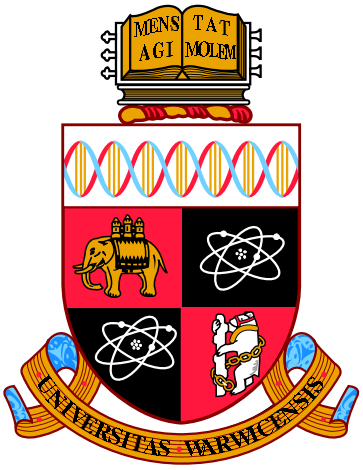

Input Estimation in Nonlinear Dynamical Systems for Drug-Discovery Applications

by

Magnus Trägårdh

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

School of Engineering

May 2017

'*"I checked it very thoroughly," said the computer, "and that quite definitely is the answer. I think the problem, to be quite honest with you, is that you've never actually known what the question is."*'

Douglas Adams, *The Hitchhiker's Guide to the Galaxy* (1979)

Contents

List of Tables **v**

List of Figures **viii**

Acknowledgments **xi**

Declarations **xii**

Summary **xv**

Abbreviations **xvi**

Chapter 1 Introduction **1**

1.1 Overview 1

1.2 Aims and objectives 3

1.3 Thesis Outline 5

Chapter 2 Theory **6**

2.1 Problem definition 6

2.2 Basics of statistical inference 8

2.2.1 Frequentist viewpoint 9

2.2.2 Bayesian viewpoint 11

2.3 Review of previous work 13

2.3.1 Introduction 13

2.3.2 Point-estimation methods 14

2.3.3 Bayesian methods 16

2.3.4 Other methods 18

2.4 A framework for input estimation 19

2.4.1 Choice of prior 21

2.4.2 Function parameterisation 24

2.4.3 Desired statistical quantity . . . 26
2.4.4 Choice of inference algorithm . . . 27
2.5 Inference algorithms for point estimates . . . 27
2.6 Algorithms for full Bayesian inference . . . 28
2.6.1 Markov chain Monte Carlo . . . 29
2.6.2 Sequential Monte Carlo . . . 30
2.6.3 Particle filters . . . 31
2.6.4 Variational Bayes and expectation propagation . . . 32
2.6.5 Extended and unscented Kalman filters . . . 33
2.7 Choice of inference algorithms and software . . . 33
2.7.1 Software for optimal control . . . 35
2.7.2 Software for Markov chain Monte Carlo . . . 37
2.8 Summary of the investigated methods . . . 37
2.9 Choice of case studies . . . 39

Chapter 3 In-depth theory **41**
3.1 The Karhunen-Loève expansion . . . 41
3.1.1 Karhunen-Loève expansion for penalisation of the first derivative 44
3.1.2 Karhunen-Loève expansion for penalisation of the second derivative . . . 46
3.2 Direct methods of optimal control . . . 48
3.2.1 Single shooting . . . 50
3.2.2 Multiple shooting . . . 51
3.2.3 Collocation . . . 52
3.3 Fundamentals of Markov chain Monte Carlo . . . 54
3.3.1 Properties of Markov chains . . . 57
3.3.2 The Metropolis-Hastings algorithm . . . 61
3.3.3 Practical considerations . . . 63
3.3.4 Random walk Metropolis-Hastings . . . 67
3.3.5 Gibbs sampling . . . 68
3.3.6 Metropolis-adjusted Langevin algorithm . . . 70
3.3.7 Riemannian manifold sampling-based methods . . . 71
3.3.8 Other methods . . . 74

Chapter 4 Case study — eflornithine absorption **78**
4.1 Background . . . 78
4.2 Model . . . 79
4.3 Data . . . 81

4.3.1 Real data . . . 81
4.3.2 Test data . . . 82
4.4 Materials and methods . . . 83
4.4.1 Optimal-control methods . . . 85
4.4.2 MCMC methods . . . 87
4.5 Results and discussion . . . 94
4.5.1 Optimal-control methods . . . 94
4.5.2 MCMC methods . . . 104
4.6 Summary . . . 112

Chapter 5 Case study — body-weight modelling 117
5.1 Background . . . 117
5.2 Models . . . 118
5.3 Data . . . 125
5.3.1 Real data . . . 125
5.3.2 Test data . . . 126
5.4 Materials and methods . . . 130
5.4.1 Optimal-control methods . . . 130
5.4.2 MCMC methods . . . 132
5.5 Results and discussion . . . 132
5.5.1 Optimal-control methods . . . 132
5.5.2 MCMC methods . . . 140
5.6 Summary . . . 147

Chapter 6 Case study — exenatide absorption 151
6.1 Background . . . 151
6.2 Model . . . 153
6.3 Data . . . 156
6.3.1 Real data . . . 156
6.3.2 Test data . . . 157
6.4 Materials and methods . . . 159
6.4.1 Optimal-control methods . . . 161
6.4.2 MCMC methods . . . 162
6.5 Results and discussion . . . 166
6.5.1 Optimal-control methods . . . 166
6.5.2 MCMC methods . . . 174
6.6 Summary . . . 182

Chapter 7 Conclusions and future work **184**
7.1 Summary of the performed work 184
7.2 Parametric and nonparametric methods 185
7.3 Comparison of input-estimation methods 186
7.4 Summary of results 188
7.5 Suggestions for future work 190

References **192**

Appendix A Data **213**
A.1 Eflornithine data 213
A.2 Mouse body-weight data 214
A.3 Human body-weight data 216
A.4 Exenatide data 219

List of Tables

4.1 Pharmacokinetic parameters of the eflornithine model 81
4.2 Parameters for the synthetic input function to the eflornithine model 82
4.3 Number of decision variables and constraints for the optimal-control problem formulations of the eflornithine model 88
4.4 A summary of datasets, methods and priors used to evaluate the optimal-control methods on eflornithine 88
4.5 A summary of datasets, methods and priors used to evaluate the MCMC methods on eflornithine 94
4.6 Accuracy measures of the optimal-control methods, evaluated on the test data 96
4.7 Median running times, number of iterations, and the proportion of successful estimation procedures, organised by optimisation method . 100
4.8 Median running times, number of iterations, and the proportion of successful estimation procedures, organised by choice of prior 100
4.9 Estimated bioavailability from the real data, using optimal-control techniques 103
4.10 Performance and robustness results for MCMC methods on the noisy eflornithine datasets 105
4.11 Accuracy measures of the MCMC methods on the noisy eflornithine test datasets 107
4.12 Performance results for the procedures using the L-curve approach to estimate the regularisation parameter τ 109
4.13 Mean estimated bioavailability for the real eflornithine datasets using MCMC 110

5.1 Model parameters for the mouse body-weight model 120
5.2 Physical activity parameters for all of the mouse body-weight time series 121
5.3 Model parameters for the human body-weight model 123

5.4 Baseline body weight and BMI for the human body-weight datasets . 126
5.5 Parameters for the synthetic input function for the mouse body-weight model 129
5.6 Parameters for the synthetic input function for the human body-weight model 130
5.7 Number of decision variables and constraints for the optimal-control problem formulations of the body-weight models 131
5.8 RMSE values for the synthetic mouse dataset 133
5.9 RMSE values for the synthetic human dataset 133
5.10 Median running times, number of iterations, and the proportion of successful estimation procedures on the mouse body-weight dataset, organised by optimisation method 137
5.11 Median running times, number of iterations, and the proportion of successful estimation procedures on the mouse body-weight dataset, organised by choice of prior 137
5.12 Median running times, number of iterations, and the proportion of successful estimation procedures on the human body-weight dataset, organised by optimisation method 137
5.13 Median running times, number of iterations, and the proportion of successful estimation procedures on the human body-weight dataset, organised by choice of prior 137
5.14 Accuracy measures for the MCMC methods on the sparse noisy mouse body-weight test datasets 143
5.15 Accuracy measures for the MCMC methods on the sparse noisy human body-weight test datasets 144
5.16 Performance and robustness results for MCMC methods on the noisy mouse body-weight datasets 145
5.17 Performance and robustness results for MCMC methods on the noisy human body-weight datasets 146

6.1 Pharmacokinetic parameters of the exenatide model 155
6.2 Synthetic input model parameters for the exenatide model 159
6.3 Number of decision variables and constraints for the optimal-control problem formulations of the exenatide model 162
6.4 Accuracy measures of the optimal-control methods for the short-term exenatide test data series 167

6.5 Accuracy measures of the optimal-control methods for the long-term exenatide test data series . . . 168
6.6 Median running times, number of iterations, and the proportion of successful estimation procedures for the short-term exenatide time series, organised by optimisation method . . . 168
6.7 Median running times, number of iterations, and the proportion of successful estimation procedures for the short-term exenatide time series, organised by choice of prior . . . 171
6.8 Median running times, number of iterations, and the proportion of successful estimation procedures for the long-term exenatide time series, organised by optimisation method . . . 171
6.9 Median running times, number of iterations, and the proportion of successful estimation procedures for the long-term exenatide time series, organised by choice of prior . . . 171
6.10 Accuracy measures of the MCMC methods on the short-term exenatide test datasets . . . 174
6.11 Accuracy measures of the MCMC methods on the long-term exenatide test datasets . . . 175
6.12 Performance and robustness results for MCMC methods on the noisy exenatide short-term datasets . . . 177
6.13 Performance and robustness results for MCMC methods on the noisy exenatide long-term datasets . . . 177

A.1 Eflornithine plasma-concentration data . . . 213
A.2 Body weight in g for the mice treated with R1c mAb opt1 . . . 214
A.3 Body weight in g for the mice treated with R1c mAb opt2 . . . 215
A.4 Human body-weight data, part 1 . . . 216
A.5 Human body-weight data, part 2 . . . 217
A.6 Human body-weight data, part 3 . . . 218
A.7 Plasma concentration in pM for the short-term exenatide data . . . 219
A.8 Plasma concentration in pM for the long-term exenatide data . . . 219

List of Figures

2.1 Fitting an input function to measurements from the system $\dot{x} = -x+u(t)$ 10
2.2 Value of the entropy-based log-prior 23
2.3 Example: Gaussian process $u(t)$ 24
2.4 A summary of the investigated priors, function parameterisations and inference algorithms. 38

3.1 A Markov chain targeting the distribution $\mathcal{N}(0, 1)$ 64
3.2 Samples drawn from a zero-mean Gaussian with standard deviation 10 68
3.3 Proposal distributions for correlated parameters 76
3.4 Analogy between the shortest path between points on the Earth and points in a statistical model 77

4.1 The nonlinear three-compartment eflornithine model from Johansson et al. (2013) 80
4.2 Plasma concentration measurements for the eflornithine data 81
4.3 Compartmental model corresponding to the synthetic Erlang input function 83
4.4 The synthetic test dataset generated for the eflornithine model . . . 84
4.5 Sample realisation of a scaled Wiener process 90
4.6 Energy content of the first few Karhunen-Loève basis functions for penalisation of the first derivative 91
4.7 Optimal-control input-estimation results for the four test datasets, using penalisation of the first derivative 97
4.8 Optimal-control estimation results for sparse test data with 10% noise 98
4.9 Example results from performing optimal-control input estimation on real data, using penalisation of the first derivative in the log domain 102
4.10 L-curves for the two real datasets, with a prior penalising the second derivative 102

4.11 Estimation results for the two real datasets, with the regularisation parameter chosen from the L-curve criterion 103
4.12 Comparison of results for successful and unsuccessful sampling . . . 106
4.13 An illustration of the effects of letting the prior over τ overpower the data 108
4.14 Kernel-density estimates of the bioavailability of the real eflornithine datasets, obtained from the MCMC analyses 111
4.15 Examples of estimation results for the sparse noisy test data for various priors using Karhunen-Loève basis functions 112
4.16 Estimates for the noisy test and real datasets, using penalisation of the second derivative in the log domain 113

5.1 Group means of measured energy intake and body weight for the mouse dataset 125
5.2 Group means of measured body weight for the human body-weight datasets 127
5.3 The synthetic input function for the human body-weight dataset . . 128
5.4 Optimal-control estimation results for the mouse body-weight model 135
5.5 Optimal-control estimation results for the human body-weight model 136
5.6 Examples of the estimated energy-intake profiles for the real mouse data, obtained using optimal-control methods with penalisation of the first derivative as the prior 138
5.7 Examples of the estimated energy-intake profiles for the real human data, obtained using optimal-control methods with penalisation of the first derivative as the prior 139
5.8 Sample trajectories for the sparse noisy human test data using B-splines and penalisation of the first derivative 141
5.9 Examples of MCMC estimation results for the sparse noisy test data for the mouse model for various priors using Karhunen-Loève basis functions 145
5.10 Examples of MCMC estimation results for the sparse noisy test data for the human model for various priors using Karhunen-Loève basis functions 146
5.11 MCMC estimation results for the real mouse data using Karhunen-Loève basis functions and a prior based on the first derivative 148
5.12 MCMC estimation results for the real human data using Karhunen-Loève basis functions and a prior based on the first derivative 150

6.1 Example of a plasma concentration profile after a 10 mg dose of Bydureon 154
6.2 The nonlinear three-compartment exenatide model from Gao and Jusko (2012) . . . 155
6.3 Comparison of the dynamical model and the steady-state approximation for short and long time scales . . . 156
6.4 Plasma concentration measurements for the exenatide dataset . . . 157
6.5 Compartmental model interpretation of the test input function . . . 158
6.6 The short time scale synthetic test dataset generated for the exenatide model . . . 160
6.7 The long time scale synthetic test dataset generated for the exenatide model . . . 160
6.8 The effect of regularisation when converting piecewise constant basis functions to B-spline coefficients . . . 163
6.9 Comparison of a Gaussian and a Student's t-distribution with 4 degrees of freedom . . . 164
6.10 Comparison of MCMC-based input estimation using a Gaussian likelihood and a Student's t likelihood with 4 degrees of freedom . . . 165
6.11 Comparison of optimal-control results for various priors using the sparse, noisy short-term test time series . . . 169
6.12 Comparison of optimal-control results for various priors using the sparse, noisy long-term test time series . . . 170
6.13 Example results from performing input estimation on the real short-term data, using an entropic prior . . . 172
6.14 Example results from performing input estimation on the real long-term data, using an entropic prior . . . 173
6.15 Examples of MCMC estimation results for the sparse noisy short-term test data for various priors using Karhunen-Loève basis functions . . . 178
6.16 Examples of MCMC estimation results for the sparse noisy long-term test data for various priors using Karhunen-Loève basis functions . . . 179
6.17 MCMC estimation results for the exenatide short-term data using Karhunen-Loève basis functions and a prior based on the second derivative in the log domain . . . 180
6.18 MCMC estimation results for the exenatide long-term data using Karhunen-Loève basis functions and a prior based on the second derivative in the log domain . . . 181
6.19 Kernel-density estimates for the bioavailability of the long-term release profiles . . . 181

Acknowledgments

Firstly, I would like express my sincere gratitude to my supervisors Dr Peter Gennemark, Prof. Michael Chappell, and Dr Neil Evans, for their patience, for always being encouraging and supportive, and for helping me to find ways to go forward at the times I came to what seemed like dead ends. Without their guidance, this thesis would not have been completed.

I also wish to thank the European Commission for funding my research through the Marie Curie FP7 People ITN European Industrial Doctorate (EID) project, IMPACT (Innovative Modelling for Pharmacological Advances through Collaborative Training), project number 316736.

Many people have contributed to making the PhD experience enjoyable. My fellow PhD students in the IMPACT project have been the best colleagues anyone could ask for. The biomedical superheroes at the School of Engineering, University of Warwick, made my stay in the UK a great experience. The people at AstraZeneca R&D in Gothenburg have provided a pleasant and friendly work environment. I have had many interesting discussions, not to mention really great lunches, with the people at the Fraunhofer-Chalmers Centre in Gothenburg.

Finally, I wish to thank my parents, and my sister and her family, for always supporting me.

Declarations

This thesis is submitted to the University of Warwick in support of my application for the degree of Doctor of Philosophy. It has been composed by myself and has not been submitted in any previous application for any degree.

Parts of this thesis have been published by the author:

Journal papers

1. Magnus Trägårdh, Michael J Chappell, Andrea Ahnmark, Daniel Lindén, Neil D Evans and Peter Gennemark. 'Input estimation for drug discovery using optimal control and Markov chain Monte Carlo approaches'. In: *Journal of Pharmacokinetics and Pharmacodynamics* 43.2 (2016), pp. 207–221, doi: 10.1007/s10928-016-9467-z

2. Magnus Trägårdh, Michael J Chappell, Johan E Palm, Neil D Evans, David LI Janzén and Peter Gennemark. 'Input Estimation for Extended-Release Formulations Exemplified with Exenatide'. In: *Frontiers in Bioengineering and Biotechnology* 5 (2017), p. 24, doi: 10.3389/fbioe.2017.00024

3. Peter Gennemark, Magnus Trägårdh, Daniel Lindén, Karolina Ploj, Anders Johansson, Andrew Turnbull, Björn Carlsson and Madeleine Antonsson. 'Translational Modeling to Guide Study Design and Dose Choice in Obesity Exemplified by AZD1979, a Melanin-concentrating Hormone Receptor 1 Antagonist'. In: *CPT: Pharmacometrics & Systems Pharmacology* 6.7 (2017), pp. 458–468, doi: 10.1002/psp4.12199

Posters and presentations

1. Poster: Magnus Trägårdh, Michael J Chappell, Neil D Evans, Mats Jirstrand, Johan Gabrielsson, Peter Gennemark. 'Input Estimation in Nonlinear Ordinary Differential Equations for Quantitative and Systems Pharmacology'. Presented at *PKUK*, Harrogate, UK, 1-2 November 2013 and at *the AstraZeneca Modelling & Simulation Symposium*, Alderley Park, UK, 3–4 December 2013.

2. Poster: Robert Andersson, David LI Janzén, Magnus Trägårdh, Michael J Chappell, Neil D Evans, Johan Gabrielsson, Peter Gennemark, Mats Jirstrand, Lambertus Peletier, James Yates. 'Dose-Response-Time Modelling, Input Estimation and Structural Identifiability as tools for Systems Pharmacology'. Presented at *the Noordwijkerhout Symposium on Pharmacokinetics, Pharmacodynamics and Systems Pharmacology*, Noordwijkerhout, the Netherlands, 23–25 April 2014 and at *the Chalmers Life Science Area of Advance Meeting*, Gothenburg, Sweden, 5 May 2014.

3. Poster: Magnus Trägårdh, Michael J Chappell, Neil D Evans, Mats Jirstrand, Johan Gabrielsson, Peter Gennemark. 'Input estimation in nonlinear dynamical systems for model-based drug discovery using optimal control techniques'. Presented at *PKUK*, Bath, UK, 4–6 November 2014 and at *the AstraZeneca Modelling & Simulation Symposium*, Mölndal, Sweden, 17–19 December 2014.

4. Poster: Magnus Trägårdh, Michael J Chappell, Neil D Evans, Johan Gabrielsson, Peter Gennemark. 'Input estimation for systems pharmacology'. Presented at *the Third School of Engineering Annual Postgraduate Symposium*, University of Warwick, UK, 9 March 2015.

5. Oral presentation: 'Input estimation in nonlinear dynamical systems for Quantitative and Systems Pharmacology'. Presented at *the Population Pharmacokinetics, Parameter Estimation and Model Validation Vacation School*, University of Warwick, UK, 21–25 September 2015.

6. Poster: Magnus Trägårdh, Michael J Chappell, Neil D Evans, Peter Gennemark.

'Input estimation for drug discovery using optimal control and Markov Chain Monte Carlo approaches'. Presented at *PKUK*, Chester, UK, 18–20 November 2015.

7. Oral presentation: 'Input estimation in nonlinear dynamical systems for Quantitative and Systems Pharmacology'. Presented at *the Fourth School of Engineering Postgraduate Symposium*, University of Warwick, UK, 22 April 2016.

8. Oral presentation: 'Input estimation in nonlinear dynamical systems for Quantitative and Systems Pharmacology'. Presented at *the QSP UK 1st Exchange Workshop*, University of Surrey, UK, 6 September 2016.

Summary

In mathematical modelling for drug discovery, nonparametric methods are an alternative to the more commonly used parametric methods, and have the advantage of requiring fewer modelling assumptions. This thesis considers nonparametric methods for performing *input estimation* (deconvolution) — inferring the input to a dynamical system based on measurements of the system's state. A typical application is to determine the absorption profile of an orally administered drug. Commonly used input-estimation methods are restricted to system models that are linear. This thesis aims to develop and evaluate methods which can be applied to nonlinear systems, and which are additionally able to provide uncertainty estimates. An input-estimation method is considered to be a particular choice of 1) prior, 2) function parameterisation, 3) desired statistical quantity, and 4) estimation algorithm. Two classes of methods have been selected and implemented: direct optimal-control methods and Markov chain Monte Carlo (MCMC) methods. These have been evaluated on two pharmacokinetic and two body-weight modelling applications, using simulated as well as real data. Evaluation was based on several criteria, including accuracy, computational speed, and usability. The results show that the methods can achieve good accuracy, provided that data are relatively densely sampled. Properly applied, optimal-control methods can achieve very high speed, approximately 0.1 s for typical problems, at the expense of not providing uncertainty estimates. For MCMC methods, the performance is highly dependent on the method settings as well as on the problem. In many cases, MCMC running times can be significantly reduced by a suitable choice of function parameterisation and sampling method. In all cases, estimation is based on clearly stated, quantifiable assumptions.

Abbreviations

AD	Automatic differentiation
BIC	Bayesian information criterion
BMI	Body-mass index
CLT	Central limit theorem
DIO	Diet-induced obese
EKF	Extended Kalman filter
ER	Extended release
ESS	Effective sample size
HAT	Human African trypanosomiasis
IV	Intravenous
ivivc	*In vitro-in vivo* correlation
LTI	Linear time-invariant
MALA	Metropolis-adjusted Langevin algorithm
MAP	Maximum a posteriori
MCMC	Markov chain Monte Carlo
ML	Maximum likelihood
MMALA	Manifold Metropolis-adjusted Langevin algorithm
ODC	Ornithine decarboxylase
ODE	Ordinary differential equation
PK	Pharmacokinetics
PLGA	Poly(lactic-co-glycolic acid)
RKHS	Reproducing kernel Hilbert space
RMSE	Root-mean-square error
RWMH	Random walk Metropolis-Hastings
SDE	Stochastic differential equation
SLLN	Strong law of large numbers
SMC	Sequential Monte Carlo

SMMALA	Simplified manifold Metropolis-adjusted Langevin algorithm
UKF	Unscented Kalman filter

Chapter 1

Introduction

1.1 Overview

Mathematical modelling is used extensively in drug discovery. Modelling aids in predicting compound properties such as pharmacokinetic (PK) parameters, efficacy and safety. Results from the analyses are useful for target validation and compound selection, and can provide relevant information for setting go/no go decision criteria. The models can be used to simulate hypothetical drug trials, which is useful for study design, including determining dosing and dosing schedules (Miller et al. 2005; Visser et al. 2013). Such modelling can hence reduce the need to perform expensive and time-consuming experimental studies. Additionally, it can make it possible that fewer animals than would otherwise be necessary are used in experimentation in order to draw statistically valid conclusions.

In a preclinical setting, relatively small, empirical models are commonly used. These models are typically expressed by systems of ordinary differential equations (ODEs), and are often linear compartmental models. The unknown model parameters, where possible, are estimated from available data (Gabrielsson and Weiner 2007). More elaborate, mechanistic models are often not feasible because of the relatively small amount or sparsity of the data available.

The system under study can also have inputs, which are time-varying external functions that affect, but are not affected by, the system. These can also be interpreted as time-varying model parameters. Typical examples include the absorption rate of an orally administered drug, and the energy intake in a body-weight model. In many cases, these input functions are represented by parametric models. As an example, drug absorption is often modelled by an absorption compartment, resulting in an exponential function. Similar models are available for energy intake (Göbel et al.

2014).

Parametric models are easy to use and interpret. By constraining the set of possible input functions to the set of functions that can be represented by a small number of parameters, overfitting can be avoided. However, the major disadvantage of these methods is that they rely on strong assumptions about the shape of the input function. An alternative approach is to use nonparametric methods. These make fewer assumptions, and allow the estimated function to take on any form, as long as it is supported by the data. In order to avoid overfitting, it is usually necessary to add additional criteria to the estimation problem, which serve to ensure that smooth solutions are preferred over unrealistic oscillatory solutions.

This thesis considers the problem where a known system model, described by a set of nonlinear ODEs, is driven by an unknown input. Measurements of the state of this system are obtained at a discrete set of time points. The aim is to recover the input function, using nonparametric methods, based on the model and measurements. In the literature, this estimation problem is often referred to as deconvolution. However, this terminology is only appropriate for linear time-invariant models, where the measured output of the model can be expressed as a convolution of the input function and the system's impulse response. As this thesis considers nonlinear systems, the more general term input estimation is used here in place of deconvolution. Note that it is still necessary to have a parametric model for the system. Constructing and analysing parametric models is outside the scope of this thesis.

A number of methods have been developed for performing input estimation (deconvolution) for pharmaceutical applications. These are, to a large extent, built on linear systems theory, and are therefore not suitable for problems where the system model is nonlinear (Verotta 1996). However, nonlinear models are widely used in drug discovery, for example Michaelis-Menten elimination (Johnson and Goody 2011), nonlinear receptor kinetics (Johansson et al. 2013), and nonlinear body-weight and body-composition models (Guo and Hall 2011; Hall et al. 2011). For this reason, this thesis aims to develop and evaluate input-estimation methods which are applicable to nonlinear systems. Additionally, in this work, much emphasis is placed on methods which are able to provide a measure of the uncertainty of the estimates. This is important, as the available data are often sparse. Hence, there may be a wide range of input functions which are all consistent with the data. A method that is able to report the full range of possible solutions provides more information than a method that merely provides a point estimate.

Problems with a similar mathematical structure have been extensively studied

in various branches of engineering and statistics (Rao 2009; Girolami and Calderhead 2011; Del Moral et al. 2006). A wide range of algorithms has been developed for this purpose, including optimal control-based methods, Markov chain Monte Carlo, sequential Monte Carlo, and nonlinear extensions of Kalman filters. However, the performance of these methods can be highly problem-dependent, making it necessary to evaluate them on models and data that are typical for the application at hand. One characteristic of data from drug-discovery experiments, which differs from many other engineering applications, is the sparsity and irregularity of measurement time points. The datasets that are analysed in this thesis are typical examples of such data, and are shown in Figs. 4.2, 5.1, 5.2 and 6.4.

For practical reasons, it may be necessary to discretise the input function so that it can be represented by a finite set of parameters. Hence, from a computational perspective, there may not be a clear distinction between parametric and nonparametric methods. For the purpose of this thesis, a method is considered to be nonparametric if the input function parameterisation is mainly intended as a computational convenience, and not intended to restrict what input functions can be represented.

It is important to recognise that nonparametric methods are not always appropriate. In many cases, the parameters in parametric models have physiological interpretations, which are themselves of primary interest. Parametric models can also be used for extrapolation, such as simulating the results of increasing the administered dose of a drug. However, when this is not required, nonparametric methods offer increased flexibility.

1.2 Aims and objectives

The aim of this thesis is to develop and evaluate methods for input estimation in drug-discovery applications which:

- are applicable to nonlinear systems,
- are able to provide uncertainty estimates, and
- satisfy the criteria outlined below as well as possible:

Accuracy. The method should be able to recover the input function with as small an error as possible. When real data are analysed, the true input function is usually not available. For this reason, accuracy is assessed using simulated data. The choice of error measure will be discussed in the application chapters.

Computational speed. While accuracy is arguably more important than speed, modellers in the pharmaceutical industry are working under time constraints.

Usability. The user of the method should not have to set any algorithm parameters, such as regularisation parameters. Those choices should be built into the method.

Statistical soundness. All assumptions made when estimating the input should be explicitly stated and well justified.

Robustness. The estimation method should be able to reliably recover the solution to the estimation problem, without computational issues. Examples of robustness failures are when an optimisation method fails to converge to an optimum, and when a Bayesian estimation method fails to characterise the posterior distribution.

Usefulness. The method should be applicable to as wide a range of input-estimation problems as possible.

While the ability to provide uncertainty estimates is desirable, methods which only provide point estimates are also investigated, as these may be able to achieve higher performance according to the criteria listed above.

To achieve these objectives, the engineering and statistics literature is reviewed in order to find suitable candidate methods. A few of these methods, that are deemed most promising for drug-discovery applications, are selected for further study. A framework has been established, where each selected method can be interpreted as a particular choice of:

- prior or regularisation criteria,
- function parameterisation,
- desired statistical quantity,
- estimation algorithm.

All combinations of such choices are tested on simulated as well as real data across a range of datasets. The methods are evaluated based on the criteria listed above.

1.3 Thesis Outline

Chapter 2 gives a formal mathematical definition of the input-estimation problem, as it is used in this thesis. A review of input-estimation methods that have previously been proposed for pharmaceutical applications is provided. These methods have mostly focused on linear systems. A set of general-purpose estimation methods which is widely used in the engineering and statistical literature is briefly reviewed. Finally, the framework for input-estimation methods is presented. This framework, which provides a taxonomy for classifying nonparametric input-estimation methods, is an original contribution of this thesis.

Chapter 3 provides a detailed, in-depth theory of key techniques. This is background material, not original in content, and not essential for understanding the application chapters of the thesis. This organisation was chosen to make the thesis more accessible for readers who are mainly interested in the applications.

Chapters 4–6 are application chapters and constitute the main original contribution of this thesis. Each chapter presents one application area with example models, together with both real and simulated data. A broad range of input-estimation methods is tested on the data. Estimation procedures were designed to be as similar as possible for all datasets, to allow comparisons between datasets and models.

Chapter 7 summarises the finding of Chapters 4–6. The applicability, strengths and limitations of the methods are discussed, and recommendations for practitioners are provided. Finally, suggestions for future development are given.

Chapter 2

Theory

2.1 Problem definition

This work considers dynamical systems that can be represented by a system of ODEs. Such systems can be described by:

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) \tag{2.1}$$

$$\mathbf{x}(t_i) = \mathbf{x}^{(0)} \tag{2.2}$$

where t is time, defined in some interval $[t_i, t_f]$, $\mathbf{x}(t) \in \mathbb{R}^{d_x}$ is the state vector, $\mathbf{u}(t) \in \mathbb{R}^{d_u}$ is an vector of external input functions, and $\mathbf{x}^{(0)}$ is the initial state. The number of state variables is denoted d_x, and the number of inputs is denoted d_u. The dynamics are fully specified by the right-hand side of Eq. (2.1), which is a map $\mathbf{f}\colon \mathbb{R} \times \mathbb{R}^{d_x} \times \mathbb{R}^{d_u} \to \mathbb{R}^{d_x}$. Hence, the state trajectory $\mathbf{x}(t)$ is the solution to an initial value problem.

