional multivariate models.

9.2 Prior choice effect to optimal designs for FGLMs

The optimal designs identified for the FGLMs are slightly more complicated, but in general similar to the designs identified for the FLMs. This mainly occurs due to the choice of zero mean prior. A difference between designs for FLMs and FGLMs is that the latter needs a prior. In the future, it is of interest to assess the effect of the prior specification to the optimal designs identified for FGLMs. After that, the characteristics and the complexity of the optimal designs can be compared to the optimal designs with zero prior mean and the FLM optimal designs.

As preliminary investigation, the $n = 8$ run FGLM, for linear parameter basis, $n_{x,1} = 8$, and prior mean equal to five for all parameters, i.e.,

$$\theta_1 \sim N(5,1), \quad \theta_{21} \sim N(5,1), \quad \theta_{22} \sim N(5,1),$$

is reconsidered. The A-optimality value of the optimal design identified is 100.062. The functions of the optimal design have at most two changes; in contrast to the optimal design for zero prior mean that have at most one change; see Figure 7.1 and Figure 9.1. Similarly, the $n = 12$ run FGLM, for quadratic parameter basis, $n_{x,1} = 8$, and prior mean equal to five for all parameters is reconsidered. Thus, in addition to the prior distribution of the parameters above,

$$\theta_{23} \sim N(5,1).$$

The A-optimality value of the optimal design identified is 726.715. The functions of the optimal design have at most three changes; in contrast to the optimal design for zero prior mean that change twice at most; see Figure 7.2 and Figure 9.2.

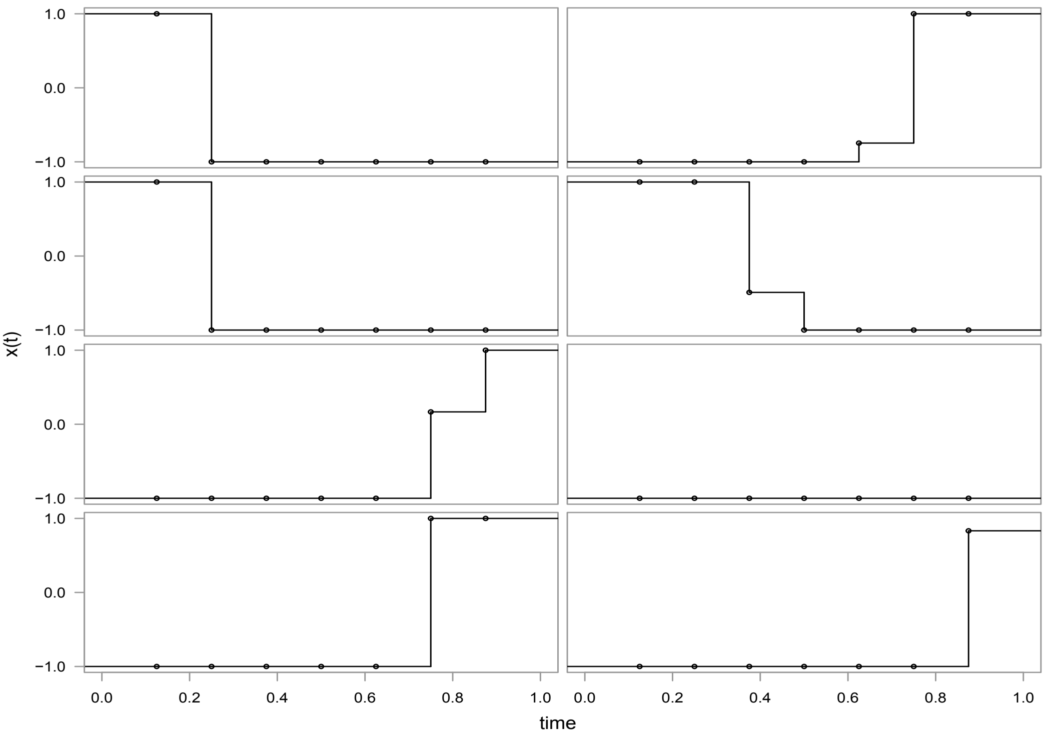


FIGURE 9.1: Eight run pseudo-Bayesian A-optimal design for $n_{x,1} = 8$ for $x_{\bullet 1}(t)$, linear basis for $\beta_2(t)$, and non-zero prior mean, for the functional logistic model.

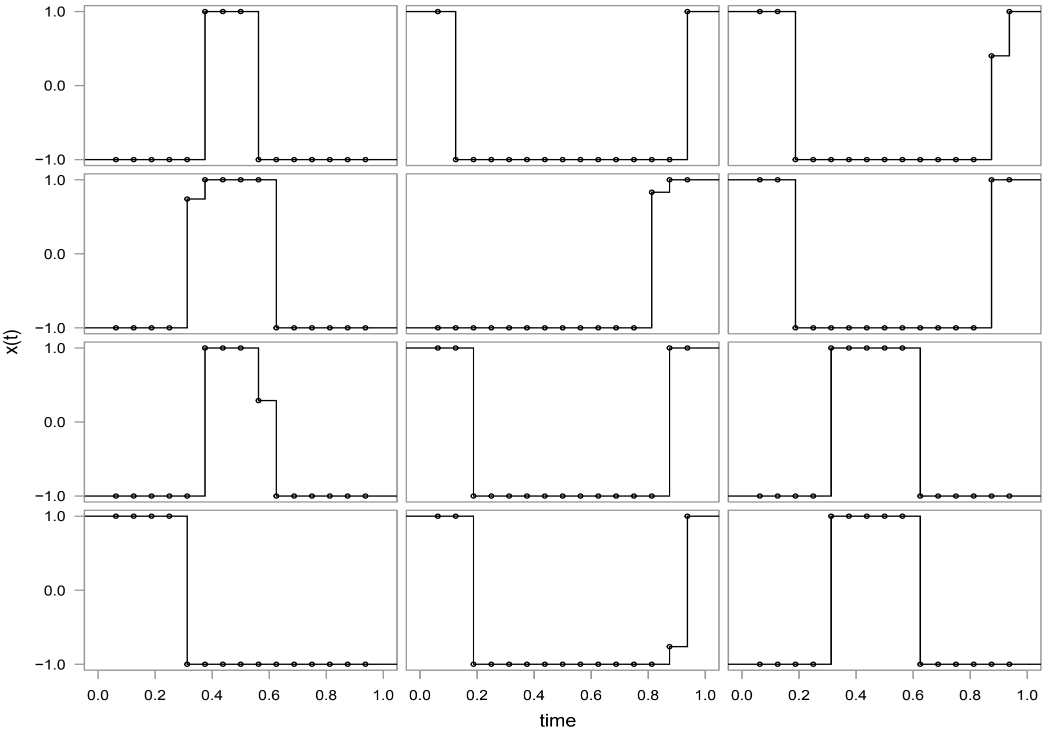


FIGURE 9.2: 12 run pseudo-Bayesian A-optimal design for $n_{x,1} = 16$ for $x_{\bullet 1}(t)$, quadratic basis for $\beta_2(t)$, and non-zero prior mean, for the functional logistic model.

Further investigation using prior specifications from the normal and uniform distributions, and expectations approximated using the Monte Carlo or quadrature methods is a future goal.

9.3 Alternative basis

In the thesis, focus has been given on the use of B-splines to expand the functions of the profile factors. For the functions of the parameters, focus has been given on power series and B-splines. The choice of the B-spline basis for the profile factors has been beneficial through its properties and computational efficiency. Moreover, for B-spline basis, the integrals with respect to time have been solved in closed form. Having said that, the basis to use is not always a straightforward choice. A good basis depends on the the problem in hand, the nature of the experiment, and the profile factors; see Georgakis (2013). In future work, it is of interest to consider alternative basis, especially orthogonal basis. An orthogonal basis, is a basis obtained from an orthonormal basis or Hilbert space, and by definition the inner product of orthogonal basis is zero; see Heuberger et al. (2005, Chapter 1). This increases the computational efficiency, since calculations are less complex; see Georgakis (2013).

The B-spline basis is not orthogonal, however, any basis can be orthogonalised. In fact, there exist orthogonalisation methods, and specifically methods that are natural for the B-splines; see Liu et al. (2019). This means that, the basis keep most of its original structure, and the computational efficiency increases due to the orthogonality property. The orthogonalised B-splines are usually refereed to as O-splines or OB-splines. As future work, it is of interest to consider the one-sided and two-sided methods discussed by Mason et al. (1993) and Liu et al. (2019), to orthogonalise the B-spline basis and adjust them to the methodology of the thesis.

After that, other properties including normality and rotatability, discussed in Roche (2018), will be considered.