The system is observed at a discrete set of time points, $t_1 \dots t_n$. Each measurement $\mathbf{y}(t) \in \mathbb{R}^{d_y}$ is represented by a measurement equation:

$$\mathbf{y}(t) = \mathbf{g}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)). \tag{2.3}$$

Here, $\mathbf{g}$ is a map $\mathbf{g}\colon \mathbb{R} \times \mathbb{R}^{d_x} \times \mathbb{R}^{d_u} \times \mathbb{R}^{d_v} \to \mathbb{R}^{d_y}$, $\mathbf{v}(t) \in \mathbb{R}^{d_v}$ is a random variable that accounts for measurement errors, d_y is the number of measured variables, and d_v is the dimension of the measurement noise.

It is assumed that $\mathbf{f}(\cdot)$, $\mathbf{g}(\cdot)$ and $\mathbf{x}^{(0)}$ are known. The function $\mathbf{f}(\cdot)$ is also assumed to satisfy technical requirements such as Lipschitz continuity, so that the system of ODEs has a unique solution. Furthermore, the distribution of $\mathbf{v}(t)$ is

assumed to be known. No further assumptions on $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ are made, although in practice, some estimation methods may have additional requirements such as differentiability. In particular, $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ can be nonlinear. The input-estimation problem will, for the purposes of this work, be formally defined as: *given a model specification, and a set of measurements* $\mathbf{y}(t_1) \dots \mathbf{y}(t_n)$*, infer the vector of input functions* $\mathbf{u}(t)$ *in the interval* $[t_i, t_f]$.

For input estimation to be possible, the system must be *structurally identifiable*. Informally, this means that, given continuous-time, error-free measurements $\mathbf{y}(t)$, there is a unique vector $\mathbf{u}(t)$ that predicts these measurements. In this situation it is possible to recover $\mathbf{u}(t)$ perfectly. Structural identifiability usually refers to estimation of static parameters, but the same concept carries over to time-varying functions.

In practice, $\mathbf{u}(t)$ cannot be perfectly recovered because the measurements are:

Sparse. Measurements are in practice commonly available only for a finite set of time points. In pharmaceutical applications, measurements are often expensive, and therefore there is a strong incentive to limit the number of samples.

Noisy. Because of the measurement error $\mathbf{v}(t)$, in a practical situation the true system state is never perfectly observed.

The task of determining unknown quantities such as parameters or functions from data is inherently a statistical problem. The two dominant schools of thought in statistics are the frequentist and Bayesian approaches (Bayarri and Berger 2004). In this work, the frequentist as well as the Bayesian view will be presented.

Previous literature on input estimation in pharmacokinetics has focused on determining the input to a linear time-invariant (LTI) system. As the name suggests, such a system has the properties:

Linearity. For any input functions $\mathbf{u}_1(t)$ and $\mathbf{u}_2(t)$ with corresponding output functions $\mathbf{y}_1(t)$ and $\mathbf{y}_2(t)$, and scalars c_1 and c_2, the following holds: the input $c_1 \cdot \mathbf{u}_1(t) + c_2 \cdot \mathbf{u}_2(t)$ will produce the output $c_1 \cdot \mathbf{y}_1(t) + c_2 \cdot \mathbf{y}_2(t)$. In other words, if the input function is scaled by a constant factor, the output will be scaled by the same factor. Furthermore, the output from a sum of two inputs will be the sum of the outputs from each individual input.

Time-invariance. For any input function $\mathbf{u}(t)$ with corresponding output $\mathbf{y}(t)$, the input function $\mathbf{u}(t - T)$ will result in the output $\mathbf{y}(t - T)$ for any scalar T. In other words, if the input function is shifted in time, the output function will be shifted by an equal amount.

For LTI systems, the relationship between a scalar input $u(t)$ and a scalar output $y(t)$ can be described by:

$$y(t) = \int_0^t I(t-\tau) \cdot u(\tau)\, \mathrm{d}\tau = I(t) * u(t). \tag{2.4}$$

$I(t)$ is the system's *impulse response*, that uniquely determines the system's response to any input function, and $*$ is the *convolution operator*. Because of this relationship, input estimation is often referred to as *deconvolution* in the literature. Since this work focuses on nonlinear models, the word deconvolution is avoided here.

For nonlinear systems, the input-output relationship cannot be expressed by a convolution operation. Instead, the relationship can be represented by its system of ODEs. These systems can be solved using numerical methods (Butcher 2008), although such methods are more computationally expensive than the analytical techniques that can be used for linear systems. However, an advantage of the ODE representation is that it is straightforward to extend the methods to systems with multiple inputs and multiple outputs. It is also straightforward to allow nonzero initial conditions, and in principle, joint estimation of model parameters, initial conditions and input functions can be done. However, this thesis focuses on the single-input single-output case with known model parameters. Another issue is that the system can become saturated in many commonly employed nonlinear models, such as in receptor kinetics. This will result in the system's states practically ceasing to respond to changes in the input. From a structural identifiability perspective, this is not an issue, as saturation is only achieved in the limit when time tends to infinity. Hence, for any experiment of finite duration, the input function can be recovered when measurements are perfectly noise-free, as is assumed in structural identifiability analysis. However, when data are sparse and noisy, it may no longer be practically possible to detect changes in the input when the system is in saturation. Therefore, input estimation is only likely to be successful when the system is not near saturation for an extended amount of time. Methods which are able to provide estimates of uncertainty can help here, as they will be able to report that there is a wide range of possible input functions which can explain the data.

2.2 Basics of statistical inference

Here, the basics of statistical inference are presented. To make the presentation as general as possible, the parameters to be estimated are denoted by $\boldsymbol{\theta}$. In the special case of input estimation, these parameters characterise the unknown input function(s)

$\mathbf{u}(t)$.

2.2.1 Frequentist viewpoint

A statistical inference problem can be formulated as follows: given an unknown parameter vector, $\boldsymbol{\theta}$, and a statistical model describing the distribution of data, $p(\mathbf{y}|\boldsymbol{\theta})$, find an estimate for $\boldsymbol{\theta}$ given a set of observations $\mathbf{y}_1 \dots \mathbf{y}_n$. In the frequentist paradigm, inference is performed by constructing an *estimator* for $\boldsymbol{\theta}$, which is a function that takes observed data as input, and gives an estimate $\hat{\boldsymbol{\theta}}$ as output (Ross 2014). The data are treated as random variables, while the true parameters $\boldsymbol{\theta}$ are considered to be fixed, but unknown. Since the estimate $\hat{\boldsymbol{\theta}}$ is a function of random variables, it is itself a random variable — if the same estimation procedure is performed multiple times on different datasets, all drawn from the same distribution $p(\mathbf{y}|\boldsymbol{\theta})$, the value of $\hat{\boldsymbol{\theta}}$ will vary. An estimator can be characterised by its *bias* and *variance* (Friedman et al. 2001):

Bias. The bias of an estimator is the difference between the true value of $\boldsymbol{\theta}$ and the expected value of $\hat{\boldsymbol{\theta}}$, i.e. $\text{Bias}\left[\hat{\boldsymbol{\theta}}\right] = \boldsymbol{\theta} - \mathbb{E}\left[\hat{\boldsymbol{\theta}}\right]$.

Variance. The variance of an estimator is defined as $\text{Var}\left[\hat{\boldsymbol{\theta}}\right] = \mathbb{E}\left[\left(\hat{\boldsymbol{\theta}} - \mathbb{E}\left[\hat{\boldsymbol{\theta}}\right]\right)^2\right]$.

A good estimator should have small variance, and zero (or at least small) bias. A common way to construct estimators is to use maximum likelihood (ML) estimation (Casella and Berger 2002). Here, the estimator is chosen to select the parameter values that maximises the probability of the data given the parameters, i.e.

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p\left(\mathbf{y}|\boldsymbol{\theta}\right). \tag{2.5}$$

The probability of the data given the parameters, viewed as a function of the parameters for fixed data, is termed the *likelihood* of the parameters, hence the name maximum likelihood. For practical purposes, it is more common to work with the log-likelihood. Since the logarithm is a monotonically increasing function, the maximum of the likelihood coincides with the maximum of the log-likelihood.

When fitting a statistical model, it is important to consider the possibility of *overfitting*. When only a small amount of data is available, a sufficiently complicated model with many parameters may be able to fit the data perfectly, even if it is, in fact, not an appropriate model. When the model is too complicated, there may be many, or indeed infinitely many, parameter settings that can fit the data equally

well. In the case of input estimation, the unknown quantities are not just parameters, but continuous functions of time. The parameters can be considered to be infinite-dimensional, since each such function is defined for every point between the endpoints t_i and t_f. This can be seen as an extreme case of the overfitting problem. As illustrated in Fig. 2.1, the ML estimate does not have a unique solution, in that multiple functions fit the data equally well. All of the displayed functions will have the same likelihood, suggesting that the likelihood alone is a poor indicator of the quality of the estimate.

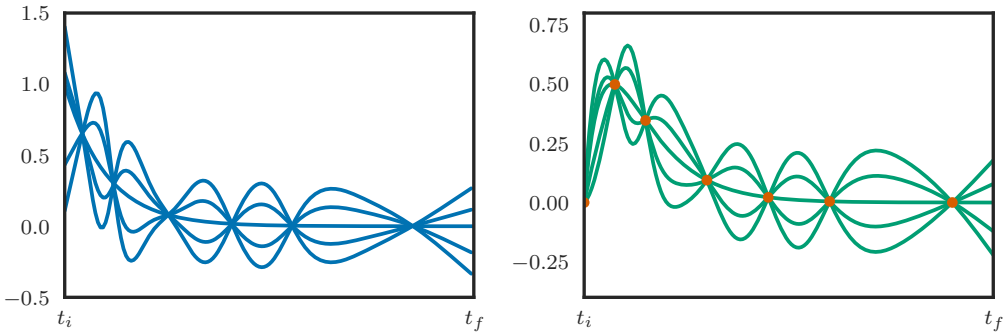

Figure 2.1: Fitting an input function to measurements from the system $\dot{x} = -x + u(t)$. Left: A number of candidate input functions $u(t)$. Right: The corresponding state trajectories $x(t)$ together with measured data (dots). All candidate input functions fit the data equally well. However, most functions oscillate more than the data seem to support. This shows that for these types of problems, a good data fit is not good enough.

Figure 2.1 suggests that while many functions can give a good data fit, most of them will be implausible since they exhibit large oscillations, which are not supported by the data. The ML estimate can be given a unique solution by adding a *regularisation term*, which penalises solutions that are overly oscillatory (Verotta 1996). Introducing the notation $E_D = -\log p(\mathbf{y}|\boldsymbol{\theta})$, the modified estimator becomes

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left(E_D + \tau \cdot E_R\right) \tag{2.6}$$

where E_R is the regularisation term. τ is a *regularisation parameter*, whose value determines the trade-off between data fit and regularity. As an example, when $\boldsymbol{\theta}$ is a parameterisation of an unknown continuous-time function $u(t, \boldsymbol{\theta})$, the regularisation can be based on the L^2 norm of the jth time derivative of the function. For a scalar function $u(t, \boldsymbol{\theta})$, this is given by

$$E_R = \frac{1}{2}\int_{t_i}^{t_f} \left(\frac{\mathrm{d}^j u(t,\boldsymbol{\theta})}{\mathrm{d}t^j}\right)^2 \mathrm{d}t. \tag{2.7}$$

For vector-valued inputs, each component can be assigned such a prior. Typically, j is 1 or 2. When $j = 1$, E_R reaches its minimum value of 0 when the function is constant, while any deviation from a constant function will result in a nonzero penalty. Similarly, when $j = 2$, any linear function will have a zero penalty. Another regularisation term that has been suggested is given by the negative entropy of a discretised version of the function (Hattersley et al. 2008). Since the smallest penalty is given to the function with the highest entropy, methods using this regulariser are often referred to as maximum entropy methods.

Various criteria have been suggested for setting a suitable value for the regularisation parameter τ. The *discrepancy criterion* suggests selecting τ so that the mean squared distance between predictions and measurements is equal to the expected squared distance under the model (Twomey 1965). In *ordinary and generalised cross-validation*, estimation is performed with some measurements left out, and the ability to predict the values of these left-out measurements is assessed for various values of τ (Golub et al. 1979). In the *L-curve approach*, estimation is performed for a large number of settings of τ. E_D and E_W are then plotted against each other. For small τ, the data fit will be almost perfect and the data cost will be almost zero. For large τ the curve is forced to be very smooth at the expense of the data fit, causing a high value of E_D. Between these extremes the curve often shows a characteristic corner, whose corresponding τ value is a reasonable trade-off between regularity and data fit (Hansen and O'Leary 1993).

2.2.2 Bayesian viewpoint

In frequentist statistics, parameters are treated as unknown but non-random. In contrast, Bayesian statistics formally treats all unknown quantities as stochastic variables, including latent variables, missing data, and parameters. A good and comprehensive introduction to the Bayesian interpretation of probability is given in Jaynes (2003). Unknown functions can be modelled as *stochastic processes*. A stochastic process (Jazwinski 2007) is a collection of random variables indexed by a variable. In the models considered here, the indexing variable is time, meaning that a single realisation of a stochastic process is a function of time.

The quantity of interest in Bayesian statistics is the *posterior distribution*, $p(\boldsymbol{\theta}|\mathbf{y})$, given by Bayes' rule (MacKay 2003) as

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{y})}. \tag{2.8}$$

The factor $p(\boldsymbol{\theta})$ is the *prior distribution*, which is a probability distribution describing the knowledge about $\boldsymbol{\theta}$ *before* any data have been observed. The term in the denominator can be computed by marginalising out $\boldsymbol{\theta}$ from the joint probability of parameters and data:

$$p(\mathbf{y}) = \int p(\mathbf{y}, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} = \int p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}. \tag{2.9}$$

This term is called the *marginal likelihood*, since it describes the probability distribution of the data when the parameters have been marginalised out (Barber 2012). Note that for a given dataset, this is a constant value. The integral in Eq. (2.9) is in practice often difficult to compute, because it involves integration over a potentially very high-dimensional space (Friel and Pettitt 2008). For parameter estimation, it is often not necessary to compute the marginal likelihood. However, when performing model selection, the marginal likelihood provides valuable information about the competing models' ability to predict the data (MacKay 1992).

In contrast to a maximum likelihood estimate, the result from Bayesian estimation is a distribution over the parameters. For practical reasons, it is desirable to summarise the estimation as a single number. Common summaries are the posterior mean (Gelman et al. 2013):

$$\hat{\boldsymbol{\theta}}_{mean} = \mathbb{E}\left[p(\boldsymbol{\theta}|\mathbf{y})\right], \tag{2.10}$$

and the maximum a posteriori (MAP) estimate (MacKay 2003):

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{y}). \tag{2.11}$$

The relationship between MAP estimation and regularisation can easily be seen by writing Bayes' rule in logarithmic form:

$$\log p(\boldsymbol{\theta}|\mathbf{y}) = \log p(\mathbf{y}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \log p(\mathbf{y}). \tag{2.12}$$

The first term is the log-likelihood. The last term is a constant and can be ignored for the purposes of maximisation. By identifying $\log p(\boldsymbol{\theta}) = -\tau E_R$, it can be seen that frequentist regularised ML estimation can be interpreted as Bayesian MAP estimation under a certain prior. τ here controls the "tightness" of the prior — a large τ means that large oscillations are improbable. In the Bayesian framework, τ can be

incorporated into the parameter vector and estimated like any other parameter. In this case, an augmented parameter vector $\overline{\boldsymbol{\theta}} = \begin{bmatrix} \boldsymbol{\theta}^T, & \tau \end{bmatrix}^T$ is introduced, and a joint prior over these parameters can be assigned, with a log-posterior given by

$$p(\boldsymbol{\theta}, \tau|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta}, \tau)p(\boldsymbol{\theta}, \tau)}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\tau)p(\tau)}{p(\mathbf{y})} \tag{2.13}$$

where the second equality relies on the assumption that $\mathbf{y}$ is conditionally independent of τ given $\boldsymbol{\theta}$. This means that for a given realisation of the input function, the distribution of the measurements is not affected by the regularisation parameter.

Bayesian inference is challenging from a computational point of view. Computing the expectation of a parameter with respect to the posterior involves integrating over all parameters. In addition, evaluating the posterior itself requires integrating over all parameters to obtain the marginal likelihood. For some models, it is possible to perform these integrations analytically. Traditional numerical integration methods, such as the trapezoidal rule and Simpson's rule (Flannery et al. 1992, Ch. 4) generally scale poorly with the number of dimensions, and are not suitable for high-dimensional problems (Robert and Casella 2004). Methods for Bayesian integration generally rely on either functional approximations or Monte Carlo methods that use sampling to approximate the integrals.

2.3 Review of previous work

2.3.1 Introduction

Several methods have been proposed in the literature on input estimation or deconvolution in pharmacokinetics and physiological modelling. Unless otherwise stated, the methods assume that the system dynamics are linear, and that the system has a single unknown input and a single measured variable. However, many concepts carry over to nonlinear systems, and systems with multiple inputs and outputs. Two common methods are the Wagner-Nelson (Wagner and Nelson 1964) and the Loo-Riegelman (Loo and Riegelman 1968) method. These methods do not make any assumptions about the form of the input, but they do assume that the system is represented by a linear one-compartment (Wagner-Nelson) or two-compartment (Loo-Riegelman) model, and they also assume that time-continuous measurements are available. They cannot be considered to be statistical methods. Rather, they are based on directly solving the system equations for the unknown input.

Parametric methods have also been suggested, that constrain the input function to have a prespecified functional form. Using sums of exponentials has been

discussed by Verotta (1996). Similar methods have been presented by Veng-Pedersen and Modi (1992). The latter method additionally assumes that the impulse response too is a sum of exponentials. Csajka et al. (2005) present a method where the input function is represented by a sum of inverse Gaussian functions, and use it within a mixed-effects model. These methods reduce the input-estimation problem to a parameter-estimation problem, and have more in common with general parameter-estimation procedures than with the nonparametric methods that are the focus of this thesis.

2.3.2 Point-estimation methods

Verotta (1996) provides a detailed review of deconvolution, with a focus on non-parametric methods. The most basic method, *quadrature*, assumes that the input function is piecewise constant between measurements. The relation between input and measurements can be expressed as a system of linear equations. Since the number of unknowns is equal to the number of equations, the system has a unique solution, provided that it is invertible. While very straightforward, this method does have obvious problems. An input function that is constant between measurements may not be a very realistic model. Additionally, by forcing exact agreement with the data, the method becomes sensitive to noise. This is especially true for systems with lowpass characteristics, which attenuate high frequencies. Any high-frequency noise will become amplified when such a system is inverted. A counterintuitive consequence of this is that a densely sampled dataset is more sensitive to noise than a more sparsely sampled dataset, since the dense dataset can represent higher frequencies. One way to overcome this is to model the input function as being piecewise constant on a set of intervals that are fewer than the number of measurements. This results in an overdetermined linear system, for which the least-squares solution can be obtained in closed form using the Moore-Penrose pseudoinverse (Penrose 1955). This helps to decrease noise sensitivity, at the expense of constraining the functional form of the input function even more. A more realistic-looking function can be obtained by using differentiable basis functions such as B-splines (Boor 1986). Regardless of the choice of basis functions, a closed-form solution is available.

An alternative to the quadrature method, suggested by Verotta (1996), is to use *penalised least-squares*, where the methods above are modified by adding a regularisation (penalisation) term, as described in Section 2.2.1. This makes it possible to have a more fine-grained input function parameterisation, and hence this method can be described as nonparametric. A finite difference approximation of the first or second derivative of the input function can be constructed, and its sum of squares can

be used as a regularisation term. Penalisation can be done either on the basis function coefficients or on the function itself. For piecewise constant functions, this makes little difference, but for B-splines, this amounts to making different assumptions. Since the finite difference approximation is a linear operator, the estimation problem is still a quadratic optimisation problem, which has a closed-form solution. It is also possible to add constraints to these problems, that disallow solutions that attain negative values. In addition to making the solution more physiologically plausible, it can also have a regularising effect, as it prevents the input function from oscillating between large positive and negative values. When constraints are added to the problem, it no longer has a closed-form solution. However, it is a convex optimisation problem, for which efficient methods with strong guarantees of finding the optimum exist (Boyd and Vandenberghe 2004).

Verotta (1996) also mentions that penalised least-squares has a Bayesian interpretation, as explained in Section 2.2.2. In this view, the regularisation term is interpreted as a quantity that is proportional to the negative log-prior. Adding a regularisation term, penalising a discrete-time approximation of the jth derivative, is equivalent to modelling the input function as a cascade of j integrators, driven by white noise referred to as the *process noise*. The continuous-time setting is conceptually similar, although such an input function would have to be modelled using tools from stochastic calculus (Klebaner 2012; Øksendal 2003) in order to be mathematically rigorous. Seeking least-squares solutions also means that there is an implicit assumption that the measurement noise has a Gaussian distribution. Non-negativity constraints can be interpreted as assigning a prior probability density of zero for any solution that takes negative values.

Similar ideas have been discussed by Sparacino and Cobelli (1996). The application under consideration is the task of estimating insulin secretion rate after a intravenous glucose tolerance test. The input function is modelled as a piecewise constant function, with a considerably larger number of intervals than the number of measurements. To make the estimation problem well-defined, a regularisation term penalising the first derivative is added, equivalent to treating the input function as a Gaussian random walk. By making a Bayesian interpretation, the optimal regularisation parameter can be shown to be the ratio of the measurement noise variance and the process noise variance. In this setting, it is possible to derive optimal settings for the regularisation parameter based on ideas presented by MacKay (1992), using the expected weighted residual sum of squares, and the expected weighted sum of squares of the function estimate. The process and measurement noise variance is then set so that the observed values are equal to the estimated ones. Sparacino and

Cobelli (1996) also mention the possibility of using splines, but consider this to be a parametric method.

De Nicolao et al. (1997) discuss similar nonparametric methods, and also discuss their similarities to discrete-time Kalman filtering. It is noted that while these methods have similar statistical justifications, a drawback with Kalman filtering is that it is not straightforward to add constraints. Furthermore, nonuniform sampling causes the discretised system to be time-varying.

2.3.3 Bayesian methods

Although the methods presented above often have a Bayesian interpretation, they are still fundamentally point-estimation methods. Here, methods that can recover the full posterior distribution are discussed.

Magni et al. (1998) consider three estimation cases: interpolation, where the system's impulse response is the identity operator, input estimation with a linear time-invariant system (deconvolution), and estimation with a linear time-varying system, where the input-output relationship is represented by a Fredholm integral equation instead of a convolution. Like Sparacino and Cobelli (1996) and De Nicolao et al. (1997), piecewise constant functions with quadratic regularisation of the first or second derivatives are considered. This model is treated in the Bayesian framework, and considers the regularisation parameter to be just another parameter to be estimated together with the input function. Estimation is performed using Markov chain Monte Carlo (MCMC). For the linear systems under consideration, it is possible to generate samples for the regularisation parameter conditional on the input function, since the prior distribution of the regularisation parameter is chosen to be a *conjugate prior* to the input function coefficients. Additionally, it is also possible to draw samples for the input function given the regularisation parameter and the data. This is because the joint distribution of the input function coefficients and the data is Gaussian for a fixed value of the regularisation parameter. Hence, the distribution of the input function coefficients conditional on the data is Gaussian too (Särkkä 2013). By alternating between drawing samples for the regularisation parameter and the input function, samples from the posterior distribution are obtained. This is an example of *Gibbs sampling* (Section 3.3.5). Magni et al. (1998) also consider joint system identification and deconvolution.

Pillonetto et al. (2002) propose similar methods to Magni et al. (1998). Here, the input function is modelled in the log domain, so the input to the dynamical system is the exponentiation of the modelled function. This has two major benefits. First, nonnegativity constraints are automatically satisfied. Second, it will result in a

model where large changes in the input function are more probable when the function value is high, something that may seem more plausible than assuming that a large change is equally probable independent of the function value. This can be especially helpful when datasets are sparsely sampled in the later stages of an experiment, where the input function has decayed to a low value. Note that this model results in a nonlinear estimation problem, even if the system is linear. Similar to Magni et al. (1998), estimation is performed with MCMC sampling, where noise variances are updated using Gibbs sampling. Input function values are updated using a random walk Metropolis-Hastings (RWMH) algorithm (Section 3.3.4), where the proposal distribution is chosen by setting the regularisation parameter to a reasonable value based on the discrepancy criterion (Twomey 1965), optimising the log-posterior with respect to the input function, and computing the Hessian at the optimum. The proposal density is chosen to be a multivariate Gaussian centred on the current parameter value, with a covariance matrix equal to the inverse Hessian, scaled by a factor empirically chosen to achieve good performance.

Bell and Pillonetto (2004) consider estimation for cases where the input-output mapping is a possibly nonlinear function, which therefore can be used for nonlinear dynamical systems. Emphasis is placed on estimating unknown parameters for the prior of the input function as well as for the system model. To accomplish this, a method is developed where the marginal likelihood of these parameters is maximised, with the function integrated out. The input function is parameterised by considering it to be an element in a reproducing kernel Hilbert space (RKHS) (Wahba 1990). For a given RKHS, a set of basis functions can be obtained by computing the eigenfunctions to the *reproducing kernel* of the space. Eigenfunctions are derived for the reproducing kernels corresponding to quadratic penalisation of first and second derivatives. The idea behind this parameterisation is that it is often possible to represent probable function realisations with relatively few basis functions, reducing the dimensionality of the estimation problem. These ideas are presented in more detail in Section 3.1, using the formalism of Gaussian processes, where the reproducing kernel is interpreted as a covariance function (Rasmussen and Williams 2006).

Pillonetto and Bell (2007) use the eigenfunction-based approach of Bell and Pillonetto (2004) to solve linear deconvolution problems. The problems are treated in a Bayesian framework, similar to Magni et al. (1998). Two methods are presented: the first obtains a MAP estimate of parameters, where the input can be marginalised out because the model is linear and Gaussian. The second performs full Bayesian inference with MCMC, where the parameters are sampled using RWMH, and the function basis coefficients are sampled using Gibbs sampling.

A good overview of these regularisation-based approaches to linear systems can be found in Sparacino et al. (2011). Emphasis is placed on point estimates, with a brief note that MCMC methods can be used to estimate uncertainty.

2.3.4 Other methods

Another way of choosing a prior over input functions is to use the *principle of maximum entropy* (Skilling 1988). Here, the input function is discretised in time, and its *entropy*, a measure of its information content, is computed (see Section 2.4.1). This method was originally developed for enhancing images in astronomy applications (Skilling and Bryan 1984). The idea is that for all input functions that agree well enough with observed data, the best estimate is obtained when the function with the largest entropy is selected. Large entropy corresponds to a function that is relatively flat. Agreement with data is usually measured by the χ^2 statistic, the weighted sum of squares of the prediction error. The entropy is maximised subject to the constraint that the χ^2 statistic does not exceed a prespecified value. This can be viewed as regularisation where the regularisation parameter is determined by the discrepancy criterion (Twomey 1965). An appealing property of this method is that positivity is enforced, since entropy is defined only for positive values. This method has been applied to pharmacokinetic data of paracetamol by Charter and Gull (1987). In addition, Hattersley et al. (2008) have used similar methods for estimating free light-chain production in patients with multiple myeloma.

Madden et al. (1996) present and compare six deconvolution techniques on a number of synthetic benchmark problems:

1. The first method is based on Fourier transforming the impulse response I and the system output u, and using the fact that convolution in the time-domain corresponds to pointwise multiplication in the Fourier domain:

$$\mathscr{F}(y) = \mathscr{F}(I) \cdot \mathscr{F}(u) \tag{2.14}$$

   where $\mathscr{F}$ denotes the Fourier transform. The input function u can be obtained by:

$$u = \mathscr{F}^{-1}\left(\frac{\mathscr{F}(y)}{\mathscr{F}(I)}\right). \tag{2.15}$$

   To ensure that the number of points in u and y match, the observations are interpolated before Fourier transformation.

2. Numerical deconvolution using system identification techniques (Vajda et al. 1988). Here, the relationship between the impulse response and the output is modelled by a higher-order linear differential equation. By interpolating the impulse response and the measurements using splines, a linear regression model can be constructed, and solved analytically.

3. CODE (constrained deconvolution) (Hovorka et al. 1998). This is an implementation of regularised least-squares with nonnegativity constraints and quadratic penalties, similar to the methods presented by Verotta (1996).

4. Spline models, similar to the methods presented by Verotta (1996).

5. Maximum entropy methods. In this case, the input function is assumed to be linear between the measurements.

6. A custom-made genetic algorithm. Rather than demanding that the likelihood or sum of squares in total is small, this method demands that observed concentration profiles are close to the prediction *at every measurement time point*. Initially, the input function is assumed to be a single line segment. At each iteration, the algorithm generates several perturbations of the current input function. The perturbation with the best agreement with the data is selected. This is repeated until agreement with the data no longer improves. At that point, the input function is split into two segments, and the process is repeated. The number of line segments continues to be doubled until the data fit reaches or exceeds a prespecified tolerance.

2.4 A framework for input estimation

Most of the methods presented in the previous section have in common that they represent the input function as a linear combination of basis functions. To avoid ill-posedness, they either restrict the input function to be a combination of a small set of basis functions, or they include a frequentist regularisation term or a Bayesian prior. The solution being sought is either a point estimate based on the ML or MAP criteria, or a posterior distribution. In the former case, inference is performed using optimisation, while in the latter case, sampling-based or Bayesian approximate inference methods are used.

Based on these considerations, a taxonomy of input-estimation methods can be developed. We see that the methods largely can be described as being a combination of a choice of:

- A prior (in the Bayesian interpretation) or regularisation term (in the frequentist interpretation).
- A function parameterisation. In the methods above, this is often piecewise constant functions, but it can also be more elaborate functions such as splines.
- A desired statistical quantity. This represents the type of output given by the method. This can be either a point estimate, or a full posterior distribution over functions.
- An inference algorithm. These can be optimisation algorithms for point estimates, or algorithms such as MCMC that return a distribution.

The likelihood — the probability of the observed data given an input function — is given by the model of the system dynamics together with the measurement noise model. Note that these are not considered to be part of the input-estimation method. Rather, they are assumed to be determined by the problem at hand. In practice, it may be beneficial to combine modelling and input estimation, making this distinction less applicable. Several methods presented in Section 2.3 do make assumptions about the likelihood — methods based on least-squares criteria implicitly or explicitly assume that the measurement noise is Gaussian. The prior over the input function and any other parameters together with the likelihood constitute a full probabilistic model. In this work, the term *inference algorithm* refers to the algorithm that computes a point estimate or a posterior distribution, given such a probabilistic model. In contrast, the term *input-estimation method* refers to the combination of prior, function parameterisation, desired statistical quantity, and inference algorithm.

The purpose of this taxonomy is to make it easy to compare and contrast existing methods. It makes it explicit what assumptions these methods make, which can help in picking a method for a particular problem. By separating the methods into parts, it becomes easier to create new methods by, for example, replacing the inference algorithm while keeping the assumptions about the prior distribution. However, it should be noted that not all methods presented above can be mapped cleanly onto this taxonomy. For example, methods that interpolate the measurements, and treat the interpolant as if it were the measurements, may not have any obvious statistical interpretation. In this thesis, we will only consider methods that do have such an interpretation.

In general, the estimated input function will depend on the function parameterisation as well as the prior. If a function parameterisation that depends on only a few parameters is used, it is possible to avoid assigning a prior altogether. In this

case, the inferred function is determined by the parameterisation together with the likelihood, and the method can be considered to be a parametric method. On the other hand, for methods with a sufficiently flexible parameterisation, the inferred function will be determined by the prior together with the likelihood. Here, a prior is in general necessary to prevent overfitting. This thesis is mainly concerned with the latter class of methods, which are considered nonparametric in the sense that the input function is, in principle, not determined by a parametric model. Note that there is no universal agreement of what constitutes a nonparametric method. As an example, non-penalised least-squares based on B-splines is presented as a nonparametric method in Verotta (1996) on the grounds that splines are flexible functions that do not correspond to any strong physiological assumptions, while the same method is considered to be parametric in Sparacino and Cobelli (1996). From a purely mathematical point of view, sums of exponentials and B-splines are merely different choices of basis functions. Hence, there is no compelling reason to classify one of them as parametric and the other as nonparametric. For our purposes, we will consider a method to be parametric only if the purpose of the function parameterisation is to constrain the set of possible input functions.

The type of desired statistical quantity determines which inference algorithms can be used. Optimisation algorithms are used to provide point estimates such as maximum likelihood or maximum a posteriori, while full posteriors require specialised algorithms such as Markov chain Monte Carlo or other Bayesian algorithms. We will now describe each part of the framework in more detail.

2.4.1 Choice of prior

In this section, priors for scalar input functions are discussed. For vector-valued input functions, priors can be independently assigned to each component, as long as they are assumed to be *a priori* independent. Assigning priors jointly over all components will not be considered further in this work. In the input-estimation literature, the most common type of prior for a scalar input function $u(t)$ is given by

$$p(u(t)) \propto e^{-\frac{\tau}{2}\int_{t_i}^{t_f}\left(\frac{\mathrm{d}^j u}{\mathrm{d}t^j}\right)^2 \mathrm{d}t} \tag{2.16}$$

where τ is the regularisation constant, and j is the order of the derivative, which is typically 1 or 2. The logarithm of this prior is proportional to the L^2 norm of the input function's derivative. Seen as a function of $u(t)$ this expression is not a norm, but a seminorm, since any polynomial of degree $j-1$ can be added to the function without changing the value of the expression. This makes the prior improper, since it

cannot be integrated to give a finite value. This is not necessarily a problem, as long as the corresponding posterior is proper. If desired, it is possible to add additional factors to the prior to make it proper. These priors will assign lower probabilities to functions that have large higher-order derivatives, making highly oscillatory functions less probable. Penalising the first derivative corresponds to the intuitive notion that rapid changes in the function value are *a priori* improbable. Similarly, penalising the second derivative corresponds to the notion that rapid changes in the slope of the function are improbable. Penalising a higher-order derivative will result in a higher degree of smoothness. This can be seen by noting that differentiation is a high-pass operation. Informally, this means that the derivative will usually be "noisier" than the function itself. Hence, by forcing the second derivative to be smooth, the function itself is forced to be even smoother.

Maximum entropy-based methods define a prior by discretising the input function in time into N_B points, and letting

$$p(u(t)) \propto e^{-\tau \sum_{k=0}^{N_B-1} u_k \log \frac{u_k}{m_k}} \tag{2.17}$$

where u_k is the input function value at time t_k. The value in the denominator, m_k, is considered to be a "baseline" value, which is the best guess for that value unless the data suggest otherwise. Hattersley et al. (2008) define m_k to be the mean of adjacent function values, $(u_{k-1} + u_{k+1})/2$. This prior discourages large deviations from the baseline. For a straight line, where $u_k = m_k$ for all k, the unnormalised log-prior evaluates to zero. Any deviation from this line will result in a smaller probability density (Fig. 2.2).

A large class of potentially useful priors can be constructed by modelling the input function as a *Gaussian process*. A Gaussian process is defined as a stochastic process for which all finite-dimensional distributions are Gaussian. This means that if n time points are selected, $t_0, \ldots t_{n-1}$, the vector of function values $\begin{bmatrix} u(t_1) & \ldots & u(t_n) \end{bmatrix}$ is a multivariate Gaussian (Rasmussen and Williams 2006). The statistics of the process can be uniquely defined by a *mean function* $m(t)$, which represents the mean of $u(t)$, and a *covariance function* $K(s,t)$ which represents the covariance between $u(s)$ and $u(t)$ This idea is illustrated in Fig. 2.3. Using the L^2 norm of the jth derivative as a log-prior is a special case of modelling the input function as a Gaussian process. In particular, the choice of $j = 1$ corresponds to $m(t) = 0$ and $K(s,t) = 1/\tau \cdot \min\{s,t\}$, and the choice of $j = 2$ corresponds to $m(t) = 0$ and $K(s,t) = \frac{1}{\tau}\frac{\min\{s,t\}^2}{6}(3\max\{s,t\} - \min\{s,t\})$ (Bell and Pillonetto 2004). Here, the regularisation parameter τ can be interpreted as the process noise precision, which is the inverse of the process noise variance.

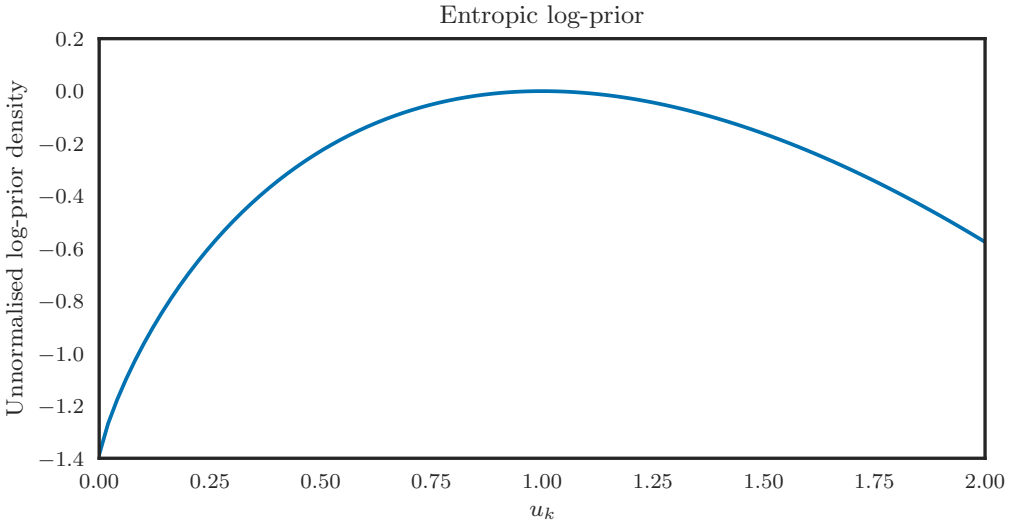

Figure 2.2: Value of the entropy-based log-prior of Eq. (2.17), where the input function at one time point, u_k, is varied, while the function at all other time points is kept fixed at 1. The log-prior attains its largest value when u_k is equal to its neighbours. Here $\tau = 1$.

In pharmacokinetic models, it is often desirable to impose nonnegativity constraints, to ensure that the input function does not attain unphysical negative values. This requirement can be added to the problem by assigning a zero probability to functions that at any point drop below 0. An alternative is to model the input function in the logarithmic domain, such that the prior is placed on $\log u(t)$ rather than on $u(t)$ directly. The latter approach can also make for more plausible models, since they capture a notion that if a function value is large, large changes in the value are more probable. As an example of why this may be more realistic, consider a typical pharmacokinetic experiment, where the input rate is large in the initial stages of the experiment, and close to zero in the later stages (see Fig. 2.1). When using an input model where the first derivative of the input function is penalised, a change in the input function by a certain amount will be penalised equally regardless of where the change occurs. In contrast, a logarithmic model penalises changes proportionally to the current function value. Hence, a large change in the region where the input value is close to zero would be considered less probable than a large change during the initial stage when the input value is large.

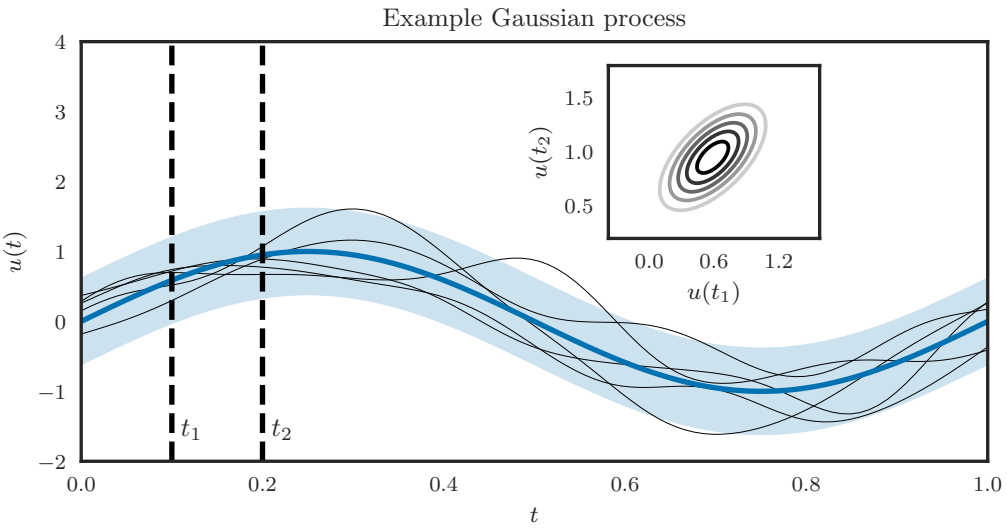

Figure 2.3: Example: Gaussian process $u(t)$ with mean function $m(t) = \sin(2\pi t)$ and covariance function $K(s,t) = 0.1e^{-\frac{1}{2}\frac{(s-t)^2}{10^{-3}}}$. This is an example of a squared exponential covariance function. The thick line shows the mean function, and the shaded area covers ± 1.96 standard deviations of the variance at each point. The thin lines are example realisations of the stochastic process. The function values at points $t_1 = 0.1$ and $t_2 = 0.2$ form a bivariate Gaussian distribution, whose mean vector and covariance matrix can be computed from $m(t)$ and $K(s,t)$ (shown in the inset). The covariance function assigns larger correlations between points that are close in time.

2.4.2 Function parameterisation

In nonparametric methods, priors are assigned directly on the space of functions, and function parameterisation has in principle no role. In practice however, the functions are discretised so that they can be represented by a finite-dimensional parameter vector. After discretisation, the input-estimation problem can be treated as a parameter-estimation problem.

The function parameterisation should ideally be flexible enough so that it can accurately represent any function that has a significant probability under the posterior. This can be easily accomplished by making a fine-grained parameterisation with a large number of parameters. On the other hand, such a high-dimensional parameter estimation problem can result in long computation times.

In *basis function models*, the input function is represented as a linear combination of a set of N_B fixed basis functions $B_j(t)$:

$$u(t) = \sum_{j=0}^{N_B-1} a_j B_j(t). \tag{2.18}$$

Here, the basis function coefficients a_j are the parameters to be estimated. In this model, there is a linear relationship between the parameters and the input function. It is also possible to allow nonlinear relationships. As an example, each basis function could be parameterised though a nonlinear mapping.

The number of basis functions can be fixed, or can be inferred from the data, as is done in the genetic algorithm presented by Madden et al. (1996). In Bayesian inference, this results in a *transdimensional* problem, where the number of model parameters is one of the parameters to be estimated. Inference algorithms that can handle these problems include Reversible Jump MCMC (Green 1995; Green 2003) and product-space MCMC (Carlin and Chib 1995).

A common choice of basis functions is piecewise constant functions. The time-interval $[t_i, t_f]$ is divided into several subintervals, and the input function is assumed to be constant over each interval. In this case, the jth basis function has the value 1 in the jth time interval and 0 elsewhere, and the jth basis function coefficient is the function value over that interval. This choice of basis functions makes the problem sparse in the sense that, at any time point, the dynamics of the system are governed by a single basis function coefficient, something that can be exploited by optimal-control methods (Andersson 2013). The disadvantage of these functions is that to make realistic-looking functions, small discretisation steps are needed. This results in a high-dimensional problem which can cause computational difficulties. Alternatively, piecewise linear functions can be used. They carry similar advantages and disadvantages as piecewise constant functions.

Basis splines (B-splines) (Boor 1986; Schumaker 2007) can be used to obtain functions that are differentiable to any desired degree. Each basis function is a polynomial of degree $k-1$, where k is called the order of the spline. To represent a function over the time interval $[t_i, t_f]$, a set of time points $t_0 = t_i < t_1 < t_2 \ldots t_{n-1}$, called *knots*, is defined. The jth B-spline basis function of order k is defined recursively by:

$$B_{j,1}(t) = \begin{cases} 1 & \text{if } t_j \leq t \leq t_{j+1} \\ 0 & \text{otherwise} \end{cases} \tag{2.19}$$

$$B_{j,k}(t) = \frac{t - t_j}{t_{j+k-1} - t_j} B_{j,k-1}(t) + \frac{t_{j+k} - t}{t_{j+k} - t_{j+1}} B_{j+1,k-1}(t). \tag{2.20}$$

The resulting function is continuously differentiable $k-1$ times. It can be seen that piecewise constant and piecewise linear basis functions are special cases of B-splines, of order 1 and 2. B-splines of higher order can represent realistic-looking functions with a relatively small number of knots. They can therefore be preferable to simpler basis functions if high-dimensional problems are to be avoided. However, when the number of knots is small, it is difficult to assess whether this parameterisation can represent any function that has a non-negligible probability under the posterior.

One way to select a set of basis functions is to consider the following: given that only a finite set of basis functions is used, it is desirable to select these such that functions that are probable under the prior can be represented with small discretisation error, at the expense of a higher discretisation error for improbable functions. The *Karhunen-Loève* basis functions are derived from these considerations. These functions have the additional advantage that the prior over the function itself can be translated to a prior over the basis function coefficients. This way, the function-estimation problem can be easily converted to a parameter-estimation problem. The Karhunen-Loève basis functions $\phi_j(t)$, corresponding to a zero-mean Gaussian process prior with covariance function $K(s,t)$, are defined as the solutions to the eigenvalue problem (Levy 2008; Wang 2008):

$$\int_{t_i}^{t_f} K(s,t)\phi_j(s)\,\mathrm{d}s = \lambda_j\phi_j(t). \tag{2.21}$$

The basis function coefficients are independent zero-mean Gaussian random variables with variance λ_j. The basis functions have to be derived separately for each prior. For Gaussian processes with non-zero means, additional basis functions can be added. A more detailed explanation is given in Section 3.1.

2.4.3 Desired statistical quantity

Considering the methods described in Section 2.3, it can be seen that most methods find the most probable solution, either the penalised maximum likelihood estimate or the maximum a posteriori estimate depending on interpretation. Methods using the Bayesian interpretation can also attempt to recover the posterior distribution over input functions. The posterior is more informative than a point estimate. Since it is a probability distribution, it allows the computation of arbitrary statistics such as means, medians and quantiles. This is important for assessing estimation uncertainty. Since our applications typically have sparse and noisy data, the uncertainty can be large, in that there is a broad range of input functions that are all reasonably probable. A point estimate will not provide any information on this uncertainty.

However, determining the posterior can be computationally much more expensive. A user working under time constraints, as is common in industry, might prefer a faster algorithm.

2.4.4 Choice of inference algorithm

The choice of inference algorithm is a vast topic. A brief overview is given in the following two sections. Detailed aspects of the most important algorithms are covered in Chapter 3.

2.5 Inference algorithms for point estimates

Finding point estimates for statistical models is an optimisation problem. In traditional input estimation, where only linear systems are considered, the resulting optimisation problem typically either has a closed-form solution, or can be efficiently solved using standard methods (Verotta 1996). In contrast, nonlinear input estimation involves solving an optimisation problem where the cost function can only be expressed in terms of a solution to a system of ODEs. This makes the cost function expensive to compute, and can result in a highly nonlinear problem that is difficult to solve. Optimisation problems involving ODEs are studied in the field of *optimal control theory* (Betts 2010; Biegler 2010; Andersson 2013). Optimal-control methods can broadly be classified into two categories, *indirect* and *direct methods* (Rao 2009).

Indirect methods (Pontryagin 1987; Bryson and Ho 1975; Liberzon 2012) are a generalisation of *calculus of variations*. While optimisation concerns finding a value or a set of values that optimises a given cost function, the calculus of variations concerns finding a *function* that optimises a given cost *functional*. Optimal control additionally has to make sure that the solution satisfies the constraints imposed by the system of ODEs. This is done by augmenting the original system with an additional set of states $\boldsymbol{\lambda}$, called *costates*, that have a role analogous to that of Lagrange multipliers in constrained optimisation. A necessary condition for optimality is that the input function at each point in time is chosen so that the *Hamiltonian* $H = \boldsymbol{\lambda}^T \cdot \mathbf{f}(t, \mathbf{x}, \mathbf{u}) + L(t, \mathbf{x}, \mathbf{u})$ is maximised, or minimised, depending on the sign convention. This is known as *Pontryagin's Maximum* (or *Minimum*) *Principle*. Here, L is the running cost, constructed such that the log posterior is given by $\int_{t_i}^{t_f} L(t, \mathbf{x}, \mathbf{u})\,\mathrm{d}t$. In this way, $\mathbf{u}(t)$ can be eliminated from the problem. The resulting system of ODEs generally has some fixed initial as well as terminal values, making it a boundary-value problem. This makes the optimisation problem a root-finding problem, where the goal is to find initial values that produce the desired

terminal values. Once such a solution is found, the input function can be obtained by substituting it back into the original equations.

In direct methods (Betts 2010; Biegler 2010), the input function is first discretised using schemes such as the basis function expansions previously mentioned. The discretisation process yields a problem that can be solved by general-purpose optimisation algorithms. For an overview of such methods, see Nocedal and Wright (2006).

Indirect methods are elegant, and they can provide analytical solutions for simple problems where the system of ODEs can be solved analytically. However, when the ODEs have to be solved numerically, the resulting root-finding problem is often highly nonlinear and numerically challenging to solve. Indirect methods also require the user to derive the costate equations and the Hamiltonian. For these reasons, direct methods are often preferred (Rao 2009).

Both indirect and direct methods proceed by repeatedly solving a system of ODEs, varying the free parameters to find a solution. In the indirect case, the initial conditions are varied to make the solution have the desired terminal state. In the direct case, the input function basis coefficients are varied to optimise the log probability. The most straightforward solution for both of these cases is: at each iteration, assign values to these parameters, and simulate the complete trajectory. More sophisticated methods, such as *multiple shooting* and *collocation*, divide the trajectory into multiple intervals, and add constraints to the problem to ensure that the complete trajectory is continuous. This can make the problem less nonlinear and therefore easier to solve, at the expense of introducing additional parameters. These methods are described in Section 3.2.

Many optimisation algorithms allow for constrained optimisation, where the objective is to find an optimum subject to equality or inequality constraints. This makes it easy to impose constraints such as requiring the input function to be nonnegative.

2.6 Algorithms for full Bayesian inference

While algorithms for point estimates provide a single value such as the ML or MAP estimate, algorithms for full Bayesian inference provide the posterior distribution. Since closed-form solutions are generally not available, such methods are necessarily based on approximations. Broadly, they can be divided into (MacKay 2003, Part IV):

Sampling-based methods, or *Monte Carlo methods*, which generate a set of samples drawn from the posterior distribution. Any statistical quantities

of interest can be approximated by the empirical quantities given by the set of samples. These methods are designed to converge to the true values as the number of samples increases. For nontrivial models, generating these samples is in itself a difficult problem. Methods for this can be computationally expensive, but have the advantage that the approximation error can be made arbitrarily small by drawing many samples.

Analytical approximations, which approximate the true distribution by simpler distributions for which exact solutions exist. These methods are often faster than sampling-based methods, but it can be difficult to assess whether the approximations are good.

Another way to classify algorithms is:

Batch methods, which treat the model as a black box, where the log posterior for a complete set of parameters and measurements is computed in a single step. The underlying model may or may not be a dynamical system. These methods are general-purpose and place little restriction on the models and the statistical quantities that can be computed.

Sequential methods, which exploit the fact that the model describes a system evolving over time. By sequentially processing the measurements one at a time, they may be able to solve the inference problem at a lower computational cost than batch methods. These methods are especially helpful for long time series. However, jointly estimating the full time-evolving input function together with fixed parameters can be less straightforward than for batch methods.

2.6.1 Markov chain Monte Carlo

Markov chain Monte Carlo (MCMC) is a sampling-based batch algorithm that is widely used in Bayesian statistics. This algorithm generates samples using a Markov chain, where each sample drawn from the posterior is generated by perturbing the previously drawn sample. This results in the set of samples forming a Markov chain, which gives the algorithm its name. The resulting samples are not independent, but are marginally distributed correctly (Brooks et al. 2011; Gilks et al. 1996). The greatest advantage of MCMC is its generality. It is relatively easy to construct a correct Markov chain for arbitrarily complicated models. However, constructing chains that are *computationally efficient* is considerably more difficult. Because of the dependence between samples, it can also be difficult to assess whether a finite set of samples is representative of the true distribution. Most MCMC algorithms

in use are variants of the Metropolis-Hastings algorithm (Metropolis et al. 1953; Hastings 1970), where new samples are generated by sampling from a user-defined *proposal distribution*, followed by an accept-reject step. The computational efficiency of the algorithm depends crucially on the choice of proposal distribution. Designing a distribution that is suitable for the problem at hand is non-trivial. For this reason, many methods to construct these proposal distributions have been developed, including random walk Metropolis-Hastings, Gibbs sampling (Geman and Geman 1984), the Metropolis-adjusted Langevin algorithm (Roberts and Stramer 2002), and Hamiltonian Monte Carlo (Duane et al. 1987; Neal 2011). More recently, algorithms based on Riemannian manifolds have been proposed (Calderhead 2011; Girolami and Calderhead 2011). The manifold methods provide a more principled way to construct proposals that provide good efficiency even for complicated posteriors. Because of its generality, MCMC is the Bayesian inference algorithm of choice in this thesis, and it is described in more detail in Section 3.3.

2.6.2 Sequential Monte Carlo

Sequential Monte Carlo (SMC) methods (Del Moral et al. 2006; Doucet et al. 2006) are a broad class of algorithms. For the purposes here, focus will be on the variants of this algorithm that can be interpreted as tempering algorithms. The idea behind the methods is to construct a sequence of distributions $\pi_0 \dots \pi_n$, where the initial distribution is a simple distribution from which it is easy to sample, and the final distribution π_n is the posterior. The intermediate distributions are chosen to interpolate between these two, such that π_{i+1} is similar to π_i. For Bayesian models, one choice of sequence of distributions is the *power posterior*, $\pi_i = p(\boldsymbol{\theta})p(\mathbf{y}|\boldsymbol{\theta})^{\beta_i}$, with $\beta_0 = 0 \leq \beta_1 \dots \leq \beta_n = 1$.

Given samples from one distribution π_i, it is possible to generate samples from another distribution π_{i+1} using the method of *importance sampling* followed by *resampling* (Del Moral et al. 2006). Starting with a set of samples from π_0, importance sampling is used to obtain a set of samples from π_1. In the next step, these samples are used to obtain a set of samples from π_2. This procedure is repeated until a set of samples from π_n has been generated. While importance sampling is usually inefficient in high dimensions, it works in SMC because the distributions are designed to be similar to each other.

Assessing the quality of the generated samples is easier in SMC methods than in MCMC. SMC methods may also be able to handle multimodal distributions better than MCMC, as the MCMC mechanism of generating new samples based on previous ones can cause the algorithm to become trapped in local maxima. However, SMC can

be computationally expensive, as a large number of samples has to be generated for every intermediate distribution between π_0 and π_n. Since the importance sampling step requires adjacent distributions to be similar, a large number of intermediate distributions may be required. Even the relatively simple example problems in Del Moral et al. (2006) use hundreds of intermediate distributions.

2.6.3 Particle filters

A particle filter is a sequential sampling method for state estimation in dynamical models. These can be adapted to input estimation by modelling the input function as a set of states. While not all statistical models of input functions can be cast this way, important models such as penalisation of first or second derivatives can. In the most basic form of a particle filter, the *bootstrap filter*, a set of trajectories is simulated forwards in time according to the dynamical model. When a time point for which a measurement exists is encountered, that measurement is incorporated by importance sampling followed by resampling (Gordon et al. 1993). Only *marginal filtering distributions* are computed, meaning that the distribution of the states and inputs for that time point is given, conditioned on previous but not future measurements. Extensions to this algorithm exist that use simulation backwards in time to provide the joint distribution of the complete state trajectory, conditioned on all measurements (Godsill et al. 2004; Lindsten and Schön 2013).

Basic particle filters do not allow for estimation of static parameters, as they require the estimated quantities to vary in time. To allow joint estimation of state trajectories and static parameters, particle filtering can be combined with MCMC sampling, resulting in algorithms such as the particle marginal Metropolis-Hastings and particle Gibbs methods (Andrieu et al. 2010; Lindsten 2013).

The bootstrap filter can be inefficient for vague priors and sparsely sampled data, as few or no particles may end up close to the measurements. This can be partly remedied by using more sophisticated updating mechanisms, such as auxiliary particle filtering (Pitt and Shephard 1999), or specialised methods for sparse sampling (Del Moral and Murray 2015).

For reviews of particle filtering, see Arulampalam et al. (2002), Chen (2003), Cappé et al. (2007) and Doucet and Johansen (2009). Finally, it can be noted that in the literature, the term sequential Monte Carlo is often used synonymously with particle filtering. A particle filter is indeed a special case of SMC, where π_i is the distribution of state trajectories up to time step i. SMC is a more general notion, where the intermediate distributions do not necessarily have any relationship with the time evolution of the modelled system.

2.6.4 Variational Bayes and expectation propagation

Variational Bayesian methods (Jordan et al. 1999; Beal 2003; Daunizeau et al. 2009; Bishop 2006; Wainwright and Jordan 2008) are batch methods that approximate the true posterior $p(\boldsymbol{\theta})$ by a simpler distribution $q(\boldsymbol{\theta})$ for which analytical calculations can be performed. A measure of similarity between those distributions is given by the *Kullback-Leibler divergence*, $D_{KL}(p \parallel q)$ defined by

$$D_{KL}(q \parallel p) = \int q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \, \mathrm{d}\boldsymbol{\theta}. \tag{2.22}$$

The distribution $q(\boldsymbol{\theta})$ is chosen to minimise $D_{KL}(q \parallel p)$. While $D_{KL}(q \parallel p)$ cannot be computed directly, it is possible to compute another quantity, the *variational free energy*, for which it is known that minimising D_{KL} is equivalent to maximising (or minimising, depending on sign convention) the variational free energy.

Typically, the posterior is approximated by dividing the parameter vector into n blocks, $\boldsymbol{\theta} = \left[\boldsymbol{\theta}_0^T, \quad \ldots, \quad \boldsymbol{\theta}_{n-1}^T\right]^T$, and assuming that the blocks are independent of each other in the posterior, so that the approximating distribution can be factored as $q(\boldsymbol{\theta}) = \prod_{i=0}^{n-1} q_i(\boldsymbol{\theta}_i)$. As an example, an input-estimation method may assign the basis function coefficients to one block, and the regularisation parameter to another. The parameters in each factor are iteratively updated to minimise $D_{KL}(q \parallel p)$. For some models, this factorisation is sufficient to make the distribution analytically tractable. If necessary, additional approximations can be introduced, such as assuming that each factor has a certain functional form. For some models, the computations can be made more efficient by exploiting the dependency structure between parameters, using message-passing techniques (Winn 2004). These methods blur the difference between batch and sequential methods.

A related algorithm is expectation propagation (Minka 2001a; Minka 2001b). This is a message-passing algorithm that aims to minimise the Kullback-Leibler divergence $D_{KL}(p \parallel q)$. Note that this is different from $D_{KL}(q \parallel p)$. In the former case, $q(\boldsymbol{\theta})$ will assign a high density to any point for which $p(\boldsymbol{\theta})$ is high, even if it means having to assign a high density for points where $p(\boldsymbol{\theta})$ is low. In the latter case, $q(\boldsymbol{\theta})$ will assign a low density to any point for which $p(\boldsymbol{\theta})$ is low, even if it means having to assign a low density for points where $p(\boldsymbol{\theta})$ is high. For problems where $p(\boldsymbol{\theta})$ is a multimodal distribution, variational Bayes tends to select a single mode from $p(\boldsymbol{\theta})$, while expectation propagation will select a wide distribution that covers all modes of $p(\boldsymbol{\theta})$ (Bishop 2006).

2.6.5 Extended and unscented Kalman filters

The Kalman filter (Kalman 1960) is a sequential state estimation algorithm. While the original derivation of the Kalman filter is non-probabilistic, it can be interpreted as an exact algorithm for computing marginal filtering densities in dynamical Bayesian estimation problems, assuming that the dynamical and measurement models are linear, and all noise is Gaussian and enters linearly into the equations. As a consequence, the distribution of the state at each time point is also Gaussian. Extensions to nonlinear models include the extended Kalman filter (EKF) and the unscented Kalman filter (UKF). In the EKF (Jazwinski 2007), the system is linearised around the current mean estimate at each time step. In the UKF (Julier and Uhlmann 2004; Merwe 2004), the system state is propagated between time steps by selecting a set of points, called *sigma points*, from the current state distribution, transforming these according to the dynamical equations, and using the transformed points to compute the mean vector and covariance matrix of the new state. This transformation has similarities to Monte Carlo methods, the main difference being that the sigma points are selected deterministically.

The considerations involved with Kalman filters are in many respects similar to those with particle filters. Since the filters are fundamentally state-estimation algorithms, the input has to be modelled as a state. The basic versions of the filter provide only marginal filtering distributions. To obtain distributions conditioned on all measurements, the Kalman filter, which works forwards in time, can be complemented with an additional pass working backwards in time. This results in the Rauch-Tung-Striebel smoother (Rauch et al. 1965). Similar solutions exist for the EKF and UKF (Särkkä 2008).

Since these filters assume that all distributions are Gaussian, they do not impose any nonnegativity constraints, although some work to introduce such constraints exists (Simon 2010; Kolås et al. 2009). Although it is possible to model the input function in the log domain, doing so creates a highly nonlinear problem, for which Gaussian approximations may not be appropriate.

For a good and comprehensive introduction to variants of Kalman filtering, see Särkkä (2013). This book also includes an introduction to particle filtering.