9.4 Design of experiments for functional ODE systems

Differential equations are equations that involve derivatives of one or more functions. They are used to provide information on how a system is changing, with respect to controllable factors; see Braun and Golubitsky (1983, Chapter 3). Types of differential equations are the ordinary differential equations (ODEs) and the partial differential equations (PDEs); see Ince (1956). ODE systems involve derivatives with respect to a single controllable factor; see Miller and Michel (2014, Chapter 1), in contrast to PDEs that involve derivatives with respect to multiple factors; see Renardy and Rogers (2006, Chapter 1). ODE systems are used in many disciplines, including engineering, biology, medicine and physics.

A future interest is to consider time varying models with profile factors, defined by a system of ODEs and develop a methodology that allows optimal designs to be found. Time-varying systems are systems which describe the change of variables over time, by linking the output derivatives to itself and controllable (functional) factors. Meaning that, a solution to the ODEs represents the evolution of the system output variables, referred to as states, with respect to time and the inputs. For instance, an ODE system depending on profile factors is formulated as,

$$\begin{aligned} \dot{\boldsymbol{y}}(t) &= f(\boldsymbol{y}(t), \boldsymbol{x}(t), t|\boldsymbol{\beta}), \quad 0 \leq t \leq T \\ \boldsymbol{y}(0) &= \boldsymbol{y}_0 \end{aligned} \tag{9.22}$$

with $\boldsymbol{y}(t)$ and $\dot{\boldsymbol{y}}(t)$ the output and output's first derivative, with the output factors also called state factors. This is to inflate the fact that they represent the mathematical state of a time-varying model. Additionally, $\boldsymbol{x}(t)$ are the profile factors, $\boldsymbol{\beta}$ are the unknown parameters defining the system, and $\boldsymbol{y}_0$ are the initial conditions. Time-varying models defined by a system of ODEs (or PDEs) are rather challenging due to high dimensional and analytically intractable integrals. Thus, solutions usually need to be approximated using numerical methods; see Ramsay et al. (2007) and Bock et al. (2013). In order to be able to fit such models, the unknown parameters $\boldsymbol{\beta}$ need to be estimated, and if the initial conditions are unknown, they are also included in the vector of unknown parameters.

For static factors, a design problem for time-varying systems is to choose the most informative combination of time points and input values, at which observations of the states should take place; see Overstall et al. (2019). Suppose a system of ODEs as in (9.22), with s state variables, J input factors, n number of runs, and $\boldsymbol{y}_0$ the initial conditions. The design problem is the choice of the static factors,

$$\boldsymbol{x}_i^T = (x_{i1}, x_{i2} \ldots, x_{iJ}), \quad i = 1, 2, \ldots, n,$$

and the time points, that minimise or maximise an objective function.

In the case of profile factors, the design problem becomes to appropriately choose the time points, and the functions of the profile factors,

$$\boldsymbol{x}_i(t)^T = (x_{i1}(t), x_{i2}(t) \ldots, x_{iJ}(t)), \quad i = 1, 2, \ldots, n,$$

Parameters are scalar, thus, restrictions would need to be imposed only on the function space of the profile factors.

Often, knowledge about the unknown parameters is not available; see Atkinson et al. (2007, Chapter 17), thus, sensitivity analysis to derive the sensitivity equations is needed. The sensitivity equations are the partial derivatives of the responses with respect to the unknown parameters and they represent the sensitivity of the responses to changes in the parameters, see Dickinson and Gelinas (1976). For instance, if small changes in a parameter lead to a large effect in the solution, the parameter is said to be sensitive, and non-sensitive if otherwise. In this way, sensitivity analysis leads to the construction of the Fisher information matrix.

The example discussed below, to motivate ODEs with profile factors, is the tank reactor equations, that is discussed in Ramsay et al. (2007).

Tank reactor equations

A motivating example of a system of ODEs, that depends on profile factors, is the tank reactor equations in Ramsay et al. (2007). Tank reactor is a common model for a chemical reactor in chemical engineering. A tank reactor is subdivided into a cooling jacket and a tank, with the jacket surrounding the tank. Also, there exists an impeller stirring the contents of the tank. A fluid enters the tank with concentration $C_{in}(t)$, temperature $T_{in}(t)$ and flow rate $F_{in}(t)$. After that, a reaction produces another product which exits the tank; with concentration and temperature C and T respectively. In the cooling jacket that surrounds the tank, there exists a coolant which has temperature $T_{co}(t)$ and flow rate $F_{co}(t)$. Moreover, the volume of the tank is set to unit. The tank reactor procedure is modelled by the following system of ODEs,

$$\begin{aligned}\dot{C}(t) &= -\beta_{CC}(T(t), F_{in}(t))C(t) + F_{in}(t)C_{in}(t),\\ \dot{T}(t) &= -\beta_{TT}(F_{co}(t), F_{in}(t))T(t) + \beta_{TC}(T(t), F_{in}(t))C(t)\\ &\quad + F_{in}(t)T_{in}(t) + \alpha(F_{co}(t))T_{co}(t),\end{aligned} \tag{9.23}$$

with the functions $\beta_{CC}, \beta_{TT}, \beta_{TC}$ and α being weight functions,

$$\begin{aligned}\beta_{CC}(T(t), F_{in}(t)) &= \kappa \exp\big\{-10^4\tau(1/T(t) - 1/T_{ref})\big\} + F_{in}(t),\\ \beta_{TT}(F_{co}(t), F_{in}(t)) &= \alpha(F_{co}(t)) + F_{in}(t),\\ \beta_{TC}(T(t), F_{in}(t)) &= 130\beta_{CC}(T(t), F_{in}(t)),\\ \alpha(F_{co}(t)) &= \frac{aF_{co}(t)^{b+1}}{F_{co}(t) + aF_{co}(t)^b/2},\end{aligned} \tag{9.24}$$

and $t \in (0, 60)$. $\dot{C}(t)$ and $\dot{T}(t)$ correspond to the first derivatives of concentration $C(t)$ and temperature $T(t)$, respectively. The time-varying multipliers are linked with five profile factors $C_{in}(t)$, $T_{in}(t)$, $F_{in}(t)$, $T_{co}(t)$ and $F_{co}(t)$, a reference temperature T_{ref} and depends on four unknown parameters $\boldsymbol{\beta} = (\kappa, \tau, a, b)$. Note that, the factor 10^4 is used to rescale the parameter τ and guarantee that the four parameters lie within the range [0.4,1.8]. According to Ramsay et al. (2007), the parameter b is kept fixed at 0.5 because an accurate estimation of all four parameters of the model is not attainable.

The experimental aim is to understand the dynamics of concentration $C(t)$ and temperature $T(t)$, as determined by the five profile factors $C_{in}(t), T_{in}(t), F_{in}(t), T_{co}(t)$ and $F_{co}(t)$ over time. Meaning that, the design of experiments problem is to find the right functions of the profile factors and the time points at which the observations should take place. Moreover, there are no initial conditions for concentration and temperature defined as $C(0)$ and $T(0)$, thus, they become part of the problem. The high number of profile factors, parameters, and the unknown initial conditions, make the design of experiments problem for the tank reactor model challenging.

9.5 Design of experiments for functional non-linear models

In the real world, often data are modelled by functions which are non-linear in the model parameters; see Seber and Wild (1989, Chapter 1). Such models are known as non-linear models and they describe a non-linear relationship. As a result, optimal experimental designs for non-linear models also depend on the values of the unknown model parameters; see Atkinson et al. (2007, Chapter 17).

A non-linear model, is an extension to the linear model in (1.1), and it takes the form,

$$y_i = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{\beta}) + \epsilon_i, \quad i = 1, 2, \ldots, n.$$

with $y_i, \boldsymbol{x}_i$ and $\boldsymbol{\beta}$ as in (1.1). The function $\boldsymbol{f}$ is a non-linear function of the controllable factors and the vector of the parameters. The reason it is non-linear, its that it cannot be represented as a linear combination of the parameters. For example, a non-linear model is,

$$y_i = x_{i1}\beta_1 + \exp(x_{i2}\beta_2^2), \quad i = 1, 2, \ldots, n.$$

The function $\boldsymbol{f}$ specifies the structure of the model. It can often be exponential, logarithmic, etc.

A future interest is to consider non-linear models with profile factors, i.e., functional non-linear models, and expand the methodology of this thesis to identify optimal designs for this more complicated models. A functional non-linear model with scalar responses collected, takes the form,

$$y_i = \boldsymbol{f}(\boldsymbol{x}_i(t), \boldsymbol{\beta}(t)) + \epsilon_i, \quad i = 1, 2, \ldots, n,$$

with $\boldsymbol{x}_i(t)$ the profile factors at the i^{th} run, and $\boldsymbol{\beta}(t)$ the functional parameters. The function $\boldsymbol{f}$ is a non-linear function, as before. This model form is rather challenging and analytically intractable, thus, numerical approximations are essential to identify optimal designs; see Atkinson et al. (2007, Chapter 17). Knowledge of the parameters can be passed through the use of prior information, as in FGLMs in Chapter 7. Subsequently, the intractable integrals involved in identifying the optimal experimental designs can be approximated using the numerical methods, MC and quadrature, also discussed in Chapter 7.