2.7 Choice of inference algorithms and software

As shown in the previous sections, there is a great variety of inference algorithms that could be used. However, it is not feasible to investigate all of them within the scope of this thesis. Instead, the algorithms that appear to be the most promising place to

start have been investigated. Both point-estimation and fully Bayesian algorithms are considered. The reason for this is that theses kinds of estimates have complementary strengths. Bayesian algorithms are able to provide uncertainty estimates, which can be important, especially when the data are sparse and noisy. However, these algorithms can be computationally expensive. They can also be difficult to use, difficult to diagnose, and require the user to have a certain amount of expertise in the internal mechanisms of the algorithms. These points will be explored in detail in the case studies. In light of this, point-estimation algorithms are an interesting alternative to Bayesian methods, particularly for modellers based in industry, where the time for them to become experts in the inference algorithms may not be available.

For point estimates, direct optimal control algorithms were investigated. These algorithms are conceptually simple, and fit well into the framework presented above, allowing arbitrary priors and basis function models. In contrast, indirect algorithms are more involved to apply.

For Bayesian estimation, there is a choice between sampling-based and approximate inference algorithms. An advantage of sampling-based algorithms is that they are asymptotically exact. This means that if the accuracy of the approximation is in doubt, it is always possible to improve it by running the sampling process for a longer time. In contrast, approximate inference methods rely on making assumptions whose validity can be difficult to assess. One possible solution is to first develop sampling-based algorithms. Once these have proved to be reliable, approximate inference algorithms can then be developed, and validated using their sampling-based counterparts. For this reason, sampling-based algorithms have been chosen for this project. Approximate algorithms would be an interesting follow-up project.

There is a choice to be made between batch and sequential algorithms. While sequential algorithms may seem like a natural fit for input-estimation problems, they have limitations. First, they limit what kind of input models can be used, as they have to fit into the state-space model framework. Second, many of these algorithms do not, in their basic form, provide estimates of the complete joint posterior, but only of certain marginal distributions. While this limitation can often be overcome by using an extension of the basic algorithm, or by modifying the problem formulation, it still decreases flexibility. This is counter to the objective that the input-estimation methods should be as automated and general-purpose as possible. An additional issue with sequential algorithms is that they can struggle with sparse data, as discussed in Section 2.6.3. Additionally, the advantages of sequential over batch algorithms are most significant for long time series. For these reasons, sequential algorithms may be less advantageous for typical pharmacokinetic and pharmacodynamic data.

Based on these considerations, MCMC is the algorithm that was used for Bayesian inference in this work. Four algorithms for constructing proposal distributions were evaluated. Two of these are variants of the random walk Metropolis-Hastings algorithm: a single-component variant, where each basis function coefficient is updated separately, and a block variant, where all basis function coefficients are updated jointly. The random walk Metropolis-Hastings algorithm is well established and has previously been used in physiological applications (Pillonetto et al. 2002; Pillonetto and Bell 2007). This makes the algorithm a suitable baseline for making comparisons with more modern algorithms. The third algorithm is the Metropolis-adjusted Langevin algorithm, which uses the gradient of the log-posterior to construct more efficient proposals. Finally, the fourth algorithm is a Riemannian manifold version of the Metropolis-adjusted Langevin algorithm. More details are presented in Section 3.3.

2.7.1 Software for optimal control

To implement optimal-control techniques, it is necessary to have access to a numerical ODE solver. An example of a high-quality solver is CVODES, part of the SUNDIALS suite of numerical software (Serban and Hindmarsh 2005; Hindmarsh et al. 2005). Another necessary piece of software is an implementation of an optimisation algorithm. Optimal-control algorithms can have hundreds or even thousands of variables, and a similar number of constraints. Typically, each constraint only involves a few of the variables, making for a sparse optimisation problem. The reasons why optimal-control problems have these characteristics are explained further in Section 3.2. Because of the large number of variables, it is important to have an optimisation implementation that can exploit this sparsity. One example of such a implementation is IpOpt (Wächter 2002; Wächter and Biegler 2006), which can use any of a number of sparse linear solvers.

It is also necessary to be able to compute the gradient and Hessian of the cost function, as well as the Jacobian of the constraint functions. While it is sometimes possible to compute these analytically, it will be error-prone, and a major objective of this work is to provide methods that are easy to use. Another possibility is to use finite differences. However, this method can be inaccurate, and is also very slow for high-dimensional problems. For a n-dimensional problem, evaluating the gradient requires $n + 1$ function evaluations, which can be prohibitively expensive. A better alternative is to use automatic differentiation (AD) (Rall and Corliss 1996; Griewank 2003; Griewank and Walther 2008). This method works by applying the chain rule of calculus to each line of the original program, which can be done automatically

without user input. AD comes in two versions: *forward mode*, where the derivatives are propagated from the input variables to the output, and *reverse mode*, where the derivatives are propagated from the output to the input. For forward-mode AD, the running time scales linearly in the number of inputs, while in reverse-mode AD, the running time scales linearly in the number of outputs. For optimisation, which can have a large number of inputs, but only a single output (the cost function), reverse-mode AD is thus a major improvement over finite differences. When the cost function involves solving a system of ODEs, it is also necessary to compute the gradient of the ODE solution with respect the decision variables. This can be achieved by using either *forward sensitivity analysis*: (Maly and Petzold 1996), where the original system of ODEs is augmented by a new set of ODEs that propagate the derivatives forward in time, or using *adjoint sensitivity analysis* (Cao et al. 2003; Jørgensen 2007), where a new set of *adjoint states* is introduced and integrated backwards in time. The latter method is closely connected to indirect optimal-control algorithms, with the states that are integrated backward in time being the costates. For a derivation of both methods, see Bartlett (2008). Both methods are implemented in CVODES.

The software chosen to accomplish these tasks is CasADi (Andersson 2013). This software allows the user to construct a computational graph, which can be automatically differentiated. CasADi provides integration with CVODES, so that the user-defined expressions can include calls to an ODE solver, which too can be differentiated using the sensitivity analysis capabilities of CVODES. It is also integrated with IpOpt, and can automatically generate the required Hessian and Jacobian matrices. While CasADi is written in C++, it has bindings for Python and MATLAB. The Python interface is the recommended way to use CasADi, and is well-documented (Andersson et al. 2016). For this reason, this project exclusively uses Python as its implementation language.

It can be noted that there are other software packages that provide similar capabilities. Two of the most well-known ones are Theano (Theano Development Team 2016), and TensorFlow (Abadi et al. 2016), which also allow the user to construct computational graphs, which can be automatically differentiated. However, these packages are primarily intended for deep learning applications, and do not provide integration with ODE solvers. This makes CasADi more convenient for the purposes here.

2.7.2 Software for Markov chain Monte Carlo

There is a wide selection of MCMC software available. The most widely known one is the BUGS family, including WinBUGS and OpenBUGS (Spiegelhalter et al. 1999; Lunn et al. 2009), whose sampling algorithms include Gibbs sampling, slice sampling and random walk Metropolis-Hastings. JAGS (Plummer 2003) is an alternative BUGS implementation, using a similar modelling language and similar sampling algorithms. A more recent alternative is Stan (Carpenter et al. 2017), which uses Hamiltonian Monte Carlo (Duane et al. 1987; Neal 2011), and its more automated variant NUTS (Hoffman and Gelman 2014). In all these implementations, models are specified in a custom modelling language.

In addition to this, PyMC version 2 (Patil et al. 2010) and version 3 (Salvatier et al. 2016) provide the ability to write models in Python. While PyMC 2 primarily uses random walk Metropolis-Hastings and slice sampling, PyMC 3 adds Hamiltonian Monte Carlo and NUTS.

Despite the availability of MCMC software, none of the packages mentioned here were deemed suitable for this thesis. A major reason for this is that using them would make it impossible to make a fair comparison between sampling algorithms. Since different packages implement different MCMC algorithms, a comparison between algorithms would in practice be a comparison between implementations. A second major reason is that none of these software packages have support for Riemannian manifold sampling algorithms. While some packages, such as PyMC, make it relatively straightforward to implement custom sampling schemes, these schemes often require higher-order derivatives that cannot easily be provided by the package.

In the light of these considerations, the MCMC algorithms investigated here were all implemented in Python. While Python is not a fast language, the running time is in practice dominated by the time to run the ODE solvers, which are written in C. CasADi was used to generate the gradients and Hessians required by the Riemannian manifold algorithms.

Example code for optimal-control as well as MCMC algorithm can be obtained at `http://www2.warwick.ac.uk/fac/sci/eng/research/biomedical/impact/project_results/project_5/`.

2.8 Summary of the investigated methods

Figure 2.4 shows a summary of all the investigated priors, function parameterisations and inference algorithms that were evaluated in the case studies. All investigated priors and parameterisations have previously been used in pharmacological or physiolo-

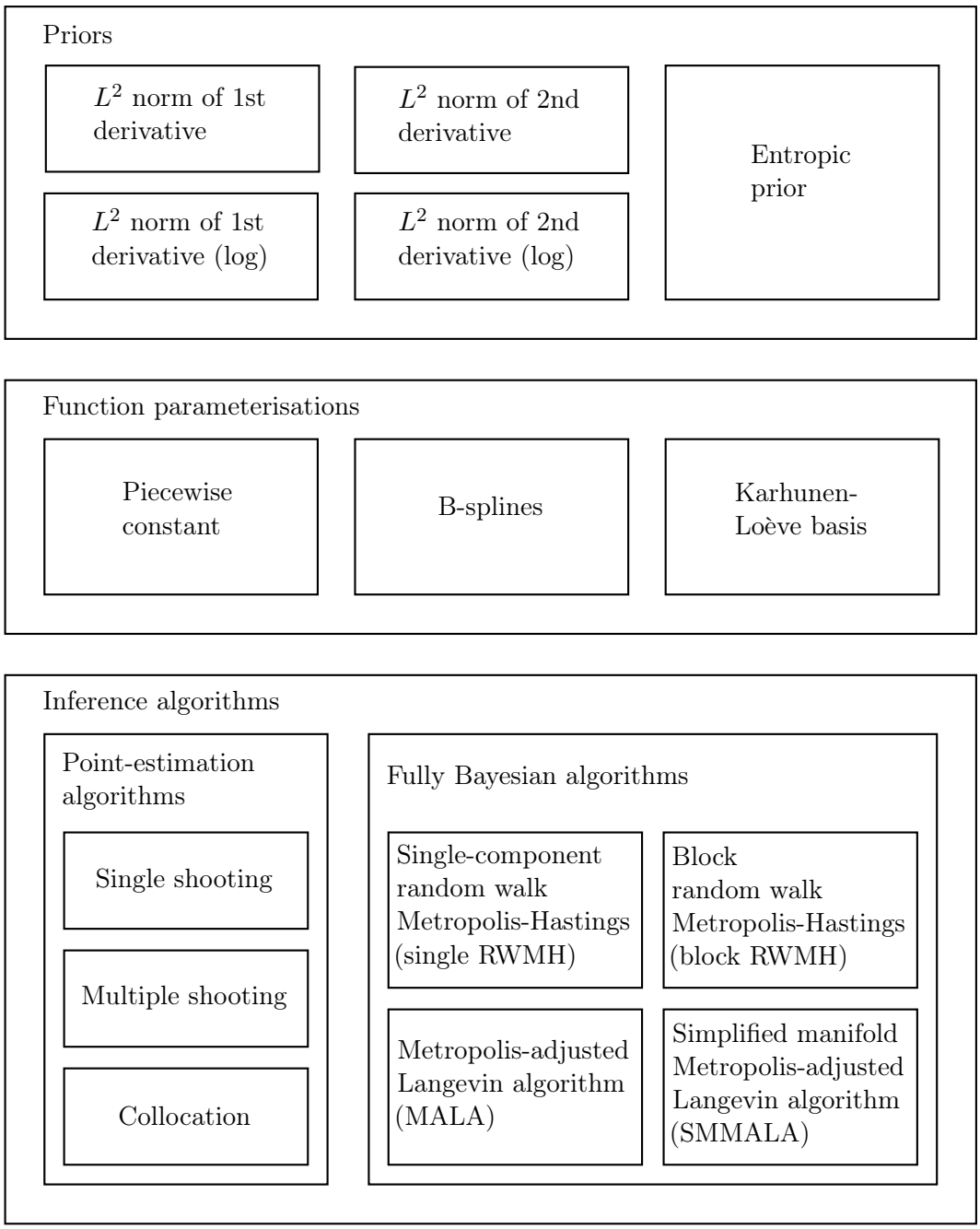

Figure 2.4: A summary of the investigated priors, function parameterisations and inference algorithms.

gical modelling (Verotta 1996; Pillonetto et al. 2002; Hattersley et al. 2008; Bell and Pillonetto 2004). Among the point-estimation inference algorithms, single shooting can be considered to be an obvious, naive algorithm. The more advanced algorithms

are well-known in the optimal-control community (Rao 2009), but have not previously been evaluated for the applications considered here. Among the MCMC methods, random-walk methods have previously been used (Pillonetto et al. 2002; Pillonetto and Bell 2007), while the Langevin and Riemannian manifold methods are novel in this context.

Implementing multiple shooting and collocation is non-trivial, especially if the derivatives have to be worked out by hand. This might be one reason why these algorithms are not commonly used for pharmacological applications. They are also not widely known in the pharmaceutical modelling community. This may partly be explained by noting that single-shooting methods perform well enough in many typical parameter-estimation problems. Nonlinear input-estimation problems can be more computationally difficult, since even seemingly simple problems become high-dimensional when the input function is discretised. New software, like CasADi, makes it easier to implement these kinds of algorithms, and therefore to provide easy-to-use tools that modellers can use.

For MCMC algorithms, the situation is relatively similar. Few modellers are familiar with recent sampling algorithms, such as those based on Riemannian manifolds. Additionally, no commonly used MCMC software implements these algorithms. Again, this suggests that the development of more use-friendly tools could help in making these algorithms become more widely used.

2.9 Choice of case studies

The case studies were chosen based on the following criteria:

- A model for the system must be available. This was deemed necessary, as the research project was not intended to be focused on model building.
- Data must be available, preferably data which have either been previously published, or can realistically be published as part of the project.
- The case study must be interesting and relevant from a drug-discovery perspective.

Additionally, it was deemed important that the case studies were significantly different from each other, in order to test the performance of the input-estimation methods across various conditions. Chapter 4 discusses a typical PK example, where the absorption rate of orally administered eflornithine is estimated from plasma-concentration measurements. Chapter 5 explores estimation of energy intake from

body-weight data of mice as well as humans, where the model, time scales and data have characteristics which differ significantly from typical PK examples. In Chapter 6, another PK example is considered, where the absorption rate after subcutaneous administration of an extended-release formulation of exenatide is estimated from plasma-concentration data. This differs significantly from the case study in Chapter 4, as the formulation results in a complicated release profile, extending over several weeks. Together, these case studies explore a wide range of models and data.

Chapter 3

In-depth theory

This chapter provides more detailed descriptions of the estimation methods that were chosen for this project, as outlined in Section 2.7. While the contents of Chapter 2 should be sufficient to understand the following chapters, the contents here provide a deeper understanding.

3.1 The Karhunen-Loève expansion

In this thesis, the input functions are parameterised either using B-splines or the Karhunen-Loève expansion for MCMC-based estimation. B-splines were briefly explained in Section 2.4.2. While they are conceptually simple, and are useful for representing realistic-looking functions with a small set of basis functions, there are some issues to address.

Firstly, it is conceptually simpler to assign priors to the coefficients of the basis functions rather than to the function itself. This transforms the input-estimation problem into a parameter-estimation problem. However, it is not straightforward to translate a prior over functions, such as penalisation of the jth derivative, into a prior over the B-spline coefficients.

Additionally, when using a low-dimensional function representation, it may be difficult to know whether all plausible input functions can be represented by the chosen basis. If input functions that have significant probability cannot be well approximated by the basis, the basis itself will influence the estimation results, which is to be avoided in non-parametric estimation.

The Karhunen-Loève expansion (Wang 2008) provides a method for constructing a basis that addresses both of these problems. To understand the underlying principles, it is helpful to first consider expansions of deterministic functions, such

as the Fourier series. In this expansion, a function can be represented by a linear combination of basis functions, where the coefficients are numbers. Analogously, a stochastic process can be represented by a linear combination of fixed basis functions, where the coefficients are stochastic variables. Here, we will restrict the presentation to Gaussian processes, where the coefficients will be Gaussian variables. To make the notation simpler, only zero-mean Gaussian processes will be considered.

The goal of the Karhunen-Loève expansion is to construct a set of basis functions, such that any real-valued stochastic process $u(t)$ defined in the interval $[t_i, t_f]$, with mean function $m(t)$ and covariance function $K(s,t)$, can be represented by

$$u(t) = \sum_{j=0}^{\infty} a_j \phi_j(t) \tag{3.1}$$

with coefficients a_j given by

$$a_j = \int_{t_i}^{t_f} u(t)\phi_j(t)\,\mathrm{d}t \tag{3.2}$$

where the basis functions $\phi_j(t)$ and the coefficients a_j satisfy the following properties:

- The basis functions are orthonormal in the L^2 sense:

$$\int_{t_i}^{t_f} \phi_j(t)\phi_k(t)\,\mathrm{d}t = \delta_{jk} \tag{3.3}$$

  where δ_{jk} is the Kronecker delta

$$\delta_{jk} = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

- The basis function coefficients a_j are zero-mean, independent Gaussian stochastic variables.

This exposition will be informal. In particular, infinite sums will be used without considering convergence. For a more rigorous treatment, see Wang (2008) or Levy (2008, Ch. 7).

The basis functions $\phi_j(t)$ are the eigenfunctions of the integral equation

$$\int_{t_i}^{t_f} K(s,t)\phi_j(s)\,\mathrm{d}s = \lambda_j \phi_j(t) \tag{3.5}$$

where the eigenvalues λ_j are the variances of the coefficients a_j. For each j, the solution is unique up to a multiplicative constant. This constant will be chosen so that all of the $\phi_j(t)$ have unit norm.

To prove that the functions $\phi_j(t)$ are pairwise orthogonal, note that

$$\begin{aligned}\lambda_j \int_{t_i}^{t_f} \phi_j(t)\phi_k(t)\,\mathrm{d}t &= \int_{t_i}^{t_f} \left(\int_{t_i}^{t_f} K(s,t)\phi_j(s)\,\mathrm{d}s \right) \phi_k(t)\,\mathrm{d}t \\ &= \int_{t_i}^{t_f} \left(\int_{t_i}^{t_f} K(t,s)\phi_k(t)\,\mathrm{d}t \right) \phi_j(s)\,\mathrm{d}s \\ &= \lambda_k \int_{t_i}^{t_f} \phi_j(t)\phi_k(t)\,\mathrm{d}t\end{aligned} \tag{3.6}$$

implying that

$$(\lambda_j - \lambda_k) \int_{t_i}^{t_f} \phi_j(t)\phi_k(t)\,\mathrm{d}t = 0. \tag{3.7}$$

This shows that the inner product of $\phi_j(t)$ and $\phi_k(t)$, $j \neq k$, is zero as long as the eigenvalues are distinct. The proof for the case of degenerate eigenvalues is omitted here, as no stochastic processes in this work have degenerate eigenvalues. By Mercer's theorem (Mercer 1909), the covariance function can be decomposed as:

$$K(s,t) = \sum_{j=0}^{\infty} \lambda_j \phi_j(s)\phi_j(t). \tag{3.8}$$

Next, it will be shown that a stochastic process generated by Eq. (3.1), with the coefficients a_j drawn independently from zero-mean Gaussian distributions with variance λ_j, will have the desired mean and covariance functions. Let $m_{KL}(t)$ be the mean function induced by this construction, whose value is given by

$$m_{KL}(t) = \mathbb{E}\left[u(t)\right] = \sum_{j=0}^{\infty} \mathbb{E}\left[a_j\right] \phi_j(t) = 0 \tag{3.9}$$

which shows that the mean function, as desired, is 0. Similarly, the covariance function is given by

$$K_{KL}(s,t) = \mathbb{E}\left[u(s)u(t)\right] = \sum_{j=0}^{\infty}\sum_{k=0}^{\infty} \mathbb{E}\left[a_j a_k\right] \phi_j(s)\phi_k(t) = \sum_{j=0}^{\infty} \lambda_j \phi_j(s)\phi_j(t) \tag{3.10}$$

where the last equality follows from $\mathbb{E}\left[a_j a_k\right] = \lambda_j \delta_{jk}$. This is equal to the decompos-

ition in Eq. (3.8). Hence the obtained covariance function is equal to the desired one.

Up to this point, the Karhunen-Loève expansion is not an approximation. Approximations are introduced by ordering the basis functions so that $\lambda_j > \lambda_{j+1}$ for all j, and truncating the infinite series to a finite sum of N_B terms. Equation (3.10) shows that each basis function contributes to the covariance function by a factor proportional to λ_j. Assuming that the λ_j values drop off sufficiently rapidly, the covariance function will be dominated by the first few basis functions. It can be shown that the resulting approximation is optimal in a mean-square sense (Le Maître and Knio 2010, Ch. 2).

In this work, the basis functions $B_j(t)$ are selected as:

$$B_j(t) = \sqrt{\lambda_j}\phi_j(t) \tag{3.11}$$

so that the prior over each basis function coefficient is a Gaussian distribution with unit variance.

3.1.1 Karhunen-Loève expansion for penalisation of the first derivative

A statistical model penalising the first derivative has a discrete-time interpretation as a random-walk model. The continuous-time equivalent of a random walk is the Wiener process $W(t)$, also known as Brownian motion, which is a Gaussian process defined by (Klebaner 2012, Ch. 3):

1. The increments $W(t) - W(s)$, for $t > s$, are independent of the value of the process before time s, that is $p(W(t) - W(s)|W(r)) = p(W(t) - W(s))$, where $0 \leq r \leq s$.

2. The increment $W(t) - W(s)$ is Gaussian distributed with mean 0 and variance $t - s$.

3. The paths $W(t)$ are continuous with probability 1.

To fully characterise the input function model, the process noise precision τ has to be considered. To account for τ, the input function can be defined as $u(t) = W(t)/\sqrt{\tau}$, so $u(t)$ has increments with variance $(t - s)/\tau$.

First, the covariance function has to be determined. Assuming that $s < t$, it becomes

$$
\begin{aligned}
K(s,t) &= \mathbb{E}\left[u(s)u(t)\right] = \mathbb{E}\left[u(s)\cdot((u(t)-u(s))+u(s))\right] \\
&= \mathbb{E}\left[u(s)\cdot(u(t)-u(s))\right] + \mathbb{E}\left[u(s)^2\right] = \mathbb{E}\left[u(s)\right]\mathbb{E}\left[u(t)-u(s)\right] + \mathbb{E}\left[u(s)^2\right] \\
&= \mathbb{E}\left[u(s)^2\right] = \frac{s}{\tau},
\end{aligned}
\tag{3.12}
$$

since $u(s)$ and $u(t)-u(s)$ are independent, and $u(s)$ as well as $u(t)-u(s)$ have zero mean. Similarly, when $s \geq t$, the covariance function is t/τ. In summary, the scaled Wiener process has zero mean and covariance function

$$
K(s,t) = \frac{\min\{s,t\}}{\tau}. \tag{3.13}
$$

The Karhunen-Loève basis functions are obtained by solving the eigenvalue problem

$$
\int_{t_i}^{t_f} \frac{\min\{s,t\}}{\tau}\phi(s)\,\mathrm{d}s = \lambda\phi(t). \tag{3.14}
$$

For simplicity, let us assume that $t_i = 0$. If this it not the case, the functions can be time-shifted. To eliminate the minimum operator, the expression can be rewritten as

$$
\int_0^t \frac{s}{\tau}\phi(s)\,\mathrm{d}s + \frac{t}{\tau}\int_t^{t_f} \phi(s)\,\mathrm{d}s = \lambda\phi(t). \tag{3.15}
$$

This problem can be solved by differentiating both sides to convert the integral equation to a differential equation. Differentiation with respect to t once yields

$$
\frac{1}{\tau}\int_t^{t_f} \phi(s)\,\mathrm{d}s = \lambda\frac{\mathrm{d}\phi(t)}{\mathrm{d}t}. \tag{3.16}
$$

Differentiating a second time yields

$$
-\frac{1}{\tau}\phi(t) = \lambda\frac{\mathrm{d}^2\phi(t)}{\mathrm{d}t^2}. \tag{3.17}
$$

This is a second-order differential equation, whose solution is given by

$$
\phi(t) = A\sin\left(\frac{t}{\sqrt{\lambda\tau}}\right) + B\cos\left(\frac{t}{\sqrt{\lambda\tau}}\right) \tag{3.18}
$$

where A and B are constants. Since the process starts at 0 when $t = 0$, it follows that $B = 0$. A second boundary condition is given by evaluating Eq. (3.16) at $t = t_f$.

The left-hand side is clearly 0, which gives

$$\left.\frac{\mathrm{d}\phi(t)}{\mathrm{d}t}\right|_{t=t_f} = \frac{A}{\sqrt{\lambda\tau}}\cos\left(\frac{t_f}{\sqrt{\lambda\tau}}\right) = 0, \tag{3.19}$$

which is satisfied when

$$\frac{t_f}{\sqrt{\lambda\tau}} = \frac{\pi}{2} + \pi \cdot j, \qquad j \in \mathbb{Z}. \tag{3.20}$$

Hence, we can define the set of eigenvalues by $\lambda_j = \frac{1}{\tau}\left(\frac{t_f}{\pi(\frac{1}{2}+j)}\right)^2$. The constant A is selected to normalise the corresponding eigenfunctions $\phi_j(t)$, which gives

$$\phi_j(t) = \sqrt{\frac{2}{t_f}}\sin\left(t\frac{1}{\sqrt{\lambda_j\tau}}\right) = \sqrt{\frac{2}{t_f}}\sin\left(\frac{t}{t_f}\pi\left(\frac{1}{2}+j\right)\right). \tag{3.21}$$

Note that although the eigenvalues depend on the process noise precision τ, the eigenfunctions do not. This makes it possible to use a linear basis function representation for the input even if τ is allowed to be an unknown parameter.

3.1.2 Karhunen-Loève expansion for penalisation of the second derivative

Penalising the second derivative can be interpreted as modelling the input as the time integral of the scaled Wiener process:

$$u(t) = \int_{t_i}^{t} \frac{1}{\tau} W(r)\,\mathrm{d}r \tag{3.22}$$

where the initial value and its derivative are 0. From here on, it will be assumed that $t_i = 0$ to simplify the expressions. The stochastic process has zero mean, and covariance function

$$K(s,t) = \mathbb{E}\left[u(s)u(t)\right] = \int_0^s \int_0^t \frac{1}{\tau}\mathbb{E}\left[W(v)W(w)\right]\mathrm{d}v\,\mathrm{d}w = \int_0^s \int_0^t \frac{1}{\tau}\min\{v,w\}\,\mathrm{d}v\,\mathrm{d}w. \tag{3.23}$$

In the inner integral, for the case that $s < t$, we have $w < t$ and the integral evaluates to

$$\begin{aligned}\int_0^t \frac{1}{\tau}\min\{v,w\}\,\mathrm{d}v &= \frac{1}{\tau}\int_0^w v\,\mathrm{d}v + \frac{1}{\tau}\int_w^t w\,\mathrm{d}v = \frac{1}{\tau}\frac{v^2}{2}\bigg|_0^w + \frac{1}{\tau}wv\bigg|_w^t \\ &= \frac{1}{\tau}\frac{w^2}{2} + \frac{1}{\tau}w(t-w) = \frac{1}{\tau}\left(wt - \frac{w^2}{2}\right).\end{aligned} \tag{3.24}$$

Evaluating the outer integral yields

$$\frac{1}{\tau}\left(\int_0^s \left(wt - \frac{w^2}{2}\right) \mathrm{d}w)\right) = \frac{1}{\tau}\left(\frac{tw^2}{2}\bigg|_0^s - \frac{w^3}{6}\bigg|_0^s\right) = \frac{s^2}{6\tau}(3t - s). \tag{3.25}$$

By symmetry, when $s \geq t$, the result is $\frac{t^2}{6\tau}(3s - t)$. In summary, the covariance function is

$$K(s,t) = \frac{\min\{s,t\}^2}{6\tau}(3\max\{s,t\} - \min\{s,t\}). \tag{3.26}$$

To obtain the Karhunen-Loève basis functions, the eigenvalue problem

$$\int_0^{t_f} \frac{\min\{s,t\}^2}{6\tau}(3\max\{s,t\} - \min\{s,t\})\phi(s)\,\mathrm{d}s = \lambda\phi(t) \tag{3.27}$$

can be rewritten as

$$\int_0^t \frac{s^2}{6}(3t - s)\phi(s)\,\mathrm{d}s + \int_t^{t_f} \frac{t^2}{6}(3s - t)\phi(s)\,\mathrm{d}s = \lambda \cdot \tau \cdot \phi(t). \tag{3.28}$$

This can be converted to a differential equation by differentiating four times with respect to time, which yields

$$\int_0^t \frac{s^2}{2}\phi(s)\,\mathrm{d}s + \int_t^{t_f} t\left(s - \frac{t}{2}\right)\phi(s)\,\mathrm{d}s = \tau \cdot \lambda\frac{\mathrm{d}\phi(t)}{\mathrm{d}t} \tag{3.29}$$

$$\int_t^{t_f} (s - t)\phi(s)\,\mathrm{d}s = \tau \cdot \lambda\frac{\mathrm{d}^2\phi(t)}{\mathrm{d}t^2} \tag{3.30}$$

$$-\int_t^{t_f} \phi(s)\,\mathrm{d}s = \tau \cdot \lambda\frac{\mathrm{d}^3\phi(t)}{\mathrm{d}t^3} \tag{3.31}$$

$$\phi(t) = \tau \cdot \lambda\frac{\mathrm{d}^4\phi(t)}{\mathrm{d}t^4}. \tag{3.32}$$

This is a fourth-order differential equation, which has the solution

$$\phi(t) = c_1 \cos\left(\frac{\alpha}{t_f}t\right) + c_2 \sin\left(\frac{\alpha}{t_f}t\right) + c_3 e^{-\frac{\alpha}{t_f}(t_f - t)} + c_4 e^{-\frac{\alpha}{t_f}t} \tag{3.33}$$

where $\alpha = t_f \frac{1}{\sqrt[4]{\tau \cdot \lambda}}$. $c_1, \ldots c_4$ are integration constants that are determined by the boundary conditions, and the requirement that the eigenfunctions are normalised. According to the definition of the integrated Wiener process, $\phi(0) = 0$ and $\frac{\mathrm{d}\phi(t)}{\mathrm{d}t}\Big|_{t=0} = 0$. It follows from Eqs. (3.30) and (3.31) that $\frac{\mathrm{d}^2\phi(t)}{\mathrm{d}t^2}\Big|_{t=t_f} = \frac{\mathrm{d}^3\phi(t)}{\mathrm{d}t^3}\Big|_{t=t_f} = 0$. The

resulting system of equations has a non-trivial solution only when α satisfies

$$\frac{1}{\cosh(\alpha)} - \cos(\alpha) = 0. \tag{3.34}$$

The set of eigenvalues λ_j can be obtained by solving this equation numerically, and arranging the solutions in decreasing order. The coefficients $c_1, \ldots, c_3$ can be expressed as functions of c_4, obtained from the boundary conditions. The value of c_4 can then be selected to make the eigenfunction have unit norm, and its value can be substituted in the expressions for $c_1, \ldots, c_3$. This gives

$$c_4 = \left(\int_0^{t_f} \left[\left(-1 + \frac{\cos(\alpha) + e^{-\alpha}}{\sin(\alpha)} \right) \cos(\alpha t/t_f) + \left(1 + \frac{\cos(\alpha) + e^{-\alpha}}{\sin(\alpha)} \right) \sin(\alpha t/t_f) \right. \right. \tag{3.35}$$

$$\left. \left. + \frac{\frac{2}{1+e^{-2\alpha}} - 1}{\sin(\alpha)} e^{-\alpha(t_f - t)/t_f} + e^{-\alpha t/t_f} \right]^2 \mathrm{d}t \right)^{-1/2} \tag{3.36}$$

$$c_3 = c_4 \frac{\frac{2}{1+e^{-2\alpha}} - 1}{\sin(\alpha)} \tag{3.37}$$

$$c_2 = c_4 - c_3 e^{-\alpha} \tag{3.38}$$

$$c_1 = -c_4 - c_3 e^{-\alpha}. \tag{3.39}$$

Similar to penalisation of the first derivative, these functions are independent of the regularisation parameter τ, so this can be used as a fixed basis function model even when τ is a variable parameter.

3.2 Direct methods of optimal control

In optimal control-based methods, the input-estimation problem is treated as an optimisation problem. There are several ways to formulate such optimisation problems, and the most straightforward ways are not necessarily the most efficient ones. Counterintuitively, increasing the number of decision variables and constraints can often make the problem easier to solve. Here, three methods are presented. They are applicable to indirect as well as direct methods, although they are presented here in the context of direct methods. It will be assumed that there are no fixed parameters, such as unknown initial conditions or regularisation parameters. For an overview of these methods, see Rao (2009). More detail is provided by Betts (2010) and Biegler (2010).

All direct methods start by discretising the input function, so that it can be represented by its basis function coefficients, $\mathbf{a} = \begin{bmatrix} a_0, & \ldots, & a_{N_B-1} \end{bmatrix}^T$, where N_B is the number of basis functions. Optimal-control methods can accommodate high-dimensional input parameterisations, so simple basis functions, such as piecewise constant functions, can be used. The resulting optimisation problem is to find the maximum of the log-posterior, $\log p(\mathbf{a}|\mathbf{y})$. Since the marginal likelihood is difficult to compute, in practice the unnormalised log-posterior is used, $\log p(\mathbf{a}) + \log p(\mathbf{y}|\mathbf{a})$. The state trajectory $\mathbf{x}(t)$ must be known in order to evaluate the log-likelihood. To make this dependence on $\mathbf{x}(t)$ explicit, and to convert the maximisation to a minimisation problem, we introduce the notation for the cost function $J(\mathbf{a}, \mathbf{x}(t)) = -\log p(\mathbf{x}) - \log p(\mathbf{y}|\mathbf{a})$. Here, the dependence on the measurements $\mathbf{y}$ is suppressed, as these are constant for a given problem. Since $\mathbf{x}(t)$ is completely determined by the input signal, it is possible to treat the cost function as a function only of $\mathbf{a}$. However, retaining $\mathbf{x}(t)$ as part of the optimisation problem can have advantages, as discussed in Sections 3.2.2 and 3.2.3.

As an aside, the input signal is usually denoted by $\mathbf{u}$ in the optimal-control literature. In the discrete-time version, optimisation is performed with respect to the basis function coefficients a_i rather than with respect to the actual function $\mathbf{u}(t)$. Therefore, to avoid confusion, optimisation is performed with respect to $\mathbf{a}$ here. In a slight abuse of notation, the dynamical equation will be written $\mathbf{f}(t, \mathbf{x}(t), \mathbf{a})$, which should be interpreted as $\mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$, where $\mathbf{u}(t)$ is the input function generated by the coefficients $\mathbf{a}$.

Thus, a solution to the optimisation problem requires selecting values for $\mathbf{a}$ as well as a state trajectory $\mathbf{x}(t)$ such that the cost function is minimised, with the constraint that the dynamical equation $\mathbf{f}(\cdot)$ must be satisfied for all times. The optimisation problem can also include additional equality or inequality constraints in both the input and the states. For the work here, these constraints will usually be nonnegativity constraints of the input function. The dynamical models will usually ensure that states are nonnegative as long as the inputs and initial conditions are. In

summary, the problem can be posed as

$$\begin{aligned}
\underset{\mathbf{a},\mathbf{x}(t)}{\text{minimise}} \quad & J(\mathbf{a},\mathbf{x}(t)) && \text{(3.40a)}\\
\text{s.t.} \quad & \frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{f}(t,\mathbf{x}(t),\mathbf{a}), \quad \forall t \in [t_i, t_f] && \text{(3.40b)}\\
& \mathbf{x}(t_i) = \mathbf{x}^{(0)} && \text{(3.40c)}\\
& \mathbf{c}(\mathbf{a},\mathbf{x}(t)) = \mathbf{0} && \text{(3.40d)}\\
& \mathbf{h}(\mathbf{a},\mathbf{x}(t)) \leq \mathbf{0} && \text{(3.40e)}
\end{aligned}$$

where $\mathbf{c}(\cdot)$ and $\mathbf{h}(\cdot)$ are the, possibly vector-valued, equality and inequality constraints.

3.2.1 Single shooting

An obvious way to eliminate the dynamical constraints in Eq. (3.40b) is to use the fact that the state trajectory is completely determined by the input signal. This can be formalised by introducing the transition function

$$\Phi(t_0, t_1, \mathbf{x}_{t_0}, \mathbf{a}) = \mathbf{x}(t_0) + \int_{t_0}^{t_1} \mathbf{f}(t,\mathbf{x}(t),\mathbf{a})\,\mathrm{d}t, \tag{3.41}$$

which is the function that gives the state at time t_1, given the input function coefficients, $\mathbf{a}$, and the state at t_0, $\mathbf{x}_{t_0}$. The transition function can be computed by using a numerical ODE solver, such as CVODE (Hindmarsh et al. 2005). The resulting optimisation problem has only $\mathbf{a}$ as its decision variables, and can be formulated as

$$\begin{aligned}
\underset{\mathbf{a}}{\text{minimise}} \quad & J(\mathbf{a},\mathbf{x}(t)) && \text{(3.42a)}\\
\text{s.t.} \quad & \mathbf{x}(t_i) = \mathbf{x}^{(0)} && \text{(3.42b)}\\
& \mathbf{c}(\mathbf{a},\mathbf{x}(t)) = \mathbf{0} && \text{(3.42c)}\\
& \mathbf{h}(\mathbf{a},\mathbf{x}(t)) \leq \mathbf{0} && \text{(3.42d)}\\
& \text{where } \mathbf{x}(t) = \Phi(t_i, t, \mathbf{x}^{(0)}, \mathbf{a}).
\end{aligned}$$

This method is called *single shooting*, as the state trajectory is determined by a single run of the ODE solver from t_i to t_f. Single shooting is the most straightforward method for solving optimal-control problems, with the optimisation algorithm treating the dynamical model as a black box. However, more efficient methods can be obtained by allowing closer interaction between the optimisation algorithm and the dynamical

model. This will be explored in the next section.

3.2.2 Multiple shooting

One disadvantage of single shooting is that the relationship between the decision variables and the cost function can be highly nonlinear. Many optimisation methods for these kinds of problems are based on making second-order approximations of the cost function (Nocedal and Wright 2006), and therefore work most efficiently when the cost function is close to quadratic. As an example, suppose that the system is linear and that the measurements are linear functions of the state variables, with Gaussian measurement noise. In this case, a linear relationship between decision variables and states would result in a quadratic cost function.

As the cost function can depend on the state trajectory up to the final time t_f, it is useful to consider the behaviour of the transition function $\Phi(t_i, t_f, \mathbf{x}^{(0)}, \mathbf{a})$. Any nonlinearities in the dynamics will accumulate as the system evolves in time, making $\Phi(\cdot)$, and therefore the cost function, potentially highly nonlinear for large t_f.

Multiple shooting is a method for making the optimisation problem less nonlinear by only integrating the system over short time intervals. For sufficiently short intervals, the function $\Phi(\cdot)$ is approximately linear, which can be seen by making a Taylor series expansion of $\Phi(\cdot)$ and choosing a time interval short enough to make the linear term dominate over higher-order terms. In multiple shooting, the time interval $[t_i, t_f]$ is divided into n intervals, with endpoints $t_i = t_0 < t_1 < \ldots < t_n = t_f$, and each interval is integrated separately. The initial states of each interval are used as decision variables together with the input function coefficients $\mathbf{a}$. Letting $\mathbf{x}_k$ denote the state vector at the start of the kth time interval, the problem definition is as follows:

$$
\begin{aligned}
\underset{\mathbf{a},\mathbf{x}_0,\ldots,\mathbf{x}_n}{\text{minimise}} \quad & J(\mathbf{a}, \mathbf{x}(t)) && \text{(3.43a)}\\
\text{s.t.} \quad & \mathbf{x}_0 = \mathbf{x}^{(0)} && \text{(3.43b)}\\
& \mathbf{x}_{k+1} = \Phi(t_k, t_{k+1}, \mathbf{x}_k, \mathbf{a}), \quad \text{for } k = \{0, \ldots, n-1\} && \text{(3.43c)}\\
& \mathbf{c}(\mathbf{a}, \mathbf{x}(t)) = \mathbf{0} && \text{(3.43d)}\\
& \mathbf{h}(\mathbf{a}, \mathbf{x}(t)) \leq \mathbf{0} && \text{(3.43e)}\\
\text{where } & \mathbf{x}(t) = \Phi(t_k, t, \mathbf{x}_k, \mathbf{a}), \quad k = \max\{k \mid t_k < t\}.
\end{aligned}
$$

The continuity constraints in Eq. (3.43c), capture the requirement that the state trajectories must be continuous, so the states at the end of interval k must be

equal to the states at the beginning of interval $k+1$. Compared to single shooting, this formulation requires $n \cdot d_x$ decision variables in addition to $\mathbf{a}$, resulting in an optimisation problem with a larger number of variables, and also a larger number of constraints. On the other hand, it results in a less nonlinear problem. Furthermore, the optimisation problem is relatively sparse in the sense that only a few decision variables are present in each continuity constraint. Modern optimisation software such as IpOpt (Wächter 2002; Wächter and Biegler 2006) can efficiently solve sparse problems. Another advantage is that the state trajectory can be initialised with an appropriate initial guess, if one is available. The multiple shooting approach is a special case of a general technique known as *lifting*, where intermediate variables are exposed to the optimiser, for the purpose of transforming a small nonlinear problem to a larger less nonlinear one (Albersmeyer and Diehl 2010).

3.2.3 Collocation

Collocation methods can be seen as a way to allow even further interaction between the optimisation algorithm and the dynamical system model. Instead of using an ODE solver, the dynamical model is introduced as constraints in the optimisation problem. First, the interval $[t_i, t_f]$ is divided into n intervals, $t_i = t_0 < t_1 < \ldots < t_n = t_f$. Within each interval, the state trajectory is approximated by a polynomial of order d, where d is typically a small number, in the range 1 to 3. The $d+1$ coefficients of each of the d_x states in each of the n intervals are included as decision variables, together with the input coefficients $\mathbf{a}$. As the state coefficients are uniquely determined by $\mathbf{a}$, the problem must include $(d+1) \cdot d_x \cdot n$ extra equality constraints.

To see how these constraints can be introduced, consider the time interval $[t_k, t_{k+1}]$. To make the notation simpler, we introduce a new time variable, $s = (t - t_k)/h_k$, where h_k is the interval length $t_{k+1} - t_k$. This way, s runs from 0 to 1. Next, $d+1$ points, $s_0, \ldots, s_d$, called *collocation points*, are selected in the interval $[0, 1]$. The locations of these points are chosen to minimise numerical errors. Common choices are the *Legendre* and *Radau* points (Biegler 2010). It is convenient to represent the state trajectories using *Lagrange polynomials*, defined by

$$L_r(s) = \prod_{v=0, v \neq r}^{d} \frac{s - s_v}{s_r - s_v}. \tag{3.44}$$

The state trajectory in this interval is given by

$$\mathbf{x}(s) = \sum_{r=0}^{d} \mathbf{x}_{k,r} L_r(s) \tag{3.45}$$

where $\mathbf{x}_{k,r}$ is the state vector at the rth collocation point in the kth interval. The Lagrange polynomials are interpolants: since $L_r(s_m) = \delta_{rm}$, the expression in Eq. (3.45) will evaluate to the coefficient $\mathbf{x}_{k,r}$ at time s_r. In this way, the decision variables are themselves the state values at the collocation points.

One set of constraints is given by noting that the state trajectories need to be continuous, so that the value at the end of interval k is equal to the value at the start of interval $k+1$. This can be formulated as

$$\sum_{r=0}^{d} \mathbf{x}_{k,r} L_r(1) = \sum_{r=0}^{d} \mathbf{x}_{k+1,r} L_r(0). \tag{3.46}$$

This still leaves d constraints per state and interval. These are chosen by requiring that the dynamical equations $\mathbf{f}(\cdot)$ are satisfied at d collocation points per interval. Since the state trajectories are polynomials, their derivatives with respect to time can be computed analytically as

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \sum_{r=0}^{d} \mathbf{x}_{k,r} \frac{\mathrm{d}L_r(s)}{\mathrm{d}s} \cdot \frac{\mathrm{d}s}{\mathrm{d}t} = \frac{1}{h_k} \sum_{r=0}^{d} \mathbf{x}_{k,r} \cdot \frac{\mathrm{d}L_r(s)}{\mathrm{d}s}. \tag{3.47}$$

The collocation equations can now be written as

$$\mathbf{f}(t_{k,m}, \mathbf{x}_{k,m}, \mathbf{a}) = \frac{1}{h_k} \sum_{r=0}^{d} C_{r,m} \cdot \mathbf{x}_{k,r} \tag{3.48}$$

for all k, m, where $C_{r,m}$ is the time derivative of $L_r(s)$ evaluated at time s_m. In summary, the collocation formulation of the optimal control problem can be written

as

$$\underset{\mathbf{a},\mathbf{x}_{0,0},\ldots,\mathbf{x}_{n-1,d}}{\text{minimise}} \quad J(\mathbf{a},\mathbf{x}(t)) \tag{3.49a}$$

$$\text{s.t.} \quad \mathbf{x}_{0,0} = \mathbf{x}^{(0)} \tag{3.49b}$$

$$\sum_{r=0}^{d} \mathbf{x}_{k,r} L_r(1) - \sum_{r=0}^{d} \mathbf{x}_{k+1,r} L_r(0) = 0, \tag{3.49c}$$

$$\text{for } k = \{0,\ldots,n-1\}$$

$$\mathbf{f}(t_{k,m},\mathbf{x}(t_{k,m}),\mathbf{a}) - \frac{1}{h_k}\sum_{r=0}^{d} C_{r,m}\cdot\mathbf{x}_{k,r} = 0 \tag{3.49d}$$

$$\text{for } k = \{0,\ldots,n-1\},\ m = \{1,\ldots,d\}.$$

$$\mathbf{c}(\mathbf{a},\mathbf{x}_{n-1,d}) = \mathbf{0} \tag{3.49e}$$

$$\mathbf{h}(\mathbf{a},\mathbf{x}_{n-1,d}) \leq \mathbf{0} \tag{3.49f}$$

The decision variables in this formulation are the coefficients of the input function $\mathbf{a}$ together with the value of the trajectory of each state at every collocation point. Compared to single shooting, this increases the number of decision variables and constraints by $n\cdot(d+1)\cdot d_x$. The advantages of collocation over other methods is that while the resulting optimisation problem has a large number of decision variables, it tends to result in problems that are even less nonlinear than in multiple shooting. The large number of decision variables is not necessarily a problem, since the resulting problem tends to be sparse: each constraint function contains only a small subset of the variables. When the cost function has a likelihood term that depends on the states at the measurement times, this term will only depend on $\mathbf{a}$ and the state variables for the collocation intervals for which measurements exist. When the input function is parameterised using piecewise constant basis functions aligned with the collocation intervals, additional sparsity is achieved, which can be exploited by the optimisation algorithm (Andersson 2013).

3.3 Fundamentals of Markov chain Monte Carlo

As mentioned in Section 2.2.2, solving Bayesian models is a computationally difficult problem, due to the difficulty of computing high-dimensional integrals. Evaluating the posterior distribution involves integrating over all parameters to obtain the marginal likelihood. Additionally, computing expectations such as means and variances involves

computing integrals. Monte Carlo methods are based on the idea that, if it is possible to draw samples from a distribution $p(\boldsymbol{\theta})$, one can approximate the expectation of any function $h(\boldsymbol{\theta})$ of interest, namely:

$$\mathbb{E}_p\left[h(\boldsymbol{\theta})\right] = \int_{\Theta} h(\boldsymbol{\theta}) p(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \tag{3.50}$$

by a sample average $\bar{h}_N$ of N samples drawn from $p(\boldsymbol{\theta})$:

$$\bar{h}_N = \frac{1}{N} \sum_{i=0}^{N-1} h(\boldsymbol{\theta}^{(i)}), \qquad \boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}). \tag{3.51}$$

By choosing an appropriate function $h(\boldsymbol{\theta})$, any quantities of interest such as means, standard deviations and quantiles can be computed. The validity of the Monte Carlo method is guaranteed by the strong law of large numbers (SLLN) (Rosenthal 2006), stating that for a sequence of independent random variables $h(\boldsymbol{\theta})$ with expectation $\mathbb{E}_p\left[h(\boldsymbol{\theta})\right]$, the following holds:

$$\bar{h}_N \to \mathbb{E}_p\left[h(\boldsymbol{\theta})\right] \qquad \text{as } N \to \infty, \tag{3.52}$$

where convergence is with probability 1. This result shows that as the number of samples tends to infinity, the sample average will converge to the true expectation. In practice, it is important to assess the accuracy of a finite number of samples. This can be assessed with the central limit theorem (CLT) (Robert and Casella 2004), which states that for a sequence of independent identically distributed random variables $h(\boldsymbol{\theta})$ with expectation $\mathbb{E}_p\left[h(\boldsymbol{\theta})\right]$ and variance σ^2, the following holds:

$$\sqrt{N} \frac{\left(\bar{h}_N - \mathbb{E}_p\left[h\left(\boldsymbol{\theta}\right)\right]\right)}{\sigma} \to \mathcal{N}(0,1) \qquad \text{as } N \to \infty, \tag{3.53}$$

where convergence is in distribution. In other words, the difference between the true expectation and the sample average tends to a Gaussian distribution with standard deviation $\sigma/\sqrt{N}$. This scaling by $\sqrt{N}$ means that, to increase the accuracy by one decimal place, the required number of samples increases 100-fold. This makes Monte Carlo methods unsuitable for problems where high accuracy is desired. On the other hand, the CLT is independent of the dimensionality of the parameter space, in contrast to deterministic numerical techniques.

Monte Carlo methods presuppose that a method exists for drawing random numbers from the distribution of interest. For drawing samples from the uniform distribution in the interval $[0, 1]$, efficient pseudo-number generators exist, such as the Mersenne Twister (Matsumoto and Nishimura 1998), which is the default random

number generator in several computing environments including MATLAB (MATLAB 2016), R (R Core Team 2016) and NumPy (Van Der Walt et al. 2011). From uniform random numbers, it is possible to obtain samples from many standard distributions using techniques such as the generalised inverse transformation, and rejection sampling (Robert and Casella 2004).

When the distribution is the posterior of a complicated Bayesian model, no standard method for drawing independent samples exists. The idea behind MCMC methods is to construct a stochastic process that generates samples that are not necessarily independent, but which are still drawn from the distribution of interest, called the *target distribution*. They can therefore be used in Eq. (3.51) to estimate expectations. One intuitive explanation for why MCMC makes sampling easier is that the generation of new samples can be guided by previous samples. If previous samples come from an area of high density, other points in the same area are likely to have a high density too. Additionally, it turns out to be relatively straightforward to construct such processes.

A Markov chain is a sequence of random variables $\boldsymbol{\theta}^{(i)}$ with the property that future values of the chain depend only on the most recent value:

$$p\left(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)}, \boldsymbol{\theta}^{(i-2)}, \ldots, \boldsymbol{\theta}^{(0)}\right) = p\left(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)}\right). \tag{3.54}$$

This means that a Markov chain can be completely specified by an initial distribution and a transition distribution κ, called a *transition kernel*, so that the distribution of a sequence is given by

$$p\left(\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N-1)}\right) = p(\boldsymbol{\theta}^{(0)}) \prod_{i=1}^{N-1} \kappa\left(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)}\right). \tag{3.55}$$

A realisation of this chain can be generated by drawing a sample from the initial distribution, and then repeatedly generating the next sample by drawing from the conditional distributions given by κ. Note that κ could in principle depend on the time index i, but here only time-independent, or *homogeneous*, transition kernels will be considered. Assume that there exists a distribution π satisfying

$$\pi(\boldsymbol{\theta}^{(i)} = y) = \int_{\Theta} \kappa(\boldsymbol{\theta}^{(i)} = y|\boldsymbol{\theta}^{(i-1)} = x)\pi(\boldsymbol{\theta}^{(i-1)} = x)\,\mathrm{d}x. \tag{3.56}$$

This is called the *stationary* or *invariant* distribution. If the marginal distribution at any time index is π, then any subsequent draws will also come from π. It appears plausible that a sequence of samples generated this way can be used in Eq. (3.51) to estimate expectations with respect to π. For this to work, the following

conditions have to be met:

1. The stationary distribution π must exist.

2. The chain has to converge to the stationary distribution, from an arbitrary starting point. This is necessary since, in general, it is not possible to initialise the chain from the stationary distribution — if that were possible, there would be no need to use MCMC in the first place.

3. The sample averages have to converge to the true expectation, so that some version of the SLLN and CLT holds. This can fail if the chain exhibits certain pathologies. For example, a transition kernel that does not move would trivially satisfy Eq. (3.56) for any π, but the resulting sequence would not be useful for estimation.

4. It has to be possible to construct a transition kernel such that π is equal to the desired target distribution.

In Section 3.3.1, points 2–3 will be explored. In Section 3.3.2, a method for constructing a kernel that gives the desired stationary distribution is presented. The resulting chain will have a stationary distribution by construction, so point 1 is automatically satisfied.

3.3.1 Properties of Markov chains

While all estimation problems in this thesis consider continuous parameter spaces, the concepts of Markov chain theory are simpler for discrete spaces. For this reason, definitions are given first for discrete spaces, followed by their continuous counterparts. For continuous parameter spaces, the probability of visiting any single value is 0. To be able to treat these chains similarly to discrete Markov chains, probabilities of single values have to be replaced by probabilities of measurable sets. For the purposes of this thesis, it suffices to note that a measurable set is a set that can be assigned a probability. For readers who desire a measure-theoretic treatment of probability, Rosenthal (2006) is recommended. A more thorough treatment of discrete spaces can be found in Roberts (1996). For comprehensive treatments of general spaces, see Robert and Casella (2004), Tierney (1994), Tierney (1996), Roberts and Rosenthal (2004) and Meyn and Tweedie (2009). In the definitions below, the parameter space is denoted by Θ, while the set of all measurable subsets of Θ is denoted $\mathcal{B}(\Theta)$.

A chain that is useful for MCMC should satisfy the following properties:

Irreducibility For discrete chains, let τ_y be the time index of the first time state y is visited. A Markov chain is said to be *irreducible* if

$$P(\tau_y < \infty | \boldsymbol{\theta}^{(0)} = x) > 0, \qquad \forall x, y \in \Theta. \tag{3.57}$$

This means that it is possible to move from any parameter value to any other in a finite number of steps. If this property does not hold, only parts of the parameter space would be visited, with the initial value determining which parts. Such a chain cannot have a unique stationary distribution that is independent of the starting value.

A Markov chain on continuous spaces is said to be φ-irreducible for a probability distribution φ if

$$P(\tau_A < \infty | \boldsymbol{\theta}^{(0)} = x) > 0, \qquad \forall x \in \Theta, A \in \mathcal{B}(\Theta) \tag{3.58}$$

for any measurable set A such that $\varphi(A) > 0$, where τ_A is the first time that a state in A is visited. As in the discrete case, this means that the probability of moving from x to a point in A in a finite number of steps is greater than zero. This definition is less strict than irreducibility, as it allows sets of zero probability to be unreachable. For the purposes of MCMC, we require that $\varphi = \pi$, so that all sets that are assigned positive probability by the stationary distribution can be reached from any starting state.

Aperiodicity A discrete Markov chain is said to aperiodic if the following holds:

$$\text{greatest common divisor} \left\{ i > 0 : \kappa^i(\boldsymbol{\theta}^{(i)} = x | \boldsymbol{\theta}^{(0)} = x) > 0 \right\} = 1, \qquad \forall x \in \Theta \tag{3.59}$$

where κ^i is the i-step transition kernel, κ applied i times. If this condition does not hold, the chain will be split into subsets that are visited in a cyclic fashion, and the chain cannot have a stationary distribution.

In the continuous case, a Markov chain is periodic if it can be divided into disjoint sets such that each set is visited in a cyclic fashion. This definition can be made more rigorous by introducing the concept of small sets, something that is beyond the scope of this thesis. For more information, see Robert and Casella (2004).

Recurrence Let η_x be the number of times the Markov chain visits the state

x, and let τ_{xx} be the number of steps for the chain to return to state x, $\tau_{xx} = \min\left\{i : \boldsymbol{\theta}^{(i)} = x | \boldsymbol{\theta}^{(0)} = x\right\}$. A Markov chain is said to be *recurrent* if

$$\mathbb{E}\left[\eta_x\right] = \infty, \qquad \forall x \in \Theta. \tag{3.60}$$

The chain is *positive recurrent* if additionally

$$\mathbb{E}\left[\tau_{xx}\right] < \infty, \qquad \forall x \in \Theta. \tag{3.61}$$

A chain that is not recurrent is called *transient*. If the chain is transient, there are states that are only visited finitely many times, so their proportion will be zero as the number of samples tends to infinity.

For continuous spaces, define η_A as the number of times the chain visits the set A. A chain is recurrent if it is φ-irreducible for some φ, and

$$\mathbb{E}\left[\eta_A\right] = \infty, \qquad \forall A \in \mathcal{B}(\Theta) : \varphi(A) > 0. \tag{3.62}$$

Often, a stronger form of recurrence is required, known as *Harris recurrence*. A chain is Harris recurrent if it is recurrent, and additionally

$$P(\eta_A = \infty) = 1, \qquad \forall A \in \mathcal{B}(\Theta) : \varphi(A) > 0. \tag{3.63}$$

Harris recurrence guarantees that any realisation will visit each set infinitely often, with probability 1. Positive recurrence on the other hand only implies that each set will *on average* be visited infinitely often. For a particular realisation of the chain however, this may not hold, if the chain was started at an unfortunate parameter value.

A φ-irreducible, aperiodic chain with π as its stationary distribution is positive recurrent (Roberts and Rosenthal 2006). It can be shown that for a chain with π as its stationary distribution, if it is π-irreducible, aperiodic and Harris recurrent, then the following holds:

$$\|\kappa^i(\cdot|\boldsymbol{\theta}^{(0)} = x) - \pi\| \to 0, \qquad \forall x \in \Theta \tag{3.64}$$

where $\|\cdot\|$ denotes *total variation distance* (Tierney 1994). This means that the chain converges to π independent of the starting state. While a proof will not be presented here, the following intuitive argument can be made for discrete chains: suppose we

start a chain C_1 from any state x. Start another chain C_2, initialised by drawing from the stationary distribution. Since C_2 is Harris recurrent, it will reach state x in a finite number of steps. Next, remove all samples from C_2 before the first time x is visited. Since the removed part has finite length, the statistics of the resulting chain will not be affected in the limit. But this modified chain starts in the same state as C_1, and these chains necessarily have the same statistics due to the Markov property. A more detailed discussion can be found in Meyn and Tweedie (2009). For continuous chains, more care has to be taken, but the basic idea remains the same. Such a chain also obeys the SLLN.

Assuming that some mild technical conditions hold (Tierney 1996), which here will always be assumed to be the case, there is a variant of the CLT that holds for π-irreducible, aperiodic, Harris-recurrent chains:

$$\sqrt{N}\frac{\bar{h}_N - \mathbb{E}_\pi\left[h\left(\boldsymbol{\theta}\right)\right]}{\sigma} \to \mathcal{N}(0,1) \qquad \text{as } N \to \infty \tag{3.65}$$

where now σ^2 is given by:

$$\sigma^2 = \operatorname{Var}\left[h\left(\boldsymbol{\theta}^{(i)}\right)\right] + 2\sum_{k=1}^{\infty}\operatorname{Cov}\left[h\left(\boldsymbol{\theta}^{(i)}\right), h\left(\boldsymbol{\theta}^{(i+k)}\right)\right]. \tag{3.66}$$

This result shows that strong correlations between samples will result in high variance. As expected, this expression reduces to the CLT (3.53) when the samples are uncorrelated. When constructing Markov chains, it is therefore important to strive to minimise correlations between samples.

In practice, Markov chains for MCMC are often constructed to satisfy the *detailed balance* condition (Geyer 2011), namely:

$$\pi\left(y\right)\kappa\left(x|y\right) = \pi\left(x\right)\kappa\left(y|x\right), \qquad \forall x, y \in \Theta. \tag{3.67}$$

This states that the probability of being in state x and moving to y is the same as the probability of being in state y and moving to x. A chain satisfying this condition is called *reversible*. Detailed balance is a special case of the stationarity condition, which can be seen by integrating both sides with respect to x:

$$\pi(y)\int_\Theta \kappa(x|y)\,\mathrm{d}x = \int_\Theta \pi(x)\kappa(y|x)\,\mathrm{d}x \iff \pi(y) = \int_\Theta \pi(x)\kappa(y|x)\,\mathrm{d}x, \tag{3.68}$$

since $\kappa(x|y)$ is a probability distribution and therefore integrates to 1 over Θ. While detailed balance is just a sufficient and not a necessary condition for a chain to have

π as its stationary distribution, it is often used in practice.

For more information on the general aspects of MCMC, see Gilks et al. (1996), Brooks et al. (2011), Chib and Greenberg (1995), Geyer (1992) and Kass et al. (1998).