Functional non-linear models have been addressed in the past by Kadri et al. (2010), using a reproducing kernel Hilbert space, but in my knowledge there is no work addressing experimental designs for functional non-linear models using a parametric approach.

Appendix A

B-spline basis as differences of the Truncated Power Functions

This section describes the connection between the B-splines and the truncated power functions. As discussed in Section 4.4, the BS basis are efficiently computed by the use of the Cox-de Boor recursion formula; see (4.13). This formula comes from the expansion of the formal definition of a normalised BS which depends on differences. In fact, the degree d BS basis can be defined as scaled differences of the TPF with equally spaced knots; see De Boor (1978, page 87) and Eilers and Marx (2010). Thus, the BS and TPS basis systems belong to the same family of basis, and for every TPS basis there exists an equivalent BS basis in the same space of functions; see Friedman et al. (2001, Chapter 6).

For the BS basis computation using differences, it is essential to have $d+1$ extra TPFs, achieved by extending the knot vector. The extended knot vector for the TPF has $2d+2$ extra knots, i.e., $d+1$ on the left and $d+1$ on the right. The extra knots are still equally spaced and can go beyond the interval 0 to $\mathcal{T}$, even negative. The extended knot vector for the TPF is,

$$\tilde{\lambda} = \lambda_{d^-}, ..., \lambda_0, \lambda_1, ..., \lambda_k, \lambda_{k+1}, ..., \lambda_{k+d+1}.$$

Without loss of generality, assume having a degree zero TPF with equally spaced knots in the time interval 0 to $\mathcal{T}$. The BS basis of degree zero is defined as TPF differences as,

$$B^0_\kappa(t) = \Upsilon^0_\kappa(t) - \Upsilon^0_{\kappa+1}(t) = -\Delta\Upsilon^0_\kappa(t), \tag{A.1}$$

with $\Upsilon^0_\kappa(t)$ the κ^{th} function of the degree zero TPF and $\Delta\Upsilon^0_\kappa(t)$ the difference of the κ^{th} and $(\kappa+1)^{th}$ functions.

For example, assume that it is desired to compute a degree zero BS basis with two basis functions from the choice of a single knot at $t = 0.5$. The extended knot vector for the

TPF is the size three vector $\tilde{\lambda} = (0, 0.5, 1)$ returning three basis functions. The first BS basis function is the difference of the first and second TPFs, and the second BS basis is the difference of the second and third TPFs. As can be seen in Figure A.1, the difference of the two basis functions in plot (a) are equivalent to the first BS basis function in plot (b) that is found using the recursion formula.

The differences notation Δ, is expanded to degree one differences, for instance,

$$\Delta^2 Y^1_{\kappa+1}(t) = Y^1_{\kappa+1}(t) - 2Y^1_{\kappa+2}(t) + Y^1_{\kappa+3}(t),$$

and so on, for higher degrees. Using properties of the BS basis and expanding the idea of using TPS basis differences to calculate BS basis, a general formula for BS basis of degree d is,

$$B_{\kappa,d}(t) = (-1)^{d+1}\Delta^{d+1}\frac{Y^d_{\kappa+d}(t)}{\lambda_{dist} d!}, \tag{A.2}$$

for λ_{dist} being the distance between the knots. For more information on TPF differences; see Eilers and Marx (2010), De Boor (1978, Chapter 9) and Binder and Sauerbrei (2008).

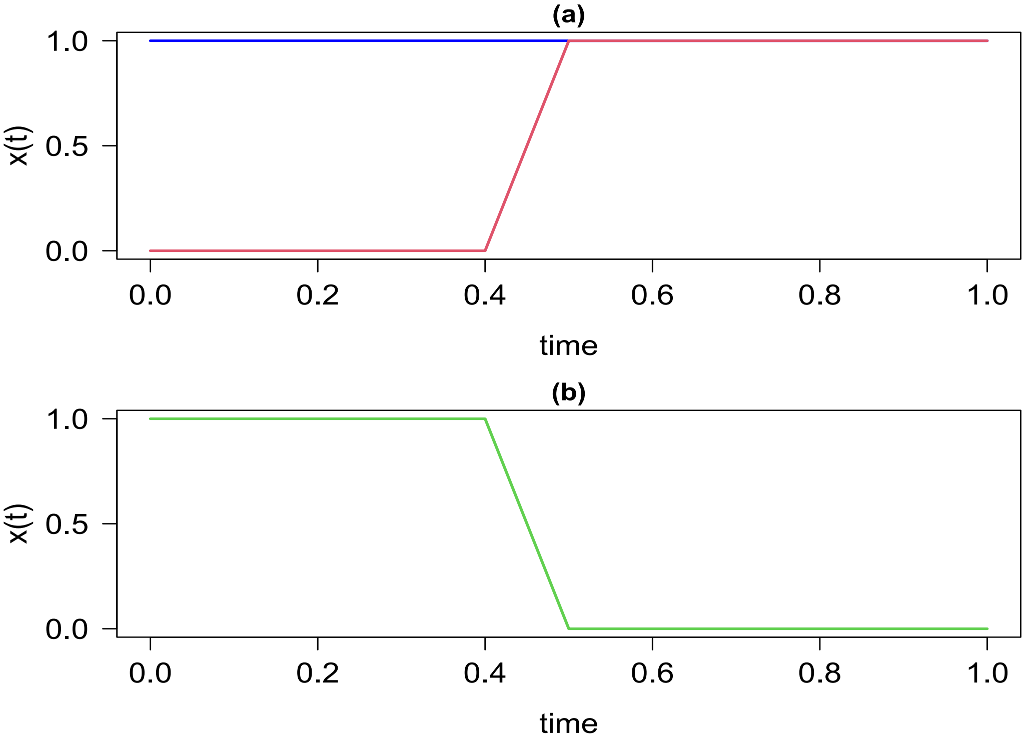


FIGURE A.1: (a) First and second TPF basis functions of degree zero, equally spaced knots and extended knot vector and (b) First BS basis function of degree zero and single knot at $t = 0.5$, computed using the recursion formula.

Appendix B

Marginal distributions

B.1 Marginal distribution for $\boldsymbol{y}$ conditional on $\boldsymbol{\beta}$

The marginal distribution for $\boldsymbol{y}$ conditional on $\boldsymbol{\beta}$ is obtained from an application of the conditional probability rule from Section 2.3 and then integrating out σ^2,

$$\begin{aligned}
\pi(\boldsymbol{y}|\boldsymbol{\beta}) &= \int \pi(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\pi(\sigma^2)\ d\sigma^2 \\
&= \int (2\pi)^{-\frac{n}{2}}(\sigma^2)^{-\frac{n}{2}}\ \exp\Big[-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{F\beta})^T(\boldsymbol{y}-\boldsymbol{F\beta})\Big] \\
&\qquad \times \frac{(b/2)^{\frac{a}{2}}}{\Gamma(\frac{a}{2})}(\sigma^2)^{-(\frac{a}{2}+1)}\ \exp\Big[-\frac{b}{2\sigma^2}\Big]\ d\sigma^2 \\
&= \frac{(2\pi)^{-\frac{n}{2}}(b/2)^{\frac{a}{2}}}{\Gamma(\frac{a}{2})}\int (\sigma^2)^{-(\frac{a+n}{2})-1}\exp\Big[-\frac{(\boldsymbol{y}-\boldsymbol{F\beta})^T(\boldsymbol{y}-\boldsymbol{F\beta})+b}{2\sigma^2}\Big]\ d\sigma^2 \\
&= \frac{(2\pi)^{-\frac{n}{2}}(b/2)^{\frac{a}{2}}}{\Gamma(\frac{a}{2})}\frac{\Gamma(\frac{a+n}{2})}{\Big[b+(\boldsymbol{y}-\boldsymbol{F\beta})^T(\boldsymbol{y}-\boldsymbol{F\beta})\Big]^{\frac{a+n}{2}}} \qquad\qquad (B.1)\\
&\propto \Big[b+(\boldsymbol{y}-\boldsymbol{F\beta})^T(\boldsymbol{y}-\boldsymbol{F\beta})\Big]^{-(\frac{a+n}{2})} \\
&\propto \Big[1+\frac{1}{b}(\boldsymbol{y}-\boldsymbol{F\beta})^T(\boldsymbol{y}-\boldsymbol{F\beta})\Big]^{-(\frac{a+n}{2})} \\
&= \Bigg[1+\frac{1}{a}(\boldsymbol{y}-\boldsymbol{F\beta})^T\Big[\frac{b}{a}\boldsymbol{I}_n\Big]^{-1}(\boldsymbol{y}-\boldsymbol{F\beta})\Bigg]^{-(\frac{a+n}{2})}.
\end{aligned}$$

Hence, the marginal distribution for $\boldsymbol{y}$ conditional on $\boldsymbol{\beta}$ is a multivariate t-distribution,

$$\boldsymbol{y}|\boldsymbol{\beta} \sim t_a\Big(\boldsymbol{F\beta}, \frac{b}{a}\boldsymbol{I}_n\Big), \qquad (B.2)$$

with a degrees of freedom, mean $\boldsymbol{F\beta}$ and scale $\frac{b}{a}\boldsymbol{I}_n$.