3.3.2 The Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm (Metropolis et al. 1953; Hastings 1970) is a general method for constructing Markov chains with any user-defined stationary distribution $\pi(\boldsymbol{\theta})$. To simplify the presentation, here it is assumed that all distributions can be represented by a probability density. However, the methodology is applicable to general distributions. To use the Metropolis-Hastings algorithm, a *proposal distribution* $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$ has to be defined. This distribution generates new parameter values $\boldsymbol{\theta}'$, and is allowed to depend on the current parameters θ. The proposal distribution does not need to be related to the target distribution. At each time step i of the algorithm, a new sample is generated by:

1. Propose a new sample $\boldsymbol{\theta}'$ using the proposal distribution $q\left(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(i-1)}\right)$.

2. Calculate the Metropolis-Hastings ratio A:

$$A(\boldsymbol{\theta}', \boldsymbol{\theta}^{(i-1)}) = \min\left\{1, \frac{\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}^{(i-1)})q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(i-1)})}\right\}. \tag{3.69}$$

3. With probability A, set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}'$. Otherwise, set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$. The former case is referred to as the proposed sample $\boldsymbol{\theta}'$ being *accepted*, while the latter case is referred to as $\boldsymbol{\theta}'$ being *rejected*.

This will produce a Markov chain, whose transition distribution $\kappa\left(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)}\right)$ is the combined action of the proposal distribution and the accept-reject step. This transition will have $\pi(\boldsymbol{\theta})$ as its stationary distribution, which can be shown as follows:

First, consider the case where $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$. Here, the detailed balance equation is trivially satisfied.

Second, consider the case where $\boldsymbol{\theta}^{(i)} \neq \boldsymbol{\theta}^{(i-1)}$. The probability density of making that transition is given by

$$\kappa(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)}) = q(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)}) \cdot A(\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(i-1)}), \qquad \boldsymbol{\theta}^{(i)} \neq \boldsymbol{\theta}^{(i-1)}. \tag{3.70}$$

For detailed balance to hold, the following condition must be met:

$$\pi(\boldsymbol{\theta}^{(i-1)})\cdot q(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)})\cdot A(\boldsymbol{\theta}^{(i)},\boldsymbol{\theta}^{(i-1)}) = \pi(\boldsymbol{\theta}^{(i)})\cdot q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^{(i)})\cdot A(\boldsymbol{\theta}^{(i-1)},\boldsymbol{\theta}^{(i)}), \quad (3.71)$$

or equivalently:

$$\frac{A(\boldsymbol{\theta}^{(i)},\boldsymbol{\theta}^{(i-1)})}{A(\boldsymbol{\theta}^{(i-1)},\boldsymbol{\theta}^{(i)})} = \frac{\pi(\boldsymbol{\theta}^{(i)})\cdot q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^{(i)})}{\pi(\boldsymbol{\theta}^{(i-1)})\cdot q(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)})}. \quad (3.72)$$

This equality holds when A is defined as in Eq. (3.69). There are two cases to consider here. The first case is when $\pi\left(\boldsymbol{\theta}^{(i)}\right)\cdot q\left(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^{(i)}\right) \geq \pi\left(\boldsymbol{\theta}^{(i-1)}\right)\cdot q\left(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)}\right)$, which leads to

$$A(\boldsymbol{\theta}^{(i)},\boldsymbol{\theta}^{(i-1)}) = 1 \quad (3.73)$$

$$A(\boldsymbol{\theta}^{(i-1)},\boldsymbol{\theta}^{(i)}) = \frac{\pi(\boldsymbol{\theta}^{(i-1)})\cdot q(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)})}{\pi(\boldsymbol{\theta}^{(i)})\cdot q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^{(i)})}. \quad (3.74)$$

The second case is when $\pi\left(\boldsymbol{\theta}^{(i)}\right)\cdot q\left(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^{(i)}\right) < \pi\left(\boldsymbol{\theta}^{(i-1)}\right)\cdot q\left(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)}\right)$, which leads to

$$A(\boldsymbol{\theta}^{(i)},\boldsymbol{\theta}^{(i-1)}) = \frac{\pi(\boldsymbol{\theta}^{(i)})\cdot q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^{(i)})}{\pi(\boldsymbol{\theta}^{(i-1)})\cdot q(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)})} \quad (3.75)$$

$$A(\boldsymbol{\theta}^{(i-1)},\boldsymbol{\theta}^{(i)}) = 1. \quad (3.76)$$

In both cases, Eq. (3.72) is satisfied, confirming that the Markov chain has π as its stationary distribution.

The Metropolis-Hastings ratio is well-defined as long as the denominator in Eq. (3.69) is not zero. To ensure that it is positive, it is enough to start the chain at a point with positive density. The other factor in the denominator, $q(\boldsymbol{\theta'}|\boldsymbol{\theta})$, will always be positive, since a point of zero density would not be proposed. Any proposal to move to a point of zero density will be rejected by the accept-reject step.

In the special case when $q(\boldsymbol{\theta'}|\boldsymbol{\theta}) = q(\boldsymbol{\theta}|\boldsymbol{\theta'})$, the Metropolis-Hastings ratio simplifies to $A = \min\{1, \pi(\boldsymbol{\theta'})/\pi(\boldsymbol{\theta^{(i-1)}})\}$. Here, a proposal is always accepted if the proposed value has a higher density that the current value. This is the original algorithm presented by Metropolis et al. (1953), and is referred to as the Metropolis algorithm.

In Bayesian statistics, the invariant distribution of interest is the posterior, $p(\boldsymbol{\theta}|\mathbf{y})$. As described in Section 2.2.2, the posterior is given by Eq. (2.8), where

the numerator is generally easy to evaluate, while the denominator, the *marginal likelihood*, is often intractable. However, the marginal likelihood does not depend on the parameters, and will therefore be cancelled in the Metropolis-Hastings ratio. Therefore, it is sufficient to be able to evaluate the prior and the likelihood in order to use the algorithm.

For the Markov chain to be useful, it is also necessary to check that the chosen proposal distribution results in a chain that is irreducible, aperiodic and Harris recurrent.

All MCMC methods presented later in this section are special cases of the Metropolis-Hastings algorithm, differing only in the choice of proposal distributions. It can also be useful to create *mixtures* and *cycles* of proposal mechanisms. In a *mixture*, at each step of the algorithm, one out of several proposal distributions is chosen at random. In a *cycle*, each proposal is applied in turn. It can be shown that a mixture is irreducible and aperiodic if at least one of its component distributions is irreducible and aperiodic. For cycles, if each transition preserves the invariant distribution, so does the cycle (Geyer 2011). One way to introduce cycles is to divide the parameter vector into blocks of parameters, and update each block separately, possibly with different proposal mechanisms. In these kinds of cycles, the individual transition distributions do preserve the invariant distribution, but they are not irreducible. The user of the method has to ensure that the combination of all updates results in an irreducible chain.

3.3.3 Practical considerations

So far, it has been shown that, under certain technical conditions, the estimates from the Markov chain will converge to the true values in the limit, as the number of samples tend to infinity. In practical applications, the chain can only have a finite length, and it is important to assess whether the generated samples are a good approximation of the true distribution. There are two things to consider:

1. The chain might be initialised far from the target distribution. For this reason, there is a possibility that the initial part of the chain is unrepresentative of the distribution, even though it is known that the chain will converge to the target distribution in the limit (Fig. 3.1). Discarding this initial part of the chain can increase the accuracy of the estimates, a practice known as *burn-in*. In order to determine the number of samples to discard, it is necessary to know when the chain has converged.

2. Even when all samples can be assumed to be drawn from the correct distribution,

it is still necessary to assess the number of samples needed to estimate the expectations of interest with a desired level of accuracy. Since the samples in general are correlated, it is necessary to generate more samples than would be needed if the samples were independent.

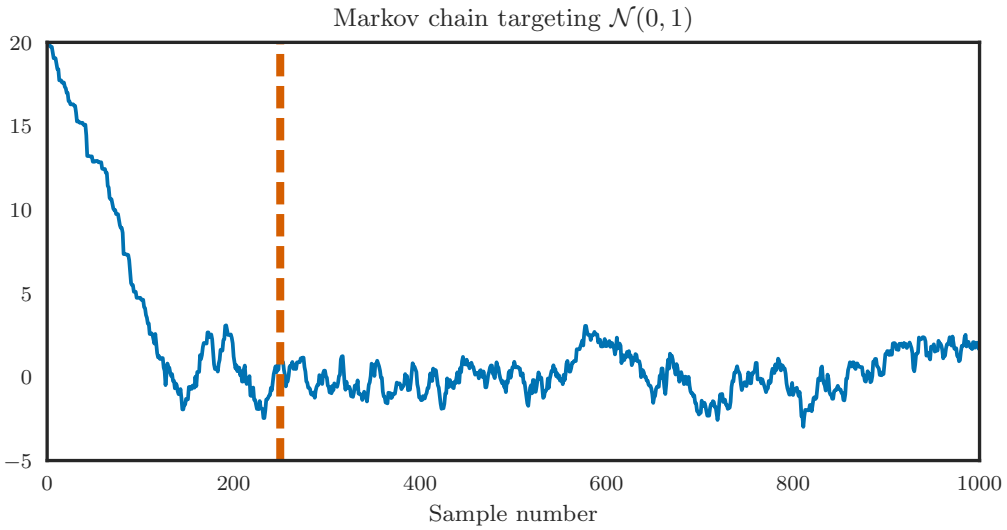

Figure 3.1: A Markov chain targeting the distribution $\mathcal{N}(0, 1)$ using the Metropolis-Hastings algorithm with proposal distribution $q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(i-1)}) = \mathcal{N}(\boldsymbol{\theta}^{(i-1)}, 0.4^2)$. Since the chain is initialised at $\boldsymbol{\theta}^{(0)} = 20$, the initial part of the chain, before the dashed line, is far from the desired distribution, and accuracy can be increased by removing it.

For general statistical models, no theoretical results exist that can guarantee that a finite set of samples from a Markov chain will result in accurate estimates (Geyer 2011). Instead, the samples generated by the chain can be analysed by heuristic methods, that rely on statistical properties of the samples, in order to detect possible problems. One straightforward method to assess convergence is to plot the samples and visually determine the number of samples to remove. In Fig. 3.1, it can be seen that excluding the initial 200–250 samples is probably enough.

One method that can be used to determine the amount of burn-in required, as well as the number of samples required, is the *Raftery-Lewis* diagnostic (Raftery and Lewis 1992; Raftery and Lewis 1995). Here, it is assumed that the user wants to estimate a quantile of a parameter, or a function of a parameter, with a given accuracy. The user provides the method with a set of samples from the Markov chain, together with four parameters: the quantile q, the desired margin of error r, the probability of attaining the desired margin of error s, and the convergence tolerance

ϵ. The interpretation of this is that a value H is sought such that $P(h(\boldsymbol{\theta}) < H) = q$, with the requirement that the estimated H falls in the interval $(q - r, q + r)$ with probability s. The role of ϵ is to give the threshold for where the initial state can be considered to have an insignificant influence on the chain.

The key idea of the Raftery-Lewis method is that, in order to determine a quantile, it is only necessary to consider whether a sample falls above or below that quantile. Therefore, it is sufficient to analyse a two-state stochastic process, where the states are either 0 or 1, indicating whether the sample is above or below q. While analysing a general Markov chain can be difficult, efficient methods exist for analysing two-state Markov chains. As a first step, the Raftery-Lewis method generates a new two-state stochastic process by computing $Z^{(i)} = \mathbb{I}(h(\boldsymbol{\theta}^{(i)}) < H_e)$, where $\mathbb{I}$ is the indicator function and H_e is an estimate of H from the samples. While this process is not Markovian, it is possible to create an approximately Markovian process by *thinning*, keeping only every kth sample. A suitable value of k is obtained by model-comparison techniques based on the Bayesian information criterion (BIC). There are two useful analytical results for a two-state Markov chain with transition probabilities $\kappa(1|0) = \alpha$ and $\kappa(0|1) = \beta$, where the first can be used to determine the amount of burn-in, and the second can be used to determine the number of samples required after the burn-in phase:

- The contribution of the initial state on the probabilities of the chain is less than ϵ after m steps, where

$$m = \frac{\log\left(\frac{\epsilon(\alpha+\beta)}{\max(\alpha,\beta)}\right)}{\log(1-\alpha-\beta)}. \tag{3.77}$$

  Since the analysis is done on a thinned chain, this result shows that $m \cdot k$ samples should be discarded as burn-in.

- When the quantile is estimated by $\hat{q} = 1/n \sum_{i=0}^{n-1} Z^{(i)}$, the estimate can be shown to have mean q and variance $\frac{1}{n}\frac{\alpha\beta(2-\alpha-\beta)}{(\alpha+\beta)^3}$. By setting $P(q-r \leq \bar{Z}_t^{(k)}) = s$, the number of thinned samples n needed is

$$n = \frac{\frac{\alpha\beta(2-\alpha-\beta)}{(\alpha+\beta)^3}}{\left(\frac{r}{\Phi^{-1}\left(\frac{1}{2}(1+s)\right)}\right)^2} \tag{3.78}$$

  where $\Phi(\cdot)$ is the standard normal cumulative distribution function. The number of samples required after burn-in is given by $n \cdot k$.

Another method to determine the number of samples required is to use the effective sample size (ESS), a measure of the quality of the samples generated by the chain. Recall that to obtain a given accuracy, more samples are needed from a Markov chain than would be required from a collection of independent samples, due to correlations in the chain. ESS is defined by

$$ESS = \frac{N}{1 + 2\sum_{k=1}^{\infty} \gamma_k \left[h\left(\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(i+k)}\right)\right]} \tag{3.79}$$

where the γ_k are the k-lag correlation coefficients. These coefficients can be estimated from the samples produced by the chain using methods such as the *initial monotone sequence estimator* (Geyer 1992). From Eq. (3.79), it can be seen that the variance of an estimate of $h(\cdot)$ based on N samples from the Markov chain is equal to the variance of an estimate based on ESS independently generated samples.

It is important to note that these methods are not able to detect all problems. While a failed convergence test is a sign of problems, a successful test does not necessarily imply a lack of problems. An example of this is *pseudo-convergence*, which can occur if the target distribution has several modes, separated by areas of low probability density. In this situation, the chain may only find one of the modes, and the generated samples could well pass the diagnostics presented above.

An alternative to burn-in is to initialise the chain at a point of high probability density, so that it can be assumed to be close to stationarity from the beginning. This can be achieved by finding a local maximum of the density using an optimisation method, such as any of the optimal-control methods presented in Section 3.2.

In all case studies, the quality of the generated samples was assessed using the Raftery-Lewis diagnostic and the ESS. Burn-in was generally avoided by initialising the Markov chain using optimal control-based estimation.

In summary, a great strength of MCMC methods is that they can, in principle, be applied to any kind of statistical model. The parameters can have arbitrary prior distributions, and can depend on each other in arbitrary, possibly nonlinear, ways. Any statistic of interest can be computed to arbitrary precision given enough computing time. However, there are challenges with these methods as well. As discussed above, it can be difficult to assess whether the samples produced by the chain are representative of the target distribution. Additionally, these methods can be computationally intensive. To obtain a sufficiently large effective number of samples in a reasonable amount of computation time, it is necessary to find good proposal distributions. This is a non-trivial task, as the performance of algorithms using these distributions is highly problem-dependent. Below, a number of methods for

generating proposals is presented.

3.3.4 Random walk Metropolis-Hastings

A common method for generating proposals is to use a standard distribution such as a Gaussian, centred on the current parameter value. For continuous parameters, updated simultaneously by a multivariate Gaussian, the resulting chain is irreducible and aperiodic. This can be seen by noting that for any parameter value, it is possible to reach any other parameter value in a single step. Alternatively, the parameters can be updated one at a time using a univariate proposal. More generally, the parameters can be divided into blocks that are updated separately.

The covariance matrix of the proposal distribution has a large impact on the correlation between samples and therefore the effective sample size. There are two sources of correlation between samples. Firstly, samples can be correlated because the proposal distribution proposes parameter values that are close to the previous values. Secondly, they can be correlated because the proposal is rejected, resulting in the new value being identical to the previous one. Figure 3.2 shows this for a one-dimensional distribution. If the variance is chosen to be small, most proposals will be accepted, and the chain will exhibit a slow random walk that explores the target only very slowly. If the variance on the other hand is large, proposals will be in regions of low target density, resulting in most proposals being rejected. In the first case, correlation is high because the proposal distribution generates new sample values close to the previous values. In the second case, correlation is high because most proposals are rejected, resulting in runs of identical values. It has been shown that for multivariate Gaussian proposals, optimal performance is achieved when the proposal variance is set to obtain an acceptance rate of 0.234 (Roberts and Rosenthal 2001).

One method to find a good proposal distribution is to monitor the acceptance rate, and adjust the proposal variance accordingly to keep it close to the optimal value. This is done during the burn-in phase, after which the proposal distribution is fixed. Adjusting the proposal distributions while samples are being recorded is in general not valid, since the resulting chain is not a Markov chain and does not necessarily satisfy detailed balance. However, there do exist adaptive methods that have the correct distribution even though the resulting chain is not Markovian (Haario et al. 2001).

In multiple dimensions, finding a suitable proposal distribution can be difficult even with adaptive methods, since different parameters may need different proposal variances. Additionally, any correlation between parameters would require the

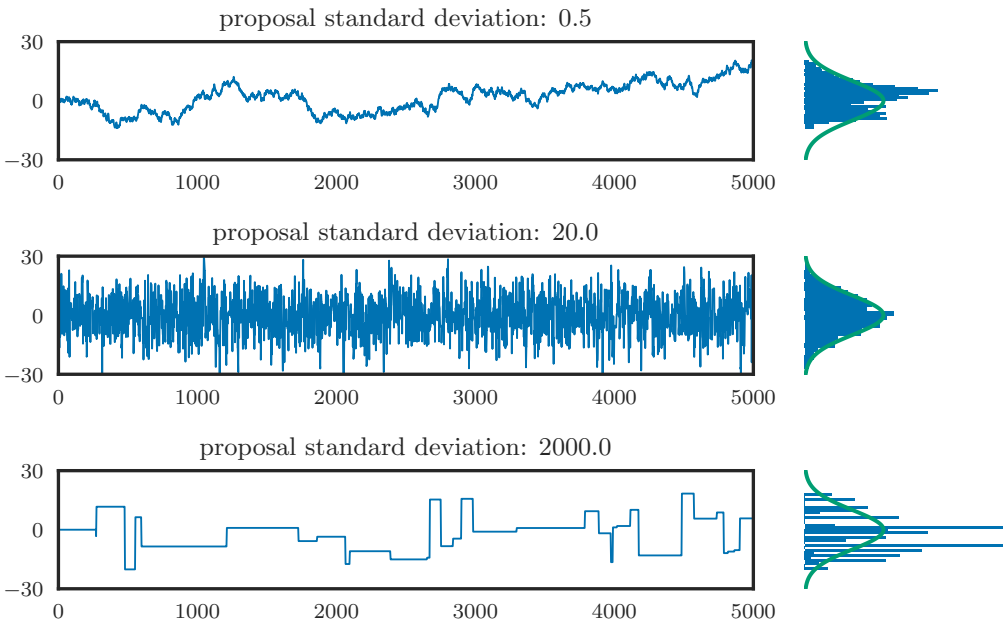

Figure 3.2: Samples drawn from a zero-mean Gaussian with standard deviation 10, using random walk Metropolis-Hastings with different proposal standard deviations. To the right of the plots, histograms of the samples as well as the true density are shown. Top: Using a standard deviation of 0.5 results in a chain with high correlation, since each proposed new value is close to the previous value. Middle: A standard deviation of 20 results in a chain with low correlation between samples. After 5000 samples, the empirical distribution is close to the target. Bottom: Using a standard deviation of 2000 results in a chain with high correlation, since most proposals are rejected. When an unsuitable proposal distribution is used, the empirical distribution differs significantly from the target even after 5000 samples.

proposal to have a similar level of correlation for good performance. For good efficiency, it may also be necessary to use different proposals in different parts of the parameter space. These issues are illustrated in Fig. 3.3.

3.3.5 Gibbs sampling

In Gibbs sampling (Geman and Geman 1984), the parameter vector is split into blocks that are updated one at a time. Each block is updated by drawing samples from its distribution, conditioned on all other parameters. Such a block could contain a single parameter or multiple parameters. For this approach to work, it must be possible to draw samples from the conditional distribution. This is possible if these have a standard form, such as a Gaussian or Gamma distribution. One way to ensure

that this is the case is to use *conjugate priors*. A conjugate prior is a prior such that its corresponding posterior has the same form as the prior (MacKay 2003, Ch. 42). If the prior has a standard form, the posterior will also have a standard form and thus be amenable to sampling. An important example for our purposes is a collection of n independent zero-mean Gaussian variables, $\theta_0, \dots \theta_{n-1}$, which have a common but unknown precision τ. If τ is assigned a Gamma prior, $p(\tau) = \text{Gam}(\alpha, \beta)$, its posterior can be computed by

$$\begin{aligned} p(\tau|\theta_0, \dots, \theta_{n-1}) \propto p(\tau)p(\theta_0, \dots, \theta_{n-1}|\tau) &= \frac{\beta}{\Gamma(\alpha)}\tau^{\alpha-1}e^{-\beta^\alpha\tau} \cdot \prod_{i=0}^{n-1}\sqrt{\frac{\tau}{2\pi}}e^{-\frac{\tau\theta_i^2}{2}} \\ &= \frac{\beta^\alpha}{\Gamma(\alpha)} \cdot \frac{\tau^{\alpha+n/2-1}}{(2\pi)^{n/2}}e^{-\tau(\beta+\sum_{i=0}^{n-1}\frac{\theta_i^2}{2})}. \end{aligned} \tag{3.80}$$

where $\Gamma(\alpha)$ is the Gamma function, not to be confused with the Gamma distribution.

The normalised posterior has to integrate to 1, which can be achieved by recognising the result as an unnormalised Gamma distribution, and assigning the normalisation constant accordingly. In summary the posterior is a $\text{Gam}(\alpha + n/2, \beta + \sum_{i=0}^{n-1}\theta_i^2/2)$ distribution.

An intuitive explanation for why Gibbs sampling preserves the stationary distribution is as follows: let $\boldsymbol{\theta}_j$ denote the jth block of parameters, while $\boldsymbol{\theta}_{-j}$ denotes all other parameters. For the jth block, suppose that the parameters $\boldsymbol{\theta}_{-j}$ are drawn from the target distribution $p(\boldsymbol{\theta}_{-j})$. If $\boldsymbol{\theta}_j$ is drawn from the target distribution conditioned on all other parameters, $p(\boldsymbol{\theta}_j|\boldsymbol{\theta}_{-j})$, obviously the joint distribution $p(\boldsymbol{\theta}_j, \boldsymbol{\theta}_{-j})$ will have the desired distribution. It is also easy to see that Gibbs sampling is a special case of the Metropolis-Hastings algorithm, where the proposal distribution is $q\left(\boldsymbol{\theta}'_j|\boldsymbol{\theta}^{(i-1)}\right) = p\left(\boldsymbol{\theta}'_j|\boldsymbol{\theta}^{(i-1)}_{-j}\right)$. The Metropolis ratio can be computed by using

$$\frac{p\left(\boldsymbol{\theta}'_j, \boldsymbol{\theta}^{(i-1)}_{-j}\right)p\left(\boldsymbol{\theta}^{(i-1)}_j|\boldsymbol{\theta}^{(i-1)}_{-j}\right)}{p\left(\boldsymbol{\theta}^{(i-1)}_j, \boldsymbol{\theta}^{(i-1)}_{-j}\right)p\left(\boldsymbol{\theta}'_j|\boldsymbol{\theta}^{(i-1)}_{-j}\right)} = \frac{p\left(\boldsymbol{\theta}'_j|\boldsymbol{\theta}^{(i-1)}_{-j}\right)p\left(\boldsymbol{\theta}^{(i-1)}_{-j}\right)p\left(\boldsymbol{\theta}^{(i-1)}_j|\boldsymbol{\theta}^{(i-1)}_{-j}\right)}{p\left(\boldsymbol{\theta}^{(i-1)}_j|\boldsymbol{\theta}^{(i-1)}_{-j}\right)p\left(\boldsymbol{\theta}^{(i-1)}_{-j}\right)p\left(\boldsymbol{\theta}'_j|\boldsymbol{\theta}^{(i-1)}_{-j}\right)} = 1. \tag{3.81}$$

This shows that by using the conditional distribution as a proposal distribution, the proposal will always be accepted.

When using Gibbs sampling, it is important to analyse whether the resulting chain is ergodic. It is easy to construct examples where this scheme can result in a

reducible chain, given an unfortunate choice of parameterisation.

3.3.6 Metropolis-adjusted Langevin algorithm

Seeing that it can be difficult to find good proposal distributions for RWMH by hand, it is of great interest to find algorithms that can find good proposals automatically. One way to do this is to exploit additional local information, such as the gradient of the log-target distribution. First, we replace the discrete-time Markov chain $\boldsymbol{\theta}^{(i)}$ by the continuous-time Langevin diffusion $\boldsymbol{\theta}(t)$, which is defined as the solution to the stochastic differential equation (SDE)

$$\mathrm{d}\boldsymbol{\theta}(t) = \frac{1}{2}\frac{\mathrm{d}\log\pi(\boldsymbol{\theta}(t))}{\mathrm{d}\boldsymbol{\theta}}\,\mathrm{d}t + \mathrm{d}\mathbf{W}(t) \tag{3.82}$$

where $\mathbf{W}(t)$ is a Wiener process with the same dimensionality as $\boldsymbol{\theta}$. This is a special case of an *Itô diffusion* (Klebaner 2012). It can be shown that the Langevin diffusion process has $\pi(\boldsymbol{\theta})$ as its stationary distribution, and also that the process will converge to $\pi(\boldsymbol{\theta})$ for any starting value (Roberts and Tweedie 1996; Roberts and Stramer 2002). Thus, samples from $\pi(\boldsymbol{\theta})$ can be obtained by simulating this system, and recording its state at selected intervals.

Equation (3.82) does not in general have a closed-form solution. To solve it, numerical methods can be used. A simple discrete-time approximation is given by the Euler-Maruyama method (Kloeden and Platen 1999), where each time step is given by

$$\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)} + \frac{\epsilon^2}{2}\frac{\mathrm{d}\log\pi(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(i-1)}} + \epsilon\mathbf{z}^{(i-1)} \tag{3.83}$$

where ϵ is a step size, and $\mathbf{z}$ is a vector of independent standard Gaussian variables. This method is similar to Euler's method for ODEs, with an added random term. The time-discretised chain does not necessarily have the correct stationary distribution. To correct this, a Metropolis-Hastings accept-reject step can be added. This results in the Metropolis-adjusted Langevin algorithm (MALA), which is a Metropolis-Hastings algorithm using Eq. (3.83) as its proposal distribution.

Similarly to RWMH, MALA generates proposals by adding a random perturbation to the current parameter value, but it additionally adds a term that moves the parameter along the gradient of the log-target distribution. This will make the proposal prefer regions that are more likely to have a high density. For MALA, the optimal acceptance rate is 0.574 (Roberts and Rosenthal 2001), a figure that can be achieved by adapting the step size ϵ. Since MALA uses additional information in the form of the gradient when generating proposals, these are more likely to have a high

target density. In contrast, RWMH proposals are unrelated to the target, and the only information about the target is obtained from the accept-reject step.

While MALA can help to generate better proposals, it still suffers from the same kind of issues as RWMH. When parameters have different scales, or are strongly correlated, the isotropic random perturbation in Eq. (3.83) will not explore the parameter space efficiently. Additionally, computing the gradient can be expensive. While MALA can increase the quality of the proposals, it will produce a smaller number of samples per unit of computing time. Thus, it is not obvious which method will perform better for a given problem.

One way to improve the performance of MALA is to perform a linear transformation of the parameters such that the transformed parameters are approximately uncorrelated and have similar length scales. Equivalently, the proposal distribution can be modified to take the correlation structure into account. This will result in the proposal

$$\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)} + \frac{\epsilon^2}{2}\mathbf{M}\frac{\mathrm{d}\log\pi(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(i-1)}} + \epsilon\mathbf{M}^{1/2}\mathbf{z}^{(i-1)} \tag{3.84}$$

where $\mathbf{M}$ is a *preconditioning matrix* performing the parameter transformation (Girolami and Calderhead 2011). This obviously causes another issue: how to choose a suitable $\mathbf{M}$.

3.3.7 Riemannian manifold sampling-based methods

The problem with using isotropic proposals like in MALA is that a small change in one parameter might result in a large change in the statistical model, while a change in another parameter might only have a small effect on the model. This is illustrated in Fig. 3.3. It is clear in the upper figure that when moving along the diagonal, the parameter values can change by a large amount without significantly changing the probability density. In contrast, even relatively small movements in other directions will result in a low probability density. In this case, it is desirable to find proposals that prefer to make large steps along the diagonal, while avoiding large steps in other directions. In general, movements should be larger in directions for which the model has low sensitivity. One solution to this is to redefine the notion of distance between points, so that points that result in similar probability densities are considered to be close even if their parameter values are not. This can be accomplished by treating the parameter space as a *Riemannian manifold*. Riemannian manifolds are used in the field of differential geometry to represent curved spaces. While the treatment of this subject is necessarily informal here, a rigorous treatment is given in Boothby (2003) and Carmo (1992). Perhaps the most familiar example of a curved space is

the surface of the Earth. Positions on the Earth are often parameterised by latitude and longitude. The distance required to travel in order to change longitude by 1° depends on the distance to the equator. Hence, naively using the Euclidean metric to compute distances would be incorrect. Differential geometry provides the necessary tools for correctly defining distances between points, independent of the choice of parameterisation. In statistical models, the same mathematics can be used to *define* a notion of distance between points that has desirable properties. The length of a curve $\boldsymbol{\theta}(t) = \Big[\theta_1(t), \quad \theta_2(t), \quad \ldots, \quad \theta_{d-1}(t)\Big]$, parameterised by $t \in [0, 1]$ is given by computing the line integral

$$L = \int_0^1 \left| \frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} \right| \mathrm{d}t \tag{3.85}$$

where $\left|\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t}\right|$ is the magnitude of the tangent vector of the curve at point t. For a Euclidean space, the magnitude is given by $\sqrt{\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t}^T \cdot \frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t}}$, which is the square root of the inner product between the tangent vector with itself. On a general Riemannian manifold, the inner product is defined by a *metric tensor*. This can be conveniently represented by a matrix $\mathbf{G}(\boldsymbol{\theta}(t))$, such that the magnitude of the tangent vector is given by $\sqrt{\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t}^T \mathbf{G}(\boldsymbol{\theta}(t)) \frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t}}$. The length of a curve is now given by

$$L = \int_0^1 \sqrt{\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t}^T \mathbf{G}(\boldsymbol{\theta}(t)) \frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t}} \, \mathrm{d}t. \tag{3.86}$$

The distance between two points can be defined as the length of the *geodesic*, the shortest path between them. In this way, the metric tensor induces a *metric*, a notion of distance between points. For the example of the Earth, the metric tensor is chosen so that the distances defined in this way correspond to real-world physical distances.

A parameter space in a statistical model can also be equipped with a metric tensor. The idea of treating a statistical model as a Riemannian manifold is the subject of the field of information geometry (Amari and Nagaoka 2000). Figure 3.4 shows two examples: the surface of the earth, and a statistical model where each point in the space represents a Gaussian distribution. For small standard deviations, a change of the mean results in a larger change of the statistical model than is the case for larger standard deviations. This can be represented by choosing a suitable metric tensor. Note that, in general, the shortest path between points is not the path that appears straight in the chosen coordinate system.

A Langevin equation defined on a Riemannian manifold is the solution to the stochastic differential equation (SDE)

$$\mathrm{d}\boldsymbol{\theta}(t) = \frac{1}{2}G^{-1}(\boldsymbol{\theta}(t))\frac{\mathrm{d}\log\pi(\boldsymbol{\theta}(t))}{\mathrm{d}\boldsymbol{\theta}}\,\mathrm{d}t + \mathrm{d}\tilde{\mathbf{W}}(t) \tag{3.87}$$

where each component of $\tilde{\boldsymbol{W}}(t)$, $\tilde{W}_k(t)$, is given by:

$$\mathrm{d}\tilde{W}_k(t) = |\mathbf{G}(\boldsymbol{\theta}(t))|^{-\frac{1}{2}}\sum_{j=0}^{d-1}\frac{\partial}{\partial\theta_j}\left(\mathbf{G}^{-1}(\boldsymbol{\theta}(t))_{kj}\,|G(\boldsymbol{\theta}(t))|^{\frac{1}{2}}\right)\mathrm{d}t + \left(\sqrt{\mathbf{G}^{-1}(\boldsymbol{\theta}(t))}\,\mathrm{d}\mathbf{W}(t)\right)_k. \tag{3.88}$$

For a discussion on this expression, see Calderhead 2011, Ch. 3. It can be shown (Calderhead 2011; Girolami and Calderhead 2011) that a discrete-time approximation of this SDE is given by

$$\begin{aligned}\theta_k^{(i)} &= \theta_k^{(i-1)} + \frac{\epsilon^2}{2}\left(\mathbf{G}^{-1}(\boldsymbol{\theta}^{(i-1)})\cdot\left.\frac{\mathrm{d}\log\pi(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(i-1)}}\right)_k \\ &\quad - \epsilon^2\sum_{j=0}^{d-1}\left(\mathbf{G}^{-1}(\boldsymbol{\theta}^{(i-1)})\cdot\left.\frac{\partial\mathbf{G}(\boldsymbol{\theta})}{\partial\theta_j}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(i-1)}}\cdot\mathbf{G}^{-1}(\boldsymbol{\theta}^{(i-1)})\right)_{kj} \\ &\quad + \frac{\epsilon^2}{2}\sum_{j=0}^{d-1}\left(\mathbf{G}^{-1}(\boldsymbol{\theta}^{(i-1)})_{kj}\cdot\mathrm{tr}\left(\mathbf{G}^{-1}(\boldsymbol{\theta}^{(i-1)})\cdot\left.\frac{\partial\mathbf{G}(\boldsymbol{\theta})}{\partial\theta_j}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(i-1)}}\right)\right)_k \\ &\quad + \left(\epsilon\sqrt{\mathbf{G}^{-1}(\boldsymbol{\theta}^{(i-1)})}\mathbf{z}^{(i-1)}\right)_k\end{aligned} \tag{3.89}$$

where ϵ is the step size and the $\mathbf{z}^{(i)}$ are independent standard Gaussian variables. As for MALA, the discrete-time equations do not in general have the correct stationary distribution. Instead, each step is used as a Metropolis-Hastings proposal. The resulting algorithm is called the manifold Metropolis-adjusted Langevin algorithm (MMALA).

A simpler proposal mechanism can be obtained by ignoring the terms that include derivatives of the metric tensor. This results in the simplified manifold Metropolis-adjusted Langevin algorithm (SMMALA), with a proposal defined by

$$\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)} + \frac{1}{2}\epsilon^2\mathbf{G}^{-1}(\boldsymbol{\theta}^{(n-1)})\left.\frac{\mathrm{d}\log\pi(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(i-1)}} + \epsilon\sqrt{\mathbf{G}^{-1}(\boldsymbol{\theta}^{(i-1)}))}\mathbf{z}^{(i-1)}. \tag{3.90}$$

Since the SMMALA is just an approximation of the full MMALA, it can be expected to make poorer proposals. On the other hand, computing derivatives of the metric tensor can be difficult or computationally expensive.

Equation (3.90) can be interpreted as a preconditioned MALA proposal, using $\mathbf{G}^{-1}(\boldsymbol{\theta}^{(i-1)})$ as a preconditioning matrix. In contrast to Eq. (3.84), the preconditioning matrix is here allowed to be position-dependent. SMMALA can therefore be seen as one way to choose the $\mathbf{M}$ matrix.

There is no single correct metric tensor for a given model — any valid metric tensor defines a valid Riemannian manifold. For Bayesian models, it has been suggested (Calderhead 2011; Girolami and Calderhead 2011) to use the Hessian of the negative log prior plus the *expected Fisher information matrix*, $\mathcal{I}(\boldsymbol{\theta})$, defined by

$$\mathcal{I}_{ij} = \mathrm{Cov}\left(\frac{\mathrm{d}\log p(\mathbf{y}|\boldsymbol{\theta})}{\mathrm{d}\theta_i}, \frac{\mathrm{d}\log p(\mathbf{y}|\boldsymbol{\theta})}{\mathrm{d}\theta_j}\right). \tag{3.91}$$

For models described by ODEs, with independent Gaussian observation noise, this becomes (Girolami and Calderhead 2011)

$$\mathcal{I}_{ij} = \sum_{k=0}^{N_{obs}-1} \frac{\partial \mathbf{Y}_k^T}{\partial \theta_i} \boldsymbol{\Sigma}_k^{-1} \frac{\partial \mathbf{Y}_k}{\partial \theta_j} \tag{3.92}$$

where $\mathbf{Y}_k$ is the vector of predicted measured variables at measurement time point k, $\boldsymbol{\Sigma}_k$ is the measurement noise covariance matrix at time k, and N_{obs} is the number of observations. The systems considered in this thesis will have a single observed output, such as plasma concentration, therefore, $\mathbf{Y}_k$ will be a scalar. The derivatives with respect to the parameters θ_i can be obtained via the sensitivity equations (see Section 2.7.1).

SMMALA can be seen as being analogous to second-order optimisation methods, in which the objective function at each step is approximated by a second-order Taylor series expansion. Here, the log-posterior is approximated by a quadratic function, resulting in the posterior being approximated by a Gaussian distribution. Proposals are generated by drawing a sample from this Gaussian distribution.

3.3.8 Other methods

There is a great variety of MCMC methods in addition to the ones already mentioned, including:

Slice sampling (Neal 2003) is a method for drawing samples from univariate distributions, although extensions to the multivariate case also exist. This method treats the area under the curve as a two-dimensional uniform distribution and performs Gibbs-like sampling from it. Since it is intended for one- or lower-dimensional distributions, it has to be combined with other updates.

Hamiltonian Monte Carlo (Duane et al. 1987; Neal 2011) is a method based on the observation that for a system of particles in physics, the log-probability density of the system's state is proportional to its negative potential energy. Sampling can therefore be performed by treating the parameter vector as the position of a hypothetical particle subject to a suitably defined potential, and simulating its time evolution. One variant of Hamiltonian Monte Carlo, NUTS (No U-Turn Sampler) can reduce the need for manual tuning (Hoffman and Gelman 2014).

The Affine Invariant Ensemble Sampler (Goodman and Weare 2010; Foreman-Mackey et al. 2013) simultaneously simulates multiple copies of the parameter space. Updates are performed by exchanging information between these copies.

In this thesis, Gibbs sampling (Section 3.3.5) is used to update the regularisation parameter, while the basis function coefficients are updated using either RWMH (Section 3.3.4), MALA (Section 3.3.6), or SMMALA (Section 3.3.7). The other methods are mentioned mainly for completeness.

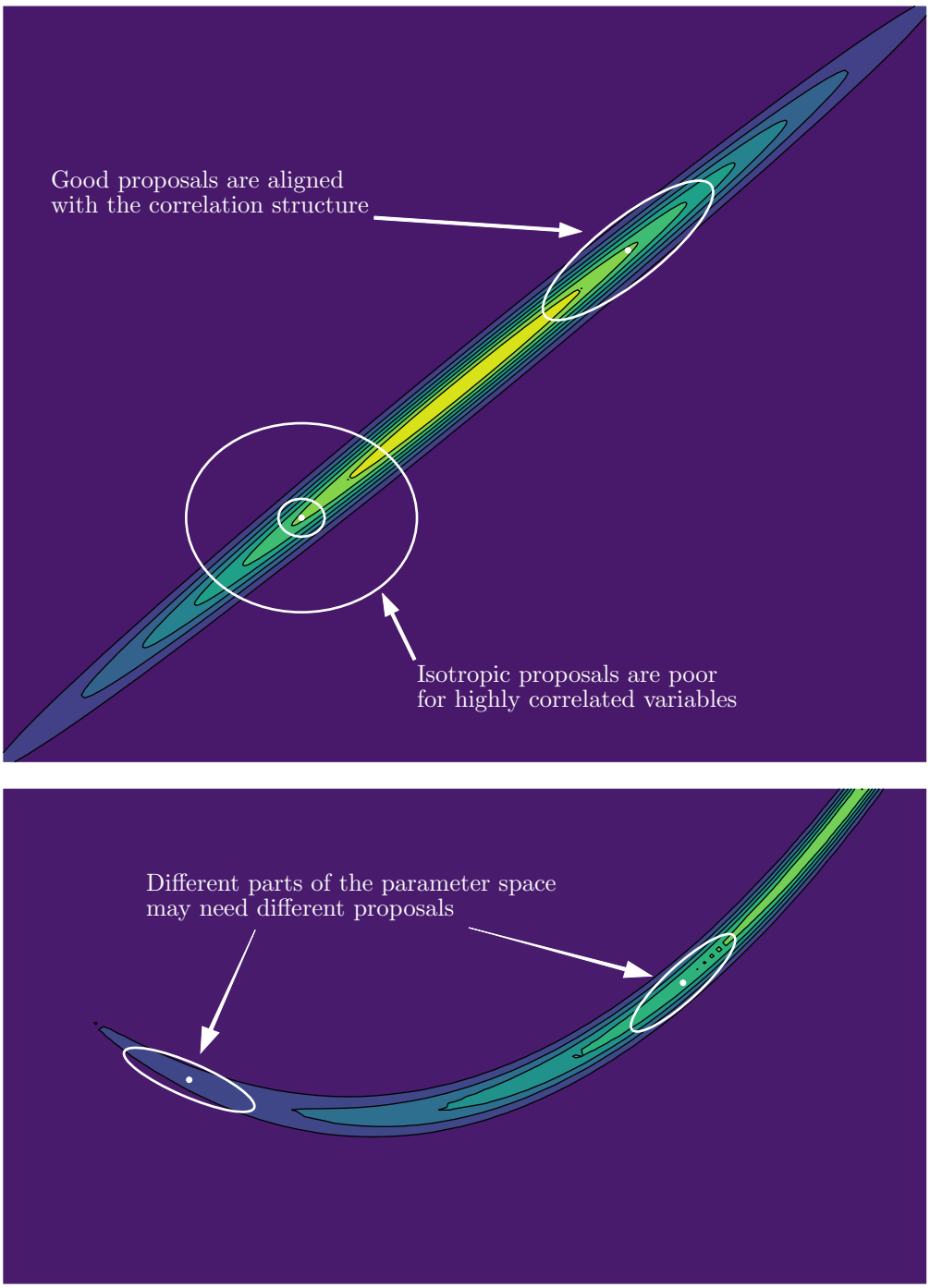

Figure 3.3: Proposal distributions for correlated parameters. Top: For correlated parameters, the proposal distribution should be aligned with the target distribution. Isotropic proposals will not be efficient. Bottom: Proposals can be more efficient if they are adapted to the local correlation structure.

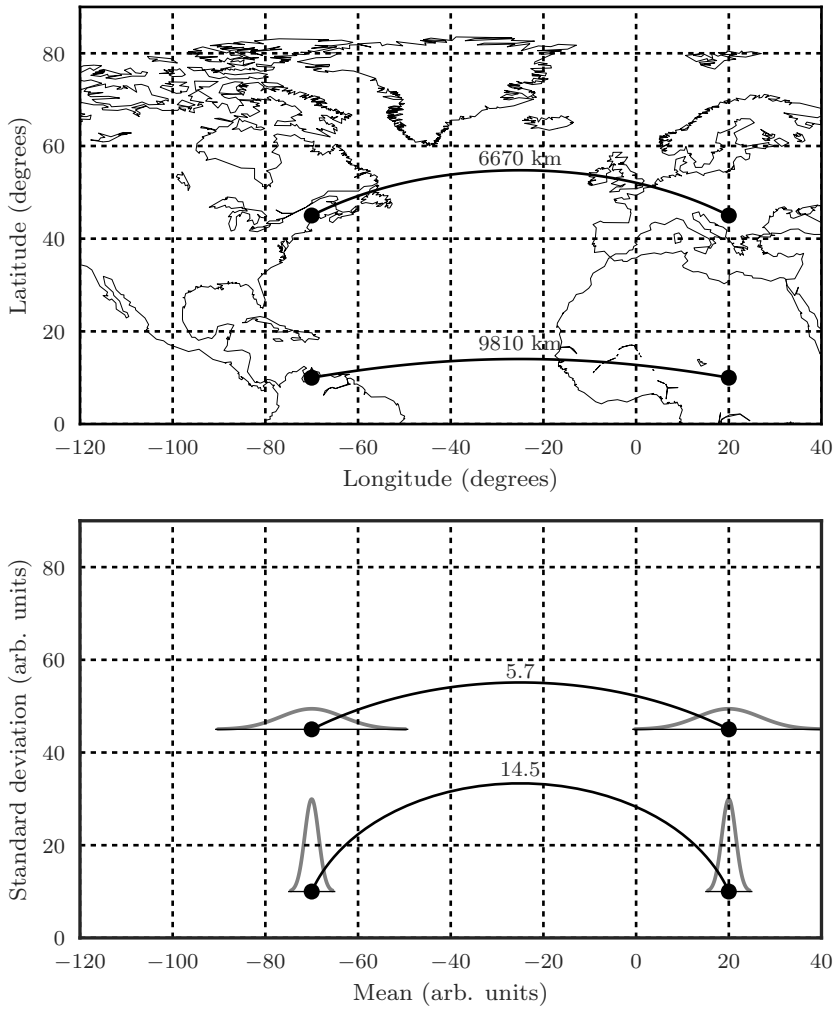

Figure 3.4: Analogy between the shortest path between points on the Earth and points in a statistical model. Top: Distances between points on the Earth. When points are parameterised using latitude and longitude, it may appear that the distance is the same between the upper and the lower two points. In reality, the distance is shorter between the upper points. Also, the shortest path between points is not the path that appears straight in the chart. This can be treated mathematically by modelling the surface of the Earth as a Riemannian manifold, where each point is equipped with a metric tensor. Bottom: A space of Gaussian distributions, parameterised using mean and standard deviation. This can also be treated as a Riemannian manifold. In this example, the Fisher information matrix with 10 observed data points is used to define a metric.

Chapter 4

Case study — eflornithine absorption

In this chapter, input-estimation methods are tested and benchmarked on a plasma-concentration dataset of the drug eflornithine, following oral administration in rats. Eflornithine is used to treat human African trypanosomiasis (HAT) (Jacobs et al. 2011). Optimal control-based and MCMC methods are applied on synthetic and real data, and evaluated based on computational speed, estimation accuracy, and robustness.

4.1 Background

HAT, also known as sleeping sickness, is a disease endemic to sub-Saharan Africa. It is caused by the parasite *Trypanosoma brucei*, which is transmitted by the tsetse fly (*Glossina*). The vast majority (>97%) of HAT infection cases are caused by the subspecies *Trypanosoma brucei gambiense* (Kennedy 2013), which is the variant of the disease that is considered here. In the first stage of the disease, the parasite invades the lymph and blood. Symptoms at this stage include headaches, weight loss, fatigue, and fever. In the second stage, the parasite invades the central nervous system, resulting in symptoms including sleep disturbances and psychiatric disorders. Typical duration of the disease is three years, equally divided between the first and second stage (Brun et al. 2010). HAT is almost always fatal if left untreated, although rare cases of recovery have been reported (Kennedy 2013). As of 2011, there are an estimated 20,000 cases of HAT per year, although historically, this has been significantly higher (World Health Organization 2015). The drugs that are available for treating second-stage HAT are melarsoprol and eflornithine (Eperon et al. 2014).

Eflornithine is preferred over melarsoprol, as it has been shown to result in a lower mortality rate, and to cause significantly less adverse effects (Chappuis et al. 2005; Balasegaram et al. 2006).

Eflornithine (DL-α-difluoromethylornithine, DFMO) is an irreversible inhibitor of ornithine decarboxylase (ODC). ODC is an enzyme required in the biosynthesis of polyamines (Ning et al. 2003), which are essential for cell growth and multiplication (Eperon et al. 2014). *Trypanosoma brucei gambiense* is more vulnerable to inhibition of ODC than human cells, possibly because ODC in humans has a faster turnaround time, causing inhibited ODC to be replaced faster (Heby et al. 2003). Standard eflornithine treatment involves infusions every six hours over fourteen days. The frequent infusions are necessary due to the short half-life of the drug (Brun et al. 2010). No oral form of the drug is available (Fairlamb 2003). Intravenous (IV) administration of eflornithine is costly and inconvenient, especially in rural hospitals (Na-Bangchang et al. 2004). Eflornithine is administered as a racemic mixture, with the L-enantiomer having a higher potency than the D-enantiomer (Johansson et al. 2013).

To make eflornithine treatment more widely available, it would be of interest to develop an oral mode of administration. A rat study has been performed (Johansson et al. 2013) with the purpose of determining the oral bioavailability and the absorption profile of the drug. Data from IV administration was used to build a nonlinear three-compartment model of the PK. Eflornithine was administered as a racemic mixture, with L- and D-enantiomers measured separately. While the same model structure was used for both enantiomers, the model parameters were estimated separately.

In this chapter, the model and data from this study are used as a benchmark problem, in order to investigate the performance of input-estimation methods.

4.2 Model

The system model developed by Johansson et al. (2013) is a three-compartment model with a nonlinear binding compartment, shown in Fig. 4.1. The model equations are

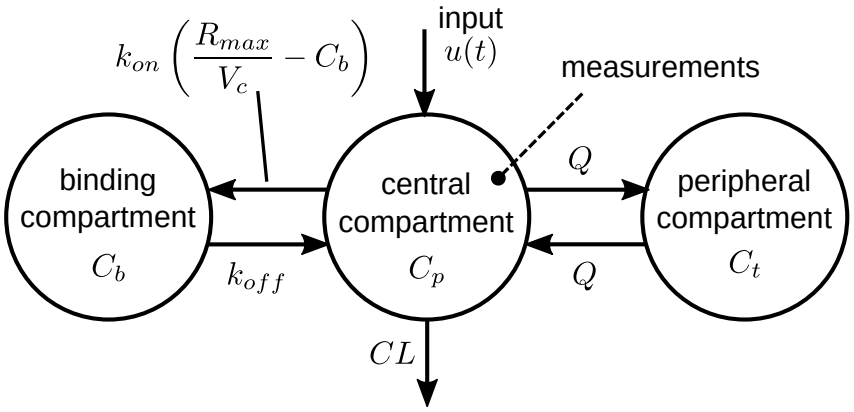

Figure 4.1: The nonlinear three-compartment eflornithine model from Johansson et al. (2013). The aim of input estimation is to estimate the function $u(t)$ from measurements of C_p.

defined as

$$\begin{aligned}\frac{\mathrm{d}C_p}{\mathrm{d}t} =& \frac{u}{V_c} - \left(\frac{CL}{V_c} + \frac{Q}{V_c}\right) C_p + \frac{Q}{V_c} C_t \\ & - k_{on} \cdot C_p \cdot \left(\frac{R_{max}}{V_c} - C_b\right) + k_{off} \cdot C_b \end{aligned} \tag{4.1a}$$

$$\frac{\mathrm{d}C_t}{\mathrm{d}t} = \frac{Q}{V_t} C_p - \frac{Q}{V_t} C_t \tag{4.1b}$$

$$\frac{\mathrm{d}C_b}{\mathrm{d}t} = k_{on} \cdot C_p \cdot \left(\frac{R_{max}}{V_c} - C_b\right) - k_{off} \cdot C_b \tag{4.1c}$$

where C_p, C_t and C_b are the drug concentrations in the central, peripheral and binding compartments, V_c is the volume in the central compartment, V_t is the volume in the peripheral compartment, CL is the clearance, Q is the intercompartmental clearance between the central and the peripheral compartment, k_{on} is the binding rate constant, k_{off} is the dissociation rate constant, and R_{max} is the total binding capacity. This model structure is used for both the L- and the D-enantiomer, although the parameter values are slightly different. In this work, the mean parameters for the L-enantiomer were used, since only datasets of measurements of this enantiomer were considered. The parameters are summarised in Table 4.1. Note that the equations presented here are slightly modified from the equations in Johansson et al. (2013), in order to properly make the conversion from amounts to concentrations.

Parameter	Definition	Value	Unit
V_c	Central compartment volume	74.7	mL
V_t	Peripheral compartment volume	31.6	mL
CL	Clearance	3.36	$\text{mL} \cdot \text{min}^{-1}$
Q	Intercompartmental clearance	0.217	$\text{mL} \cdot \text{min}^{-1}$
k_{on}	Binding rate constant	0.00275	$\text{mL} \cdot \text{min}^{-1} \cdot \mu\text{mol}^{-1}$
k_{off}	Dissociation rate constant	0.000468	min^{-1}
R_{max}	Total binding capacity	73.3	μmol

Table 4.1: Pharmacokinetic parameters of the eflornithine model.

4.3 Data

4.3.1 Real data

In the study reported by Johansson et al. (2013), oral administration of eflornithine hydrochloride monohydrate was given in doses of 40, 150, 400, 1,200, and 3,000 mg/kg body mass, with five or six animals per dose group. Since racemic mixtures were administered, the dose per enantiomer is half of the reported dose. The molecular weight of the administered substance is 236.65 g/mol (Johansson et al. 2013), resulting in doses *per enantiomer* of 85, 320, 850, 2,500, and 6,300 μmol/kg body mass. Plasma concentration time series were reported per individual. Two of these time series were selected for evaluating the input-estimation methods: one for the lowest and one for the highest dose. The data are shown in Fig. 4.2, and are presented in tabular form in Table A.1.

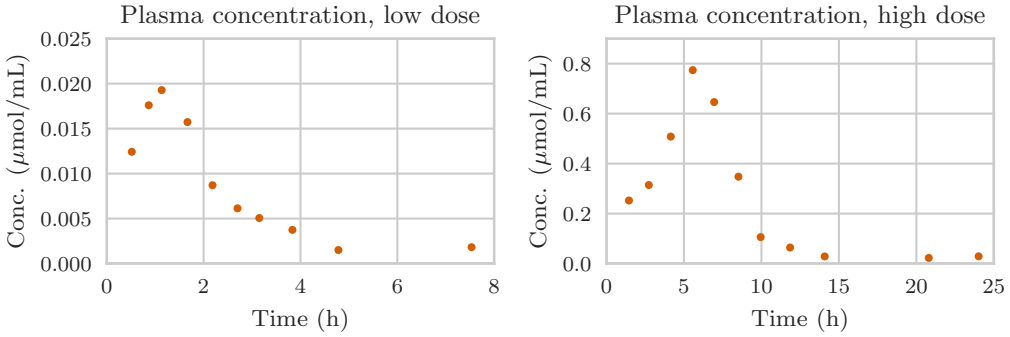

Figure 4.2: Plasma concentration measurements for the eflornithine low-dose data (left) and high-dose data (right). Note that the concentration and time scales are different for the two plots.

Parameter	Interpretation	Value	Unit
a	Initial amount	8.63	μmol
k_{tr}	Transition rate constant	0.025	min^{-1}
n	Number of compartments	2	-

Table 4.2: Parameters for the synthetic input function to the eflornithine model.

4.3.2 Test data

In order to compare the results to a known input, the system was simulated using a synthetic input function. When selecting a form for the input function, the following criteria were considered:

- The function should result in data that are qualitatively and quantitatively similar to real data.
- The function should have a simple functional form.
- The function should ideally be interpretable as the output to a compartmental model. The reason for this is that absorption is often modelled by a compartmental model, suggesting that this is a good model that can capture typical absorption processes. This also makes it possible to give the function a model structure interpretation.

The input model was chosen to have the following form:

$$u(t) = a\frac{k_{tr}^n \cdot t^{n-1} \cdot e^{-k_{tr}t}}{(n-1)!} \tag{4.2}$$

where a and k_{tr} are real-valued parameters, and n is an integer-valued parameter. By varying these parameters, various absorption profiles can be captured. This is an Erlang distribution (Forbes et al. 2010). It can be interpreted as the output from a linear compartmental model consisting of a chain of n compartments, connected with the rate constant k_{tr}, and where the first compartment receives the initial amount a of the drug, and all other compartments have zero initial conditions. This is known as the "Linear Chain Trick" (Jacquez 1985, Ch. 8). The resulting compartmental model is shown in Fig. 4.3.

The input function parameters were chosen to generate data similar to the low-dose real data. No synthetic data were created for the high dose, in order to keep the size of the test dataset manageable. It was found that a good fit to the real data for the lower dose was obtained by the values given in Table 4.2. Simulated

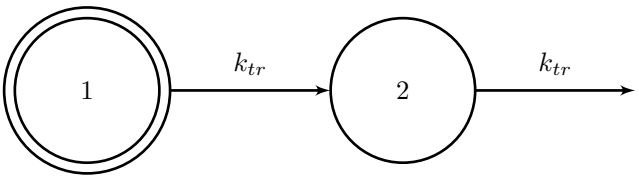

Figure 4.3: Compartmental model corresponding to the synthetic Erlang input function. A double circle is a compartment with non-zero initial conditions.

continuous-time plasma concentration profiles were obtained by using this function as input to the PK model. From this profile, four datasets were created:

Densely sampled, no noise. Samples were obtained at 100 equispaced time points between 0 and 7.53 hours, which was the last sampled time for the real data.

Densely sampled, with noise. This dataset has the same sampling schedule as above, but the measurements were perturbed with zero-mean, Gaussian noise, with a proportional standard deviation of 10% of the true value.

Sparsely sampled, no noise. In this dataset, samples were obtained at the same time points as the real data.

Sparsely sampled, with noise. This dataset had samples obtained at the same time points as the real data, with 10% proportional Gaussian noise.

The 10% noise standard deviation level was chosen based on the residual error reported by Johansson et al. (2013), which was found to be 17.7% with a relative standard error of 8.82%. As the main purpose of this work is to benchmark the input-estimation methods, it was deemed sufficiently accurate to use a standard deviation of the same order of magnitude as previously reported values. The test data are shown in Fig. 4.4.

4.4 Materials and methods

The test data and the real data were analysed using the same methods. For the noise-free test data, it could be argued that is it desirable to require the predicted plasma concentration to coincide exactly with all measurements. For optimal control-based methods, this could be achieved by replacing the discrepancy criterion constraint by one additional equality constraint for each measurement time point in the optimisation problem. For the purposes here, this approach has the disadvantage that it results in algorithms that are substantially different from the algorithms that are used

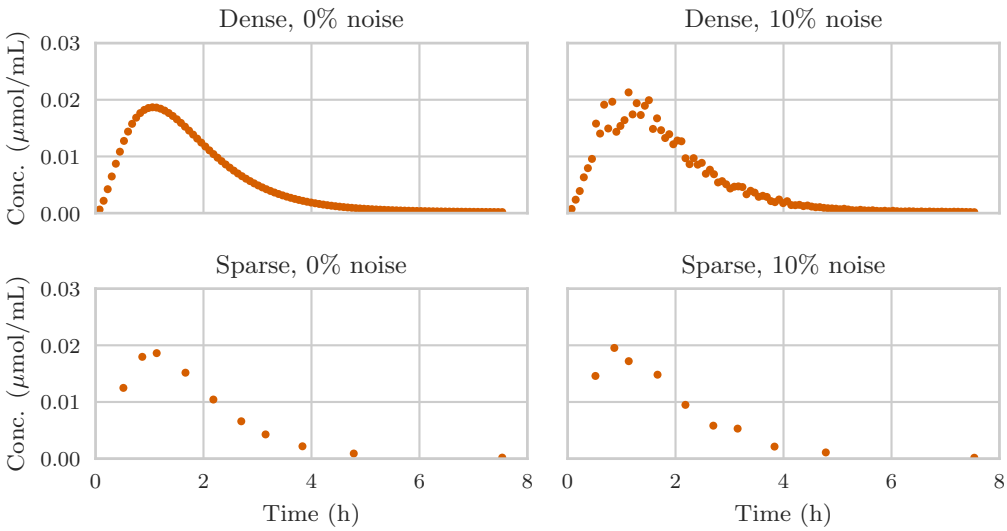

Figure 4.4: The synthetic test dataset generated for the eflornithine model.

to analyse the real data, where zero measurement noise is never assumed. As a pragmatic compromise, the noise-free test data were assumed to have a very small proportional error with a standard deviation of 0.1%, which is a factor 100 smaller than the standard deviation of the noisy data. This figure was also used for the MCMC analysis of the noise-free test data. The noisy test data as well as the real data were assumed to have a 10% proportional standard deviation.

When selecting methods for evaluation, those which can be used without any manual intervention or tuning were mainly considered. There are two major reasons for this:

1. A very large number of estimation procedures have to be performed to evaluate various combinations of priors, function parameterisations, and optimisation or sampling methods. Any requirements for manual intervention would make the analysis impractical.

2. One of the major evaluation criteria for these methods is ease of use for non-experts. A method that requires manual steps to be performed would be less appealing to a non-expert, and make the method impractical in an industry setting.

One exception to this approach was made, where the L-curve method was applied to determine the regularisation parameter for the optimal-control methods as well as for the MCMC methods.

4.4.1 Optimal-control methods

Optimal control-based methods were applied to the four synthetic datasets as well as to the two real datasets. The following priors were used: penalisation of the first derivative, penalisation of the second derivative, and penalisation based on the maximum entropy prior. Penalisation of the derivatives was also applied to the logarithm of the input function. The maximum entropy prior was not applied to the logarithm of the function, as this prior inherently enforces non-negativity. Applying a maximum entropy prior in the log domain would therefore constrain the input function to be greater than 1 at all time points, which is clearly undesirable. For the cases where the derivative of the input function was penalised directly, non-negativity constraints were added to the optimisation problem in order to avoid non-physical solutions.