B.2 Marginal distribution of $\boldsymbol{y}$ conditional on σ^2

The marginal distribution of $\boldsymbol{y}$ conditional on σ^2 makes sense to argue that follows a normal distribution, using that $\boldsymbol{y}|\boldsymbol{\beta},\sigma^2 \sim \mathcal{N}(\boldsymbol{F}\boldsymbol{\beta},\sigma^2\boldsymbol{I}_n)$ and $\boldsymbol{\beta}|\sigma^2 \sim \mathcal{N}(\boldsymbol{\mu},\sigma^2\boldsymbol{V})$. To derive analytically the mean and variance, the law of total expectation and the law of total variance fromWeiss (2006) are used to get,

$$\begin{aligned}\mathbb{E}_{\beta}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2) &= \mathbb{E}_{\beta}\left[\mathbb{E}_{y|\beta}(\boldsymbol{y}|\boldsymbol{\beta})\right]\\ &= \mathbb{E}_{\beta}(\boldsymbol{F}\boldsymbol{\beta})\\ &= \boldsymbol{F}\boldsymbol{\mu},\end{aligned} \tag{B.3}$$

and

$$\begin{aligned}\mathrm{Var}_{\beta}(\boldsymbol{y}) &= \mathbb{E}_{\beta}\left[\mathrm{var}_{y|\beta}(\boldsymbol{y}|\boldsymbol{\beta})\right] + \mathrm{Var}_{\beta}\left[\mathbb{E}_{y|\beta}(\boldsymbol{y}|\boldsymbol{\beta})\right]\\ &= \mathbb{E}_{\beta}(\sigma^2\boldsymbol{I}_n) + \mathrm{Var}_{\beta}(\boldsymbol{F}\boldsymbol{\beta})\\ &= \sigma^2\boldsymbol{I}_n + \sigma^2\boldsymbol{F}\boldsymbol{V}\boldsymbol{F}^T\\ &= \sigma^2(\boldsymbol{I}_n + \boldsymbol{F}\boldsymbol{V}\boldsymbol{F}^T).\end{aligned} \tag{B.4}$$

Hence, the marginal distribution of y conditional on σ^2 is a normal distribution,

$$\boldsymbol{y}|\sigma^2 \sim \mathrm{N}\Big(\boldsymbol{F}\boldsymbol{\mu},\sigma^2(\boldsymbol{I}_n + \boldsymbol{F}\boldsymbol{V}\boldsymbol{F}^T)\Big). \tag{B.5}$$

B.3 Marginal distribution of $\boldsymbol{y}$

The marginal distribution of $\boldsymbol{y}$ is obtained by considering that $\boldsymbol{y}|\sigma^2 \sim \mathcal{N}(\boldsymbol{F\mu}, \sigma^2(\boldsymbol{I}_n + \boldsymbol{FVF}^T))$ and $\sigma^2 \sim IG(a/2, b/2)$ and then integrating out σ^2,

$$\begin{aligned}
\pi(\boldsymbol{y}) &= \int \pi(\boldsymbol{y}|\sigma^2)\pi(\sigma^2)\, d\sigma^2 \\
&= \int (2\pi)^{-\frac{n}{2}}(\sigma^2)^{-\frac{n}{2}}|\boldsymbol{I}_n + \boldsymbol{FVF}^T|^{-\frac{1}{2}} \\
&\qquad \times \exp\Big[-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{F\mu})^T(\boldsymbol{I}_n + \boldsymbol{FVF}^T)^{-1}(\boldsymbol{y}-\boldsymbol{F\mu})\Big] \\
&\qquad\qquad \times \frac{(b/2)^{\frac{a}{2}}}{\Gamma(\frac{a}{2})}(\sigma^2)^{-(\frac{a}{2}+1)} \exp\Big[-\frac{b}{2\sigma^2}\Big]\, d\sigma^2 \\
&= \frac{(2\pi)^{-\frac{n}{2}}|\boldsymbol{I}_n + \boldsymbol{FVF}^T|^{-\frac{1}{2}}(b/2)^{\frac{a}{2}}}{\Gamma(\frac{a}{2})} \\
&\qquad \times \int (\sigma^2)^{-(\frac{a+n}{2})-1} \exp\Big[-\frac{(\boldsymbol{y}-\boldsymbol{F\mu})^T(\boldsymbol{I}_n + \boldsymbol{FVF}^T)^{-1}(\boldsymbol{y}-\boldsymbol{F\mu}) + b}{2\sigma^2}\Big]\, d\sigma^2 \\
&= \frac{(2\pi)^{-\frac{n}{2}}|\boldsymbol{I}_n + \boldsymbol{FVF}^T|^{-\frac{1}{2}}(b/2)^{\frac{a}{2}}}{\Gamma(\frac{a}{2})} \\
&\qquad \times \frac{\Gamma(\frac{a+n}{2})}{\Big[b + (\boldsymbol{y}-\boldsymbol{F\mu})^T(\boldsymbol{I}_n + \boldsymbol{FVF}^T)^{-1}(\boldsymbol{y}-\boldsymbol{F\mu})\Big]^{\frac{a+n}{2}}} \\
&\propto \Big[b + (\boldsymbol{y}-\boldsymbol{F\mu})^T(\boldsymbol{I}_n + \boldsymbol{FVF}^T)^{-1}(\boldsymbol{y}-\boldsymbol{F\mu})\Big]^{-(\frac{a+n}{2})} \\
&\propto \Big[1 + \frac{1}{b}(\boldsymbol{y}-\boldsymbol{F\mu})^T(\boldsymbol{I}_n + \boldsymbol{FVF}^T)^{-1}(\boldsymbol{y}-\boldsymbol{F\mu})\Big]^{-(\frac{a+n}{2})} \\
&= \Bigg[1 + \frac{1}{a}(\boldsymbol{y}-\boldsymbol{F\mu})^T\Big[\frac{b}{a}(\boldsymbol{I}_n + \boldsymbol{FVF}^T)\Big]^{-1}(\boldsymbol{y}-\boldsymbol{F\mu})\Bigg]^{-(\frac{a+n}{2})}.
\end{aligned} \tag{B.6}$$

Hence, the marginal distribution for $\boldsymbol{y}$ is a multivariate t-distribution,

$$\boldsymbol{y} \sim t_a\Big(\boldsymbol{F\mu}, \frac{b}{a}(\boldsymbol{I}_n + \boldsymbol{FVF}^T)\Big), \tag{B.7}$$

with a degrees of freedom, mean $\boldsymbol{F\mu}$ and scale $\frac{b}{a}(\boldsymbol{I}_n + \boldsymbol{FVF}^T)$.

Appendix C

Methodology example proof

In this appendix, the aim is to provide a proof through an example for (5.23) that is,

$$\begin{aligned}\boldsymbol{Z}_{\bullet q} &= \boldsymbol{\Gamma}_{j_1 j_2 \cdots j_K} \int_0^{\mathcal{T}} \left(\boldsymbol{c}_{j_1}(t) \otimes \boldsymbol{c}_{j_2}(t) \otimes \cdots \otimes \boldsymbol{c}_{j_K}(t)\right) \boldsymbol{b}_q^T(t)\, dt \\ &= \boldsymbol{\Gamma}_{j_1 j_2 \cdots j_K} \int_0^{\mathcal{T}} \boldsymbol{c}_{j_1 j_2 \cdots j_K}(t)\, \boldsymbol{b}_q^T(t)\, dt, \quad q = J+1, J+2, \ldots, Q.\end{aligned}$$

For the purpose of this example $K = 3$, meaning an interaction of three profile factors is considered. The number of runs is $n = 4$, and the number of basis functions for the three profile factors are $n_{x,1} = 4, n_{x,2} = 3, n_{x,3} = 2$. The coefficient matrices are such that:

$$\boldsymbol{\Gamma}_1 = \begin{pmatrix} -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \end{pmatrix}, \qquad \boldsymbol{\Gamma}_2 = \begin{pmatrix} -1 & 1 & -1 \\ 1 & 1 & -1 \\ 1 & 1 & -1 \\ 1 & 1 & -1 \end{pmatrix}, \qquad \boldsymbol{\Gamma}_3 = \begin{pmatrix} 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

The basis functions for the profile factors are such that:

$$\boldsymbol{c}_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \qquad \boldsymbol{c}_2 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \qquad \boldsymbol{c}_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The Hadamard product of the coefficient matrices times the basis functions is given by:

$$\boldsymbol{\Gamma}_1 \boldsymbol{c}_1 \circ \boldsymbol{\Gamma}_2 \boldsymbol{c}_2 \circ \boldsymbol{\Gamma}_3 \boldsymbol{c}_3 = \begin{pmatrix} 0 \\ -2 \\ 2 \\ -2 \end{pmatrix}.$$

The matrix $\mathbf{\Gamma}_{123}$ becomes:

$$\mathbf{\Gamma}_{123} = \begin{pmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \end{pmatrix},$$

and the vector $\boldsymbol{c}_{123}$ becomes:

$$\boldsymbol{c}_{123} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T.$$

Multiplying the two gives the matrix:

$$\mathbf{\Gamma}_{123}\ \boldsymbol{c}_{123} = \begin{pmatrix} 0 \\ -2 \\ 2 \\ -2 \end{pmatrix},$$

that match exactly the result of Hadamard products.

Appendix D

D-optimal designs for linear parameters and knot at $t = 0.5$

The aim of this section is to discuss some results from the FLM example with a single profile factor, discussed in Section 6.1. Specifically, for linear parameters, the objective values of the D-optimal designs does not change as the number of basis functions for the profile factor increases. The only exception is for $n_{x,1} = 3$. This is because for three basis functions, the knot vector of equally spaced knots does not include 0.5.

The model in Section 6.1 includes the intercept and a single profile factor, i.e., $Q = 2$. Thus, the model matrix $\boldsymbol{Z}$ is a $n \times \sum_{q=1}^{2} n_{\beta,q}$ matrix, that contains a column of 1's for the intercept, and $n_{\beta,2}$ columns resulting from the product of the design matrix and the solution to the integral of the product of basis functions,

$$\boldsymbol{Z}_{\bullet 2} = \boldsymbol{\Gamma}_1 \int_0^1 \boldsymbol{c}_1(t)\, \boldsymbol{b}_2^T(t)\, dt, \tag{D.1}$$

which for linear parameters is a $n \times 2$ matrix, i.e., $n_{\beta,2} = 2$. For the choice of BS basis of degree zero for the profile factor and linear power series for the functional parameter, the result is,

$$\boldsymbol{Z}_{\bullet 2} = \begin{pmatrix} \frac{1}{n_{x,1}} \sum_{l=1}^{n_{x,1}} \Gamma_{1l} & \frac{1}{2n_{x,1}^2} \sum_{l=1}^{n_{x,1}} \Gamma_{1l}(2l-1) \\ \frac{1}{n_{x,1}} \sum_{l=1}^{n_{x,1}} \Gamma_{2l} & \frac{1}{2n_{x,1}^2} \sum_{l=1}^{n_{x,1}} \Gamma_{2l}(2l-1) \\ \vdots & \vdots \\ \frac{1}{n_{x,1}} \sum_{l=1}^{n_{x,1}} \Gamma_{nl} & \frac{1}{2n_{x,1}^2} \sum_{l=1}^{n_{x,1}} \Gamma_{nl}(2l-1) \end{pmatrix}, \tag{D.2}$$

thus, the model matrix $\boldsymbol{Z}$ is given by,

$$\boldsymbol{Z} = \begin{pmatrix} 1 & \frac{1}{n_{x,1}}\sum_{l=1}^{n_{x,1}}\Gamma_{1l} & \frac{1}{2n_{x,1}^2}\sum_{l=1}^{n_{x,1}}\Gamma_{1l}(2l-1) \\ 1 & \frac{1}{n_{x,1}}\sum_{l=1}^{n_{x,1}}\Gamma_{2l} & \frac{1}{2n_{x,1}^2}\sum_{l=1}^{n_{x,1}}\Gamma_{2l}(2l-1) \\ \vdots & \vdots & \vdots \\ 1 & \frac{1}{n_{x,1}}\sum_{l=1}^{n_{x,1}}\Gamma_{nl} & \frac{1}{2n_{x,1}^2}\sum_{l=1}^{n_{x,1}}\Gamma_{nl}(2l-1) \end{pmatrix} = \begin{pmatrix} \mathbf{1}_n & \boldsymbol{Z}_{\cdot 2} \end{pmatrix}, \tag{D.3}$$

with $\mathbf{1}_n$ a $n \times 1$ vector of ones. The information matrix is defined as $\boldsymbol{Z}^T\boldsymbol{Z}$ and it is given by,

$$\boldsymbol{Z}^T\boldsymbol{Z} = \begin{pmatrix} \mathbf{1}_n^T\mathbf{1}_n & \mathbf{1}_n^T\boldsymbol{Z}_{\cdot 2} \\ \boldsymbol{Z}_{\cdot 2}^T\mathbf{1}_n & \boldsymbol{Z}_{\cdot 2}^T\boldsymbol{Z}_{\cdot 2} \end{pmatrix} = \begin{pmatrix} n & \mathbf{0}_2^T \\ \mathbf{0}_2 & \boldsymbol{Z}_{\cdot 2}^T\boldsymbol{Z}_{\cdot 2} \end{pmatrix}, \tag{D.4}$$

with $\mathbf{0}_2$ a 2×1 vector of zeros. In the second equality, $\mathbf{1}_n^T\boldsymbol{Z}_{\cdot 2} = [\boldsymbol{Z}_{\cdot 2}^T\mathbf{1}_n]^T = \mathbf{0}_2$, because the sum of every column of the D-optimal designs identified is 0, i.e.,

$$\sum_{i=1}^{n}\boldsymbol{\Gamma}_{1_{il}} = 0, \quad l = 1, 2, \ldots, n_{x,1}, \tag{D.5}$$

with $\boldsymbol{\Gamma}_{1_{il}}$ the il^{th} entry of the design matrix. As a consequence of (D.5), the sum of all entries of the design matrix is zero,

$$\sum_{i=1}^{n}\sum_{l=1}^{n_{x,1}}\boldsymbol{\Gamma}_{1_{il}} = 0. \tag{D.6}$$

Remember that the objective value of D-optimality depends on the determinant of the information matrix. Following the result from (D.4) and properties of the determinant,

$$|\boldsymbol{Z}^T\boldsymbol{Z}| = n|\boldsymbol{Z}_{\cdot 2}^T\boldsymbol{Z}_{\cdot 2}|, \tag{D.7}$$

with $\boldsymbol{Z}_{\cdot 2}^T\boldsymbol{Z}_{\cdot 2}$ a 2×2 matrix.

The upper diagonal entry of $\boldsymbol{Z}_{\cdot 2}^T\boldsymbol{Z}_{\cdot 2}$, denoted as $\boldsymbol{Z}_{\cdot 2}^T\boldsymbol{Z}_{\cdot 2_{[1,1]}}$ is given by,

$$\begin{aligned} \boldsymbol{Z}_{\cdot 2}^T\boldsymbol{Z}_{\cdot 2_{[1,1]}} &= \frac{1}{n_{x,1}^2}\left[\sum_{i=1}^{n}\left(\sum_{l=1}^{n_{x,1}}\boldsymbol{\Gamma}_{1_{il}}\right)^2\right] \\ &= \frac{1}{n_{x,1}^2}\left[\frac{n \times n_{x,1}^2}{2}\right] \\ &= \frac{n}{2}, \end{aligned} \tag{D.8}$$

using that all the entries of the design matrix are at the boundaries, with half of the runs constant at the boundaries, and half runs changing once at $t = 0.5$. The off diagonal entries of $\boldsymbol{Z}_{\cdot 2}^T\boldsymbol{Z}_{\cdot 2}$, denoted as $\boldsymbol{Z}_{\cdot 2}^T\boldsymbol{Z}_{\cdot 2_{[1,2]}}$ and $\boldsymbol{Z}_{\cdot 2}^T\boldsymbol{Z}_{\cdot 2_{[2,1]}}$ are identical and they are given by,

$$\begin{aligned}
\boldsymbol{Z}_{\cdot 2}^T\boldsymbol{Z}_{\cdot 2_{[1,2]}} &= \boldsymbol{Z}_{\cdot 2}^T\boldsymbol{Z}_{\cdot 2_{[2,1]}} \\
&= \frac{1}{2n_{x,1}^3}\left[\sum_{i=1}^{n}\sum_{l=1}^{n_{x,1}}\boldsymbol{\Gamma}_{1_{il}}\sum_{k=1}^{n_{x,1}}\boldsymbol{\Gamma}_{1_{ik}}(2k-1)\right] \\
&= \frac{1}{2n_{x,1}^3}\left[\frac{n \times n_{x,1} \times n_{x,1}^2}{4}\right] \\
&= \frac{1}{2n_{x,1}^3}\left[\frac{n \times n_{x,1}^3}{2}\right] \\
&= \frac{n}{4},
\end{aligned} \tag{D.9}$$

using that half runs are changing once, i.e., $n/2$ runs the sum of the entries is zero,

$$\sum_{l=1}^{n_{x,1}}\boldsymbol{\Gamma}_{1_{il}} = 0, \quad i = 1, 2, \dots, n/2, \tag{D.10}$$

and the other half runs are constant, i.e., $n/2$ runs the sum of the entries is $\pm n_{x,1}$,

$$\sum_{l=1}^{n_{x,1}}\boldsymbol{\Gamma}_{1_{il}} = \pm n_{x,1}, \quad i = n/2, n/2+1, \dots, n. \tag{D.11}$$

The lower diagonal entry of $\boldsymbol{Z}_{\cdot 2}^T\boldsymbol{Z}_{\cdot 2}$, denoted as $\boldsymbol{Z}_{\cdot 2}^T\boldsymbol{Z}_{\cdot 2_{[2,2]}}$ is given by,

$$\begin{aligned}
\boldsymbol{Z}_{\cdot 2}^T\boldsymbol{Z}_{\cdot 2_{[2,2]}} &= \frac{1}{4n_{x,1}^4}\left[\sum_{i=1}^{n}\left(\sum_{l=1}^{n_{x,1}}\boldsymbol{\Gamma}_{1_{il}}(2l-1)\right)^2\right] \\
&= \frac{1}{4n_{x,1}^4}\left[\frac{n \times n_{x,1} \times n_{x,1}^4}{2} + \frac{n}{4}\left(\sum_{l=1}^{n_{x,1}/2} -(2l-1)\right)^2 + \frac{n}{4}\left(\sum_{l=n_{x,1}/2+1}^{n_{x,1}}(2l-1)\right)^2\right] \\
&= \frac{1}{4n_{x,1}^4}\left[\frac{n \times n_{x,1}^4}{4} + \frac{n \times n_{x,1}^4}{16} + \frac{n \times n_{x,1}^4}{16}\right] \\
&= \frac{n}{8} + \frac{n}{64} + \frac{n}{64} \\
&= \frac{5n}{32}.
\end{aligned} \tag{D.12}$$

Thus, the information matrix is updated to,

$$\boldsymbol{Z}^T\boldsymbol{Z} = \begin{pmatrix} n & 0 & 0 \\ 0 & n/2 & n/4 \\ 0 & n/4 & 5n/32 \end{pmatrix}, \tag{D.13}$$

which does not depend on the number of basis functions $n_{x,1}$, giving the proof for the non-changing D-optimality objective value for linear basis and knot at $t = 0.5$. The determinant of the information matrix is calculated using (D.7),

$$|\boldsymbol{Z}^T\boldsymbol{Z}| = n\left(\frac{5n^2}{64} - \frac{n^2}{16}\right) = \frac{n^3}{64}. \tag{D.14}$$

Finally, when doubling and tripling the number of runs, the D-optimality objective values are becoming a half and third, respectively. The reason is that the objective function is adjusted on the total number of basis functions; see (5.30).

Appendix E

Documentation of the functions in fdesigns

E.1 Arguments of the function `pflm()`

Argument	Description
`formula*`	Object of type formula, to create the model equation. Elements need to match the list names for `startd`. Main effects are called using the names of the profile factors in `startd`, interactions are called using the names of the profile factors in `startd` separated with :, and polynomial effects are called using the function `P()`. Scalar factors are called using the same way and degree and knots through the arguments `dx` and `knotsx` are used to specify the scalar factors. A scalar factor is equivalent to a profile factor with degree 0 and no interior knots.
`nsd`	The number of starting designs. The default entry is 1.
`mc.cores`	The number of cores to use. The option is initialized from environment variable if set. Must be at least one, and for parallel computing at least two cores are required. The default entry is 1.
`npf*`	The total number of (profile) factors in the model.
`tbounds*`	A time vector of length 2, representing the boundaries of time, i.e., 0 and T.
`nruns*`	The number of runs of the experiment.

`startd`	Representing the starting design but if NULL then random designs are automatically generated. It should be a list of length `nsd`, and each component should be a list of length `npf`.
`dx*`	A vector of length `npf`, representing the degree of B-spline basis functions for the functions of the functional factors. A scalar factor must have a zero degree entry.
`knotsx*`	A list of length `npf`, with every object in the list representing the knot vectors of each functional factor. A Scalar factor must have no interior knots, i.e., an empty knot vector.
`pars*`	A vector of length equal to the total terms in formula, representing the basis choice for the (functional) parameters. Entries should be "power" or "bspline". A scalar parameter is represented through a "power" basis.
`db*`	A vector of length equal to the total terms in formula, representing the degree of the basis for the (functional) parameters. For power series basis the degree is: 1 for linear, 2 for quadratic, etc. A scalar parameter must have degree 0.
`knotsb`	A list of length equal to the total terms in formula, representing the knot vector of each (functional) parameter. For parameters represented by a power series basis, the knot vector should be empty or NULL.
`lambda`	Smoothing parameter to penalise the complexity of the functions of the profile factors. The default value is 0, i.e., no penalty.
`criterion*`	The choice of objective function. Currently there are two available choices: A-optimality (`criterion` = "A") and D-optimality (`criterion` = "D").
`tol`	The tolerance value in the optimisation algorithm. Default value is 0.0001.
`dlbound`	The design's lower bound. The default lower bound is -1.
`dubound`	The design's upper bound. The default upper bound is 1.
`progress`	If TRUE, it returns the progress of iterations from the optimisation process. The default entry is FALSE.

TABLE E.1: The arguments of the function `pflm()` with a description for each. The asterisk * indicates a mandatory argument.

E.2 Output of the function `pflm()`

Output	Description
`objval`	The objective value of the final design found from `pflm()`.
`design`	The final design found by `pflm()`. The final design is a list of length equal to the number of profile factors, exactly as the starting design `startd`.
`nits`	The total number of iterations needed to identify the final design.
`time`	The computational elapsed time in finding the final design.
`startd`	If starting designs were passed as an argument in `pflm()`, then this is the starting design from the argument `startd` that led to the final design. If no starting designs were passed to `pflm()`, this is the starting design generated randomly by `pflm()` that led to the final design.
`tbounds`	The argument `tbounds` .
`npf`	The argument `npf`.
`criterion`	The argument `criterion`.
`nruns`	The argument `nruns`.
`formula`	The argument `formula`.
`dx`	The argument `dx`.
`knotsx`	The argument `knotsx`.
`lambda`	The argument `lambda`.
`dbounds`	A vector of length 2, containing the arguments `dlbound` and `dubound`.
`bestrep`	A scalar value indicating the repetition that led to the final design.
`allobjvals`	A vector of length equal to `nsd`, representing the objective value from all of the repetitions.
`alldesigns`	A list of length equal to `nsd` of all the final designs. Each component of the list is a list of length equal to `npf` representing the final design in each repetition of the coordinate exchange algorithm.
`allstartd`	If starting designs were passed as an argument in `pflm()`, then this is the argument. If no starting designs were passed to `pflm()`, this is the starting designs generated randomly by `pflm()`.

TABLE E.2: The output of the function `pflm()` with a description for each.

E.3 Additional arguments of the function `pfglm()`

Argument	**Description**
`family*`	Specifies the error distribution and the link function of the functional generalised linear model. It can be the name of a family in the form of a character string, or an `R` `family` function; see the `R` function `family()` for details. Currently, the methodology is implemented only for the binomial family with the logit link, i.e., `family` = binomial(link = "logit"), and the Poisson family with the log link, i.e., `family` = poisson(link = "log").
`method*`	A character argument specifying the method of approximation of the expectation of the objective function with respect to a prior distribution of the parameters. Currently there are two available choices: 1. Deterministic quadrature approximation (`method` = "quadrature"); and 2. Stochastic Monte Carlo approximation (`method` = "MC").
`level`	An optional argument that specifies the accuracy level in the quadrature approximation. It is the number of points in each dimension. If NULL and `method` = "quadrature", then it defaults to 5. A high value of level may increase the computation time; especially for complicated models. If the model is complicated, i.e., several profile factors or interactions and polynomials, prefer to use `method` = "MC".
`B`	An optional argument that specifies the size of the Monte Carlo samples. If NULL and `method` = "MC", then it defaults to 10000. For `method` = "quadrature", B is computer automatically according to the dimensionality of the functional model and the `level` argument.

`prior*`	An argument to specify the prior distribution. For `method` = "MC", it should be a function of two arguments `B` and `Q`. Both arguments are integers. The value of `B` corresponds to the argument `B`, and the value of `Q` represents the total number of basis functions of the functional parameters. The function must generate a matrix of dimensions `B` by `Q`, that contains a random sample from the prior distribution of the parameters. For `method` = "quadrature", normal and uniform prior distribution for the parameters are allowed. For a normal prior distribution, the argument prior needs to be a list of length 2, with the entries named "mu" for the prior mean and "sigma2" for the prior variance-covariance matrix. The prior mean can be a scalar value that means all parameters have the same prior mean, or a vector of prior means with length equal to the number of parameters in the functional model. The prior variance-covariance can be a scalar value that means all parameters have a common variance, or a vector of prior variances with length equal to the number of parameters in the functional model, or a square matrix with the number of rows and columns equal to the number of parameters in the functional model. For a uniform prior distribution, the argument prior needs to be a list of a single entry named "unifbound" for the lower and upper bounds of the prior distribution. The bounds can be a vector of length 2 that means all parameters have the same bounds, or a matrix with the number of rows equal to 2 and the number of columns equal to the number of parameters in the functional model.

TABLE E.3: The additional arguments of the function `pfglm()`, that are not used in the function `pflm()`, with a description for each. The asterisk * indicates a mandatory argument.

E.4 Additional output of the function `pfglm()`

Output	Description
`family`	A vector of length equal to 2, containing the `family` and the link function.
`method`	The argument `method`.
`B`	The argument `B`.
`prior`	The argument `prior`.

TABLE E.4: The additional output of the function `pfglm()`, that is not included in the output of the function `pflm()`, with a description for each.

E.5 Arguments of the function `P()`

Argument	Description
`x*`	A coefficient matrix from the basis expansion of a profile factor. The name passed needs to match the name of the profile factor from the argument `startd` in `pflm()` and `pfglm()`.
`deg*`	The degree of the polynomial effect for the profile factor.

TABLE E.5: The arguments of the function `P()` with a description for each. The asterisk * indicates a mandatory argument.

Bibliography

Abramowitz, M. and Stegun, I. A. (1964), *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Vol. 55, US Government printing office.

Agresti, A. (2018), *An Introduction to Categorical Data Analysis*, Hoboken, New Jersey: John Wiley & Sons.

Antony, J. (2014), *Design of Experiments for Engineers and Scientists*, Amsterdam: Elsevier.

Asprey, S. and Macchietto, S. (2002), 'Designing robust optimal dynamic experiments', *Journal of Process Control* **12**(4), 545–556.

Atkinson, A. C. and Woods, D. C. (2015), 'Designs for generalized linear models', *Handbook of design and analysis of experiments* **7**, 471–514.

Atkinson, A., Donev, A. and Tobias, R. (2007), *Optimum Experimental Designs, with SAS*, 2 edn, Oxford: Oxford University Press.

Bajpai, R. and Reuss, M. (1980), 'A mechanistic model for penicillin production', *Journal of Chemical Technology and Biotechnology* **30**(1), 332–344.

Balakrishnan, V. (2003), 'All about the dirac delta function (?)', *Resonance* **8**(8), 48–58.

Balsa-Canto, Eva, M. R.-F. and Banga, J. R. (2007), 'Optimal design of dynamic experiments for improved estimation of kinetic parameters of thermal degradation', *Journal of Food Engineering* **82**(2), 178–188.

Binder, H. and Sauerbrei, W. (2008), 'Increasing the usefulness of additive spline models by knot removal', *Computational Statistics & Data Analysis* **52**(12), 5305–5318.

Bock, H. G., Körkel, S. and Schlöder, J. P. (2013), 'Parameter estimation and optimum experimental design for differential equation models', *Model Based Parameter Estimation* pp. 1–30.

Bolstad, W. M. and Curran, J. M. (2016), *Introduction to Bayesian Statistics*, Hoboken, New Jersey: John Wiley & Sons.

Boukhezzar, B. and Siguerdidjane, H. (2005), Nonlinear control of variable speed wind turbines without wind speed measurement, *in* 'Proceedings of the 44th IEEE Conference on Decision and Control', IEEE, pp. 3456–3461.

Box, G. E. and Tiao, G. C. (1992), *Bayesian Inference in Statistical Analysis*, Hoboken, New Jersey: John Wiley & Sons.

Box, G. E. and Wilson, K. B. (1951), 'On the experimental attainment of optimum conditions', *Journal of the Royal Statistical Society: Series B (Methodological)* **13**(1), 1–45.

Braun, M. and Golubitsky, M. (1983), *Differential Equations and their Applications*, Vol. 1, New York: Springer.

Butterfield, K. R. (1976), 'The computation of all the derivatives of a B-spline basis', *IMA Journal of Applied Mathematics* **17**(1), 15–25.

Caflisch, R. E. (1998), 'Monte Carlo and quasi-Monte Carlo methods', *Acta numerica* **7**, 1–49.

Chaloner, K. and Verdinelli, I. (1995), 'Bayesian experimental design: A review', *Statistical Science* **10**(3), 273–304.

Cuervo, D. P., Goos, P. and Sörensen, K. (2016), 'Optimal design of large-scale screening experiments: a critical look at the coordinate-exchange algorithm', *Statistics and Computing* **26**(1-2), 15–28.

Dantzig, G. B. (2002), 'Linear programming', *Operations Research* **50**(1), 42–47.

De Boor, C. (1978), *A Practical Guide to Splines*, springer-verlag New York.

Dean, A., Voss, D. and Draguljić, D. (2017), *Design and Analysis of Experiments*, New York: Springer.

Dickinson, R. P. and Gelinas, R. J. (1976), 'Sensitivity analysis of ordinary differential equation systems—a direct method', *Journal of Computational Physics* **21**(2), 123–143.

Dobson, A. J. and Barnett, A. G. (2018), *An Introduction to Generalized Linear Models*, Chapman and Hall/CRC.

Dragalin, V., Fedorov, V. and Wu, Y. (2008), 'Adaptive designs for selecting drug combinations based on efficacy–toxicity response', *Journal of Statistical Planning and Inference* **138**(2), 352–373.

Eddelbuettel, D., François, R., Allaire, J., Ushey, K., Kou, Q., Russel, N., Chambers, J. and Bates, D. (2011), 'Rcpp: Seamless R and C++ integration', *Journal of Statistical Software* **40**(8), 1–18.

Eddelbuettel, D. and Sanderson, C. (2014), 'RcppArmadillo: Accelerating R with high-performance C++ linear algebra', *Computational Statistics & Data Analysis* **71**, 1054–1063.

Eilers, P. H. and Marx, B. D. (1996), 'Flexible smoothing with B-splines and penalties', *Statistical science* **11**(2), 89–121.

Eilers, P. H. and Marx, B. D. (2010), 'Splines, knots, and penalties', *Wiley Interdisciplinary Reviews: Computational Statistics* **2**(6), 637–653.

Fedorov, V. (2010), 'Optimal experimental design', *Wiley Interdisciplinary Reviews: Computational Statistics* **2**(5), 581–589.

Ferraty, F. and Vieu, P. (2006), *Nonparametric Functional Data Analysis: Theory and Practice*, Springer Science and Business Media.

Friedman, J., Hastie, T. and Tibshirani, R. (2001), *The Elements of Statistical Learning*, number 10, New York: Springer.

Fuchs, K., Scheipl, F. and Greven, S. (2015), 'Penalized scalar-on-functions regression with interaction term', *Computational Statistics & Data Analysis* **81**, 38–51.

Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A. and Rubin, D. B. (2013), *Bayesian Data Analysis*, Chapman and Hall/CRC.

Georgakis, C. (2013), 'Design of dynamic experiments: A data-driven methodology for the optimization of time-varying processes', *Industrial & Engineering Chemistry Research* **52**(35), 12369–12382.

Goos, P. and Jones, B. (2011), *Optimal Design of Experiments: A Case Study Approach*, Hoboken, New Jersey: John Wiley & Sons.

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

Grove, D. M., Woods, D. C. and Lewis, S. M. (2004), 'Multifactor B-spline mixed models in designed experiments for the engine mapping problem', *Journal of Quality Technology* **36**(4), 380–391.

Hale, N. and Townsend, A. (2013), 'Fast and accurate computation of Gauss–Legendre and Gauss–Jacobi quadrature nodes and weights', *SIAM Journal on Scientific Computing* **35**(2), A652–A674.

Hall, P. (2011), 'Principal component analysis for functional data: methodology, theory, and discussion', *The Oxford Handbook of Functional Data Analysis* pp. 210–234.

Hastie, T. and Mallows, C. (1993), 'A statistical view of some chemometrics regression tools: Discussion', *Technometrics* **35**(2), 140–143.

Heuberger, P. S., Van den Hof, P. M. and Wahlberg, B. (2005), *Modelling and Identification with Rational Orthogonal Basis Functions*, Springer.

Ince, E. L. (1956), *Ordinary Differential Equations*, Massachusetts: Courier Corporation.

Jones, B., Allen-Moyer, K. and Goos, P. (2021), 'A–optimal versus D–optimal design of screening experiments', *Journal of Quality Technology* **53**(4), 369–382.

Kadri, H., Duflos, E., Preux, P., Canu, S. and Davy, M. (2010), Nonlinear functional regression: a functional RKHS approach, *in* 'Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics', JMLR Workshop and Conference Proceedings, pp. 374–380.

Klebanov, N. and Georgakis, C. (2016), 'Dynamic response surface models: a data-driven approach for the analysis of time-varying process outputs', *Industrial & Engineering Chemistry Research* **55**(14), 4022–4034.

Kotz, S. and Nadarajah, S. (2004), *Multivariate t-distributions and their Applications*, Cambridge: Cambridge University Press.

Lapeyre, B. (2007), 'Introduction to Monte-Carlo Methods', *Lecture, Halmstad, Sweden* pp. 2–4.

Lether, F. G. (1978), 'On the construction of Gauss–Legendre quadrature rules', *Journal of Computational and Applied Mathematics* **4**(1), 47–52.

Li, B. and Marx, B. D. (2008), 'Sharpening P–spline signal regression', *Statistical Modelling* **8**(4), 367–383.

Liu, Q. and Pierce, D. A. (1994), 'A note on Gauss–Hermite quadrature', *Biometrika* **81**(3), 624–629.

Liu, X., Nassar, H. and PodgÓrski, K. (2019), 'Splinets–efficient orthonormalization of the B-splines', *arXiv preprint arXiv:1910.07341* .

Magnus, J. R. and Neudecker, H. (2019), *Matrix Differential Calculus with Applications in Statistics and Econometrics*, Hoboken, New Jersey: John Wiley & Sons.

Marx, B. D. and Eilers, P. H. (1999), 'Generalized linear regression on sampled signals and curves: a p-spline approach', *Technometrics* **41**(1), 1–13.

Mason, J., Rodriguez, G. and Seatzu, S. (1993), 'Orthogonal splines based on B-splines—with applications to least squares, smoothing and regularisation problems', *Numerical Algorithms* **5**(1), 25–40.

Mathai, A. M. and Provost, S. B. (1992), *Quadratic Forms in Random Variables: Theory and Applications*, Dekker.

McGree, J. M. and Eccleston, J. A. (2012), 'Robust designs for Poisson regression models', *Technometrics* **54**(1), 64–72.

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

Miller, R. K. and Michel, A. N. (2014), *Ordinary Differential Equations*, Academic press.

Montgomery, D. C. (2017), *Design and Analysis of Experiments*, Hoboken, New Jersey: John wiley & sons.

Morris, J. S. (2015), 'Functional regression', *Annual Review of Statistics and Its Application* **2**, 321–359.

Overstall, A. M. and Woods, D. C. (2017), 'Bayesian design of experiments using approximate coordinate exchange', *Technometrics* **59**(4), 458–470.

Overstall, A. M., Woods, D. C. and Parker, B. M. (2019), 'Bayesian optimal design for ordinary differential equation models with application in biological science', *Journal of the American Statistical Association* **115**(530), 583–598.

Pan, R. and Saleh, M. (2019), 'On designing experiments for a dynamic response modeled by regression splines', *Applied Stochastic Models in Business and Industry* **36**(2), 251–267.

Perperoglou, A., Sauerbrei, W., Abrahamowicz, M. and Schmid, M. (2019), 'A review of spline function procedures in R', *BMC medical research methodology* **19**(1), 1–16.

Putranto, A., Chen, X. D., Devahastin, S., Xiao, Z. and Webley, P. A. (2011), 'Application of the reaction engineering approach (REA) for modeling intermittent drying under time-varying humidity and temperature', *Chemical Engineering Science* **66**(10), 2149–2156.

Rameez, S., Mostafa, S. S., Miller, C. and Shukla, A. A. (2014), 'High-throughput miniaturized bioreactors for cell culture process development: Reproducibility, scalability, and control', *Biotechnology Progress* **30**(3), 718–727.

Ramsay, J. O. and Dalzell, C. (1991), 'Some tools for functional data analysis', *Journal of the Royal Statistical Society: Series B (Methodological)* **53**(3), 539–561.

Ramsay, J. O., Hooker, G., Campbell, D. and Cao, J. (2007), 'Parameter estimation for differential equations: a generalized smoothing approach', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **69**(5), 741–796.

Ramsay, J. O. and Silverman, B. W. (2005), *Functional Data Analysis*, 2 edn, Springer.

Renardy, M. and Rogers, R. C. (2006), *An Introduction to Partial Differential Equations*, New York: Springer.

Riascos, C. A. and Pinto, J. M. (2004), 'Optimal control of bioreactors: a simultaneous approach for complex systems', *Chemical Engineering Journal* **99**(1), 23–34.

Rippin, D. W. (1983), 'Simulation of single-and multiproduct batch chemical plants for optimal design and operation', *Computers & Chemical Engineering* **7**(3), 137–156.

Roche, A. (2015), 'Response surface methodologiy for functional data with application to nuclear safety'.
URL: *https://hal.archives-ouvertes.fr/hal-01144628v2*

Roche, A. (2018), 'Local optimization of black-box functions with high or infinite-dimensional inputs: application to nuclear safety', *Computational Statistics* **33**(1), 467–485.

Russell, K. G., Eccleston, J., Lewis, S. M. and Woods, D. C. (2009), 'Design considerations for small experiments and simple logistic regression', *Journal of Statistical Computation and Simulation* **79**(1), 81–91.

Salzer, H. E., Zucker, R. and Capuano, R. (1952), 'Hermite polynomials', *Journal of Research of the National Bureau of Standards* **48**(2), 111.

Seber, G. A. (2008), *A Matrix Handbook for Statisticians*, John Wiley & Sons.

Seber, G. and Wild, C. (1989), *Nonlinear Regression*, Wiley, New York.

Tai, M., Ly, A., Leung, I. and Nayar, G. (2015), 'Efficient high-throughput biological process characterization: Definitive screening design with the ambr250 bioreactor system', *Biotechnology Progress* **31**(5), 1388–1395.

Titterington, D. (1980), 'Aspects of optimal design in dynamic systems', *Technometrics* **22**(3), 287–299.

Trummer, E., Fauland, K., Seidinger, S., Schriebl, K., Lattenmayer, C., Kunert, R., Vorauer-Uhl, K., Weik, R., Borth, N., Katinger, H. et al. (2006), 'Process parameter shifting: Part i. effect of dot, ph, and temperature on the performance of epo-fc expressing cho cells cultivated in controlled batch bioreactors', *Biotechnology and Bioengineering* **94**(6), 1033–1044.

Uciński, D. and Bogacka, B. (2005), 'T-optimum designs for discrimination between two multiresponse dynamic models', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**(1), 3–18.

Uciński, D. and Bogacka, B. (2007), 'A constrained optimum experimental design problem for model discrimination with a continuously varying factor', *Journal of Statistical Planning and Inference* **137**(12), 4048–4065.

Usset, J., Staicu, A.-M. and Maity, A. (2016), 'Interaction models for functional regression', *Computational Statistics & Data Analysis* **94**, 317–329.

Van De Ven, P. and Woods, D. (2014), 'Optimal blocked minimum-support designs for non-linear models', *Journal of Statistical Planning and Inference* **144**, 152–159.

Vermeulen, A. H., Bartels, R. H. and Heppler, G. R. (1992), 'Integrating products of B-splines', *SIAM Journal on Scientific and Statistical Computing* **13**(4), 1025–1038.

Weiser, C. (2016), *mvQuad: Methods for Multivariate Quadrature.* (R package version 1.0-6).
URL: *http://CRAN.R-project.org/package=mvQuad*

Weiss, N. A. (2006), *A Course in Probability*, Addison-Wesley.

Wood, S. N. (2017*a*), *Generalized Additive Models: An Introduction with R*, CRC press.

Wood, S. N. (2017*b*), 'P-splines with derivative based penalties and tensor product smoothing of unevenly distributed data', *Statistics and Computing* **27**(4), 985–989.

Woods, D. C., Lewis, S. M., Eccleston, J. A. and Russell, K. (2006), 'Designs for generalized linear models with several variables and model uncertainty', *Technometrics* **48**(2), 284–292.

Woods, D. C., Overstall, A. M., Adamou, M. and Waite, T. W. (2017), 'Bayesian design of experiments for generalized linear models and dimensional analysis with industrial and scientific application', *Quality Engineering* **29**(1), 91–103.

Woods, D., Lewis, S. and Dewynne, J. (2003), 'Designing experiments for multi-variable b-spline models', *Sankhyā: The Indian Journal of Statistics* pp. 660–677.

Wu, C. J. and Hamada, M. S. (2011), *Experiments: Planning, Analysis, and Optimization*, Vol. 552, Hoboken, New Jersey: John Wiley & Sons.

Yao, F. and Müller, H.-G. (2010), 'Functional quadratic regression', *Biometrika* **97**(1), 49–64.

Yoon, S. K., Kim, S. H. and Lee, G. M. (2003), 'Effect of low culture temperature on specific productivity and transcription level of anti-4–1bb antibody in recombinant chinese hamster ovary cells', *Biotechnology Progress* **19**(4), 1383–1386.

Young, N. (1988), *An introduction to Hilbert spaces*, Cambridge University Press.