For the choice of function parameterisation, only piecewise constant basis functions were used. The rationale behind this decision is that, as long as the parameterisation is sufficiently fine-grained, it is for all practical purposes able to represent any function of interest. For this dataset, the input function was discretised in 100 intervals, uniformly distributed from time 0 to the time of the final measurement. Optimal-control methods can in many cases handle such fine parameterisation, as will be shown in the results section. Furthermore, this parameterisation makes the multiple shooting and collocation schemes maximally sparse, since each function coefficient only participates in the continuity and collocation constraints of a single interval.

As the piecewise constant functions are not differentiable at the jump points, the derivative-based priors were computed using finite differences, and integration was approximated by summing over all intervals. Denoting the input function value in the kth interval by u_k, the approximation of the first-derivative regularisation term is given by

$$E_R = \int_{t_i}^{t_f} \left(\frac{\mathrm{d}u(t)}{\mathrm{d}t} \right)^2 \mathrm{d}t \approx \sum_{k=0}^{N_B-2} \frac{(u_{k+1} - u_k)^2}{\Delta t}. \tag{4.3}$$

where Δt is the length of each discretisation interval. Similarly, the second derivative regularisation term is given by

$$E_R = \int_{t_i}^{t_f} \left(\frac{\mathrm{d}^2u(t)}{\mathrm{d}t^2} \right)^2 \mathrm{d}t \approx \sum_{k=1}^{N_B-2} \frac{(u_{k+1} - 2u_k + u_{k-1})^2}{(\Delta t)^3}. \tag{4.4}$$

The maximum entropy prior is defined on a discrete-time version of the input

function and does not require any further discretisation.

All three direct optimal-controls methods presented in Section 3.2 were used: single shooting, multiple shooting, and collocation. For simplicity, the shooting and collocation intervals were chosen to coincide with the intervals chosen for the input parameterisation. The collocation method used Lagrange polynomials of degree 3 with Radau collocation points, which are given by the roots to the polynomial

$$\frac{\mathrm{d}^{d-1}}{\mathrm{d}t^{d-1}}(t^{(d-1)}(t-1)^d) \tag{4.5}$$

where d is the order of the polynomial (Hairer and Wanner 1999). For $d = 3$, the roots are located at the positions $\left[\frac{2}{5} - \frac{\sqrt{6}}{10}, \quad \frac{2}{5} + \frac{\sqrt{6}}{10}, \quad 1\right]$. Although other collocation schemes could have been investigated, these settings were assumed to be sensible defaults.

For the most part, the discrepancy criterion was used to determine the regularisation parameter τ, despite the fact that this has been shown to overestimate τ to some extent (Twomey 1965). The reason for this choice is that this criterion makes it possible to determine τ as well as the optimal solution using a single optimisation procedure. The discrepancy criterion suggests that χ^2, the sum of squared prediction residuals, scaled by the measurement standard deviation, should not exceed the number of measurements:

$$\chi^2 = \sum_{j=0}^{n-1} \left(\frac{y_j^{(pred)} - y_j}{\sigma_j} \right)^2 \leq n, \tag{4.6}$$

where n is the number of measurements, $y_j^{(pred)}$ is the predicted measurement at time j, y_j is the measurement at time j, and σ_j is the assumed measurement standard deviation at time j. This makes it possible to pose the optimal-control problem, for the case of single shooting, as:

$$\begin{aligned}
\underset{\mathbf{a}}{\text{minimise}} \quad & E_R(\mathbf{a}) && \text{(4.7a)}\\
\text{such that} \quad & \chi^2 \leq n && \text{(4.7b)}\\
& \mathbf{x}(t_i) = \mathbf{x}^{(0)} && \text{(4.7c)}\\
\text{and} \quad & \mathbf{h}(\mathbf{a}, \mathbf{x}(t)) \leq \mathbf{0} && \text{(4.7d)}\\
\text{where} \quad & \mathbf{x}(t) = \Phi(t_i, t, \mathbf{x}^{(0)}, \mathbf{a}).
\end{aligned}$$

The term $E_R(\mathbf{a})$ is the regularisation term. The inequality constraints $\mathbf{h}(\cdot)$ are in this case the nonnegativity constraints for the input function. For multiple shooting and collocation, the problem is modified analogously: the objective is changed from the penalised log-likelihood to the regulariser, and the χ^2 value is introduced as an extra inequality constraint. When negative entropy is used as the regulariser E_R, and the system is linear, this algorithm becomes identical to the maximum entropy algorithm presented by Charter and Gull (1987) and Hattersley et al. (2008).

In addition to the discrepancy criterion, the L-curve approach was applied to the real datasets, using penalisation of the second derivative. Estimation was performed for 80 values of τ, logarithmically spaced between 10^3 and 10^{11}. This range was considered to cover any reasonable setting of τ. Collocation was used as the optimisation method, in order to keep the total running time low. The resulting L-curves were plotted, and suitable values of τ were manually selected. To assess the sensitivity of the estimate to τ, three values were selected for each dataset: one at the "knee" of the curve, and one on either side of the knee.

For the single shooting method, there are $N_B = 100$ decision variables, each representing the input function for each discretisation interval. This method has one inequality constraint representing the discrepancy criterion, and an additional N_B inequality constraints enforcing nonnegativity of the input function, for the cases where such constraints are desired. The multiple shooting method adds an additional $N_B \cdot d_x$ decision variables, representing the states at the beginning of each discretisation interval, and an additional $N_B \cdot d_x$ equality constraints enforcing the initial conditions and the continuity constraints. The collocation methods require, in addition to the variables and constraints of single shooting, an additional $N_B \cdot (d+1) \cdot d_x$ decision variables, as the trajectory of each state in each interval is represented by a polynomial of degree d. They also add an additional $N_B \cdot (d+1) \cdot d_x$ equality constraints, accounting for the initial conditions, continuity constraints, and collocation constraints. Table 4.3 reports the number of variables and constraints per method.

In summary each of the four test datasets and the two real datasets were analysed using 15 combinations of priors and optimisation methods. This is summarised in Table 4.4.

4.4.2 MCMC methods

MCMC estimation was applied to the four synthetic and the two real datasets. Four priors were used: penalisation of the first derivative, penalisation of the second derivative, applied to the input function as well as to the logarithm of the input function. For the derivative-based priors, the regularisation parameter τ can be

Method	Decision variables	Equality constraints	Inequality constraints
Single shooting	100	0	1/101
Multiple shooting	400	300	1/101
Collocation	1300	1200	1/101

Table 4.3: Number of decision variables and constraints for the optimal-control problem formulations of the eflornithine model. For single shooting, only the input function values are used as decision variables. For the other methods, the states at the discretisation interval boundaries (multiple shooting and collocation) and the states at the collocation points (collocation) are added to the problem. The number of inequality constraints is either 1 or 101, depending on whether nonnegativity of the input function is enforced.

Datasets	Methods	Priors
Test data, dense noise-free	Single shooting	1st derivative
Test data, dense 10% noise	Multiple shooting	1st derivative, log
Test data, sparse noise-free	Collocation	2nd derivative
Test data, sparse 10% noise		2nd derivative, log
Real data, low dose		Entropy
Real data, high dose		

Table 4.4: A summary of datasets, methods and priors used to evaluate the optimal-control methods on eflornithine. All combinations of dataset, method, and prior were tested. In total, this amounted to $6 \times 3 \times 5 = 90$ combinations.

interpreted as the process noise variance of the input function (see Section 2.4.1). This makes it straightforward to assign a joint probability distribution over τ and the input function, which is necessary in order to incorporate τ as a variable in the probabilistic model. For maximum entropy priors, similar interpretations are more difficult to make. For this reason, the maximum entropy criterion was not used.

Two choices of function parameterisation were investigated: B-splines and Karhunen-Loève (KL) basis functions. Because of the high computational cost of MCMC methods, it was deemed important to ensure that the number of parameters was kept relatively small. For this reason, piecewise constant functions were not investigated, as these require a large number of parameters in order to not result in unrealistic staircase-like functions. For the B-spline parameterisation, cubic splines were used. For simplicity, the knots of the spline were placed at time 0, and at each measurement time point. The first and last knots have to be treated specially. This is because cubic splines represent the function in each interval by a polynomial of degree 3, requiring 4 parameters. Therefore, each interval must be covered by

4 basis functions. As each basis function extends over 4 intervals, this condition is automatically satisfied for intervals sufficiently far away from the first and last knot. However, if a single knot was placed at time 0, the first interval would only be covered by a single basis function. A simple remedy is to introduce 4 knots at time 0. Similarly, 4 knots can be introduced at the last measurement time. To compute the prior for a particular realisation of the input, the function was evaluated at the spline knots. For penalisation of the first derivative, the continuous-time prior is given by

$$\log p(u) = -\frac{1}{2}\tau \int_{t_i}^{t_f} \left(\frac{\mathrm{d}u(t)}{\mathrm{d}t}\right)^2 \mathrm{d}t + C(\tau), \tag{4.8}$$

where $C(\tau)$ is a normalisation term that is included to ensure that the distribution integrates to 1. Discretising the first term into n time intervals, where the interval between knot j and $j+1$ has length Δt_j, yields

$$-\frac{1}{2}\tau \int_{t_i}^{t_f} \left(\frac{\mathrm{d}u(t)}{\mathrm{d}t}\right)^2 \mathrm{d}t \approx -\frac{1}{2}\tau \sum_{j=0}^{n-1} \left(\frac{u(t_{j+1}) - u(t_j)}{\Delta t_j}\right)^2 \cdot \Delta t_j, \tag{4.9}$$

where the derivative has been replaced by finite differences. In this discrete version, the increments of the input function, $u(t_{j+1}) - u(t_j)$, are independent Gaussian random variables with mean 0 and variance $\Delta t_j/\tau$. This result is in agreement with the discussion in Section 3.1.1, where a prior corresponding to penalisation of the first derivative was identified with a Wiener process. The discrete-time version of the normalisation term $C(\tau)$ can be identified as the normalisation constant of this Gaussian distribution, yielding a discrete-time log-probability density of:

$$\log p(u) \approx -\frac{1}{2}\tau \sum_{j=0}^{n-1} \frac{(u(t_{j+1}) - u(t_j))^2}{\Delta t_j} + \frac{1}{2}\sum_{j=0}^{n-1} \log\left(\frac{\tau}{2\pi\Delta t_j}\right). \tag{4.10}$$

If τ is kept fixed, the normalisation terms are constant and can be omitted from the computation. When τ is treated as an unknown parameter to be estimated, these terms have to be kept. For penalisation of the second derivative, the above procedure can be performed on the derivative of the input function. This differentiation can be performed analytically, as each basis function is a polynomial.

When estimating τ, it is important that the spline is only evaluated at the knots, and not at any intermediate points. The cubic spline interpolant is smoother than the original continuous-time model. Hence, any estimate of τ relying on arbitrary

intermediate points would severely overestimate τ in order to account for the apparent smoothness. This is illustrated in Fig. 4.5.

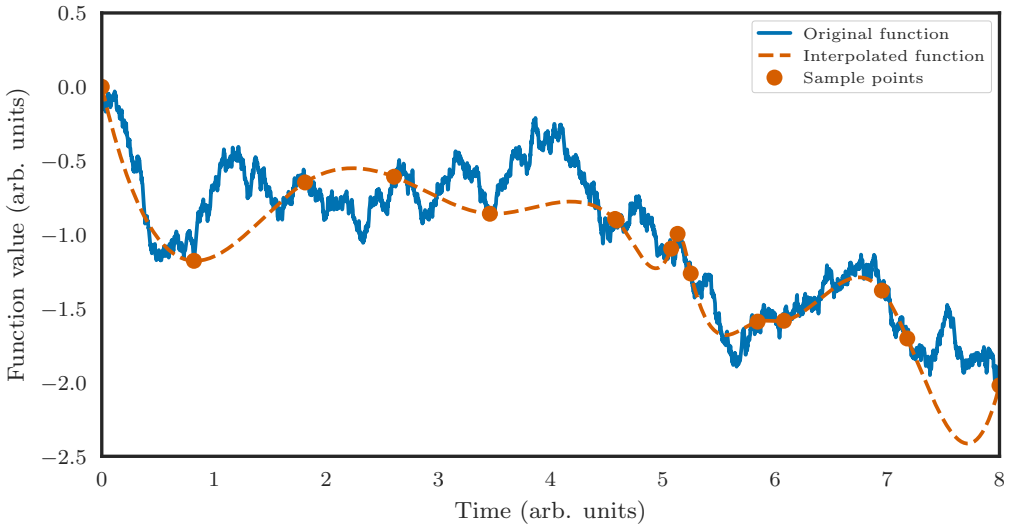

Figure 4.5: Sample realisation of a scaled Wiener process with $\tau = 5$, corresponding to penalisation of the first derivative. This function is sampled at 15 time points. A maximum likelihood estimate of τ based on these points yields a value of 4.5, reasonably close to the true value. Interpolating these points with a cubic spline, and performing maximum likelihood estimation using 5000 equispaced values from the interpolant yields an estimate of τ of 685. The reason for this discrepancy is that the interpolating spline is smoother than the original function.

For the Karhunen-Loève basis functions for penalisation of the first derivative, an additional constant basis function was added, to allow the input function to start with a nonzero value. Similarly, for penalisation of the second derivative, one constant and one linear basis function were added to allow for a nonzero initial value and slope. The only other choice that needs to be made is the total number of basis functions. In all analyses, the number of basis functions was set to 20. One way to assess whether the number of basis functions is adequate is to consider what fraction of the expected energy is contained in the included basis functions. The energy content of a function $u(t)$ over the interval $[t_i, t_f]$ is given by

$$\int_{t_i}^{t_f} (u(t))^2 \, \mathrm{d}t = \int_{t_i}^{t_f} \sum_{j=0}^{\infty} (a_j \phi_j(t))^2 \, \mathrm{d}t = \sum_{j=0}^{\infty} a_j^2, \tag{4.11}$$

where the last equality follows from the fact that the functions $\phi_j(t)$ are an orthonormal set. As shown in Section 3.1, the coefficients a_j are independent zero-mean

Gaussian random variables with variance λ_j. The expected energy under this prior is therefore given by $\mathbb{E}\left[\sum_{j=0}^{\infty} a_j^2\right] = \sum_{j=0}^{\infty} \mathbb{E}\left[a_j^2\right] = \sum_{j=0}^{\infty} \lambda_j$. For penalisation of the first derivative, the expected energy is

$$\sum_{j=0}^{\infty} \frac{1}{\tau} \left(\frac{t_f - t_i}{\pi \left(\frac{1}{2} + j\right)} \right)^2 = \frac{(t_f - t_i)^2}{2\tau}. \tag{4.12}$$

As the first basis function is a constant offset function, in practice there are 19 Karhunen-Loève basis functions, accounting for 98.9% of the expected energy (Fig. 4.6). Penalisation of the second derivative can be analysed similarly. As the eigenvalues are computed numerically, no closed-form expression for the total energy is available. By numerical computation, it can be determined that truncation to 18 basis functions accounts for more than 99.99% of the total energy. Penalisation of the second derivative results in smoother functions, where very little energy is contained in the higher eigenvalues, that represent higher frequencies.

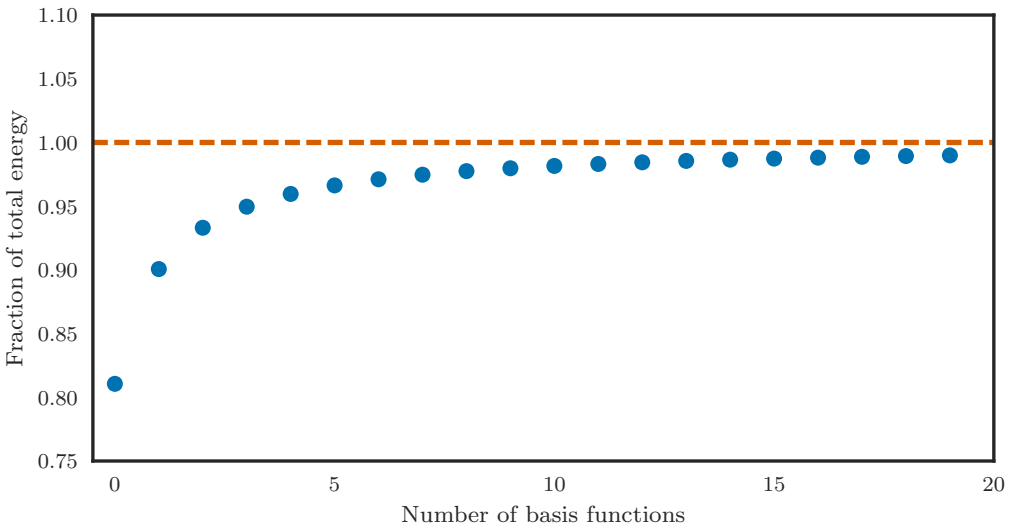

Figure 4.6: Energy content of the first few Karhunen-Loève basis functions for penalisation of the first derivative, as a fraction of total signal energy. Most of the energy ($\sim 99\%$) is contained in the first 20 basis functions.

If the MCMC sampling procedure is initialised with parameter values far from its stationary distribution, it is necessary to discard an initial part of the chain, a practice known as *burn-in* (see Section 3.3.3). For a poor initial state, the number of samples that have to be discarded may be impractically large. A reasonable initialisation strategy is to start at the MAP estimate, which can be found using

the optimal-control methods investigated here. The MCMC sampler chose an initial state from the results of the optimal-control methods based on the following strategy:

1. All optimal-control results for the dataset in question were searched for entries matching the following criteria: 1) the prior must match, and 2) the optimiser must have reported that optimisation was successful.

2. If no such matches were found, the requirement that the prior must match was dropped.

3. Once a match was found, the piecewise-constant parameterisation was converted to the B-spline or Karhunen-Loève parameterisation by least-squares fitting.

The optimal control methods do not provide a value for τ. However, initialisation of τ was not necessary, as it was updated using Gibbs sampling, where the updated value does not depend on the previous value.

In contrast to the optimal-control methods, imposing nonnegativity constraints for the MCMC methods is non-trivial. Inequality constraints on parameters can be handled by transforming the parameters (Carpenter et al. 2017). Another option is to use proposal distributions that assign zero probability density to the infeasible regions. These methods would be of limited value in this case, as inequality constraints on the basis function coefficients do not necessarily result in similar constraints for the functions. As an example, a Karhunen-Loève basis for penalisation of the first derivative is a set of trigonometric functions, which can attain negative values even when all of the coefficients are nonnegative. Another possibility could be to assign a probability density of 0 to any function that drops below 0 at any point. However, this could potentially be very inefficient, with a large number of rejected proposals. For the purposes of this work, no nonnegativity constraints have been included. Instead, any priors that result in a large number of negative solutions are considered to be unsuitable.

Four sampling methods were tested on the datasets:

- Single-component RWMH. In this method, the coefficients are updated one at a time, using univariate Gaussian proposals, centred on the current value.

- Block RWMH, where the coefficients are updated jointly using a multivariate Gaussian proposal, centred on the current value.

- MALA with preconditioning, with joint updating of all coefficients.

- SMMALA, with joint updating of all coefficients.

For SMMALA sampling, the Hessian of the negative log prior plus the expected Fisher information was used as the metric tensor. For all sampling methods with joint updates, the covariance of the proposal distribution was initially set to the inverse metric tensor at the optimal parameter setting as computed by the optimal-control methods. For the single-component RWMH method, the diagonal entries of the inverse metric tensor were used as proposal variances. This method may not necessarily result in an optimal proposal. In particular, the individual variances for single-component RWMH are the marginal variances, which can be significantly larger than the conditional variances if the coefficients are strongly correlated. Therefore, an initial tuning phase was added to the sampling process, where the proposal distributions were scaled according to the following procedure:

- The proposals were assigned a target acceptance ratio range, which was 20–50% for RWMH, 40–80% for MALA, and 40–100% for SMMALA.
- Every 50 samples, the acceptance ratio was estimated. If the ratio fell below the target range, the proposal covariance matrix was multiplied by a factor of $(0.9)^2$. If the ratio rose above the target range, the proposal covariance matrix was multiplied by a factor of $(1.1)^2$.
- Once all parameters had achieved acceptance ratios within the target range, or a timeout of 1000 samples had been exceeded, the tuning phase was considered to be complete, and updating of proposals was stopped.

Samples were only recorded after the tuning phase, in order to avoid the risk that the tuning might modify the stationary distribution of the Markov chain. The timeout was found to be necessary to ensure that no estimation procedure could become stuck indefinitely in the tuning phase.

In all cases, τ was assigned a Gamma distribution prior with parameters $\alpha_{prior} = \beta_{prior} = 10^{-3}$, and was updated using Gibbs sampling. These parameter value are commonly used (Gelman 2006), but care must be taken to ensure that they are appropriate, a point that will be discussed in the results section. For B-splines, the increments of the function value between knot j and $j+1$ have a precision of $\tau/\Delta t_j$. For Karhunen-Loève basis functions, the function coefficients have a precision of τ. τ was updated by drawing from a Gamma distribution with parameters $\alpha = \alpha_{prior} + n/2$ and $\beta = \beta_{prior} + \sum_{j=0}^{n-1} \theta_j^2/2$ (Section 3.3.5). For B-splines, j ranges over all increments between knots, and θ_j is the quantity $(u_{j+1} - u_j)/\sqrt{\Delta t_j}$, where u_j is the input function value at knot j (for penalisation of the first derivative), or the derivative of the input function at knot j (for penalisation of the second derivative).

For Karhunen-Loève basis functions, j ranges over all basis functions, excluding the constant and linear term, and θ_j is the coefficient for function j.

Sampling was performed by alternating between updating τ with Gibbs sampling, and updating the function coefficients with one of the methods presented above.

In total, 32 estimation procedures were performed for each dataset, for each combination of function parameterisation, prior, and sampling method. Each procedure was allowed to run for 5 minutes of processing time, only counting time when the process was executing. Table 4.5 shows a summary of the tested combinations.

Datasets	Methods	Parameterisations	Priors
Test data, dense noise-free	Single RWMH	B-spline	1st der
Test data, dense 10% noise	Block RWMH	Karhunen-Loève	1st der, log
Test data, sparse noise-free	MALA		2nd der
Test data, sparse 10% noise	SMMALA		2nd der, log
Real data, low dose			
Real data, high dose			

Table 4.5: A summary of datasets, methods and priors used to evaluate the MCMC methods on eflornithine. All combinations were tested, amounting to $6 \times 4 \times 2 \times 4 = 192$ combinations.

Additionally, estimation of τ using the L-curve approach was performed on the real datasets for the case of B-splines with single-component RWMH sampling, penalising the second derivative. As this method requires a substantial amount of manual intervention, it was not deemed feasible to perform this on a greater number of examples. The L-curves were generated by finding the MAP estimate of the spline input model, treating it as a single-shooting problem. As in the optimal-control case, the investigated values of τ were logarithmically spaced between 10^3 and 10^{11}. Adaptation of the proposal variances was performed using the same method as previously described, where the variances are adjusted during an initial tuning period to achieve a predetermined acceptance ratio.

4.5 Results and discussion

4.5.1 Optimal-control methods

To assess the performance of the optimal control-based methods, the following questions were addressed:

- How accurately did the methods recover the true input function?

- What was the typical running time of the methods?
- Are the methods robust? Did they always find the optimal solution?

The accuracy of the recovered input function can only be evaluated on the test dataset, as the true input functions of the real data are not known. The similarity between the true input and the estimate were measured using the root-mean-square error (RMSE) (Lindsten and Schön 2013). The RMSE between two vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$RMSE(\mathbf{x}, \mathbf{y}) = \sqrt{\frac{1}{n}\sum_{j=0}^{n-1}(x_j - y_j)^2} \tag{4.13}$$

where n is the number of entries in the vectors. Here, the entries of the vectors are the input functions evaluated in each discretisation interval. Additionally, the estimated total amount of drug absorbed was compared to the true amount. This quantity has a special interest in many PK applications, as it makes it possible to compute the bioavailability. The total amount is determined by computing the time integral of the input function. Since the input function is defined as a piecewise constant function, this is trivially achieved by summing the function values, scaled by the discretisation interval. The true total amount for the input dataset is a from Table 4.2, which follows from the fact that the Erlang distribution has to have an integral of 1. The RMSE values and the estimated amount, expressed as a fraction of the true amount, are shown in Table 4.6.

The estimate, and therefore the RMSE and total amount values, depend only on the prior. The optimisation methods are merely reformulations of the same problem, and the choice of method does not influence the solution, as long as the optimisation procedure is successful.

Examples of estimation results for all four test datasets are shown in Fig. 4.7. Figure 4.8 shows results for all priors on the noisy test data, which represent the most realistic situation. When the data are dense and noise-free, the estimated function is very close to the true function. The choice of prior has practically no effect. For this kind of data, the function is practically completely determined by the data. The estimated absorbed amount is also correct within two significant digits. This result confirms that, when presented with ideal data, the methods will achieve a correct result. When the data are noisy, the peak at time 0.7 hours tends to be underpredicted, especially for the penalisation of the first derivative. The likelihood is determined by the sum of the squared scaled residuals, regardless of whether these values are distributed evenly across time or are clustered within a small interval. By

Dataset	Prior	RMSE (μmol/h)	Normalised total amount
Dense, 0% noise	1der	0.10	1.00
	1der, log	0.10	1.00
	2der	0.10	1.00
	2der, log	0.10	1.00
	entropy	0.10	1.00
Dense, 10% noise	1der	0.29	0.94
	1der, log	0.21	0.97
	2der	0.26	0.97
	2der, log	0.19	0.99
	entropy	0.21	0.98
Sparse, 0% noise	1der	0.25	1.02
	1der, log	0.24	1.02
	2der	0.14	1.01
	2der, log	0.17	1.01
	entropy	0.22	1.02
Sparse, 10% noise	1der	0.44	0.96
	1der, log	0.36	1.00
	2der	0.55	1.00
	2der, log	0.50	1.02
	entropy	0.46	1.00

Table 4.6: Accuracy measures of the optimal-control methods, evaluated on the test data. The normalised total amount is the ratio between the estimated amount and the true amount, which is 8.63 μmol. The designations '1der' and '2der' refer to penalisation of the first and second derivative, while 'log' signifies that the function was modelled in the log domain.

choosing an input function that agrees with the data everywhere except at this peak, the method can find an input function with a higher prior that would be the case if the residual values were evenly distributed. For this reason, the total amount is also underpredicted. This effect is far less pronounced when penalisation is performed in the log domain. In the log domain, the initial peak is less pronounced, making the penalty for capturing the peak much lower. For noise-free data, the plasma concentration is forced to agree almost perfectly with the data, resulting in the peak being closely followed.

In general, a sparse sampling scheme with noise-free data results in a higher RMSE value than a dense sampling scheme with noisy data. In particular, sparse sampling causes the input function before the first measurement to deviate substan-

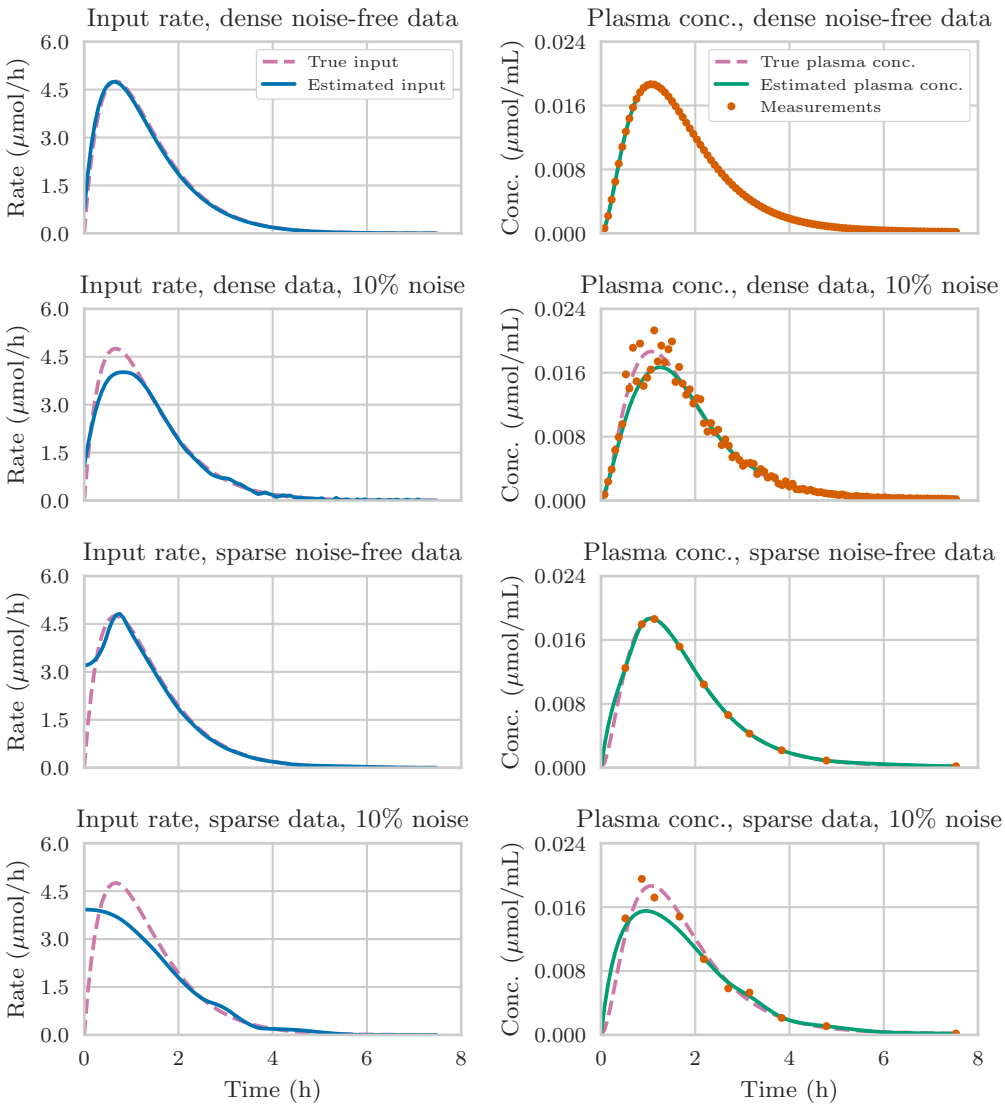

Figure 4.7: Optimal-control input-estimation results for the four test datasets, using penalisation of the first derivative. The estimate from dense, noise-free data is essentially identical to the true input function. For noisy data, the peak is underpredicted, as it can be partly explained by the noise. For sparse noise-free data, the estimate agrees well with the data except for the time up to the first measurement. Sparse noisy data exhibit both of these phenomena.

tially from the true function. While the true function is 0 at time 0, this fact is not supported by the data. Any input function that results in a plasma concentration similar to the first measurement is possible, and the sharp initial slope of the true function is penalised by all priors. If there is reason to believe that the function should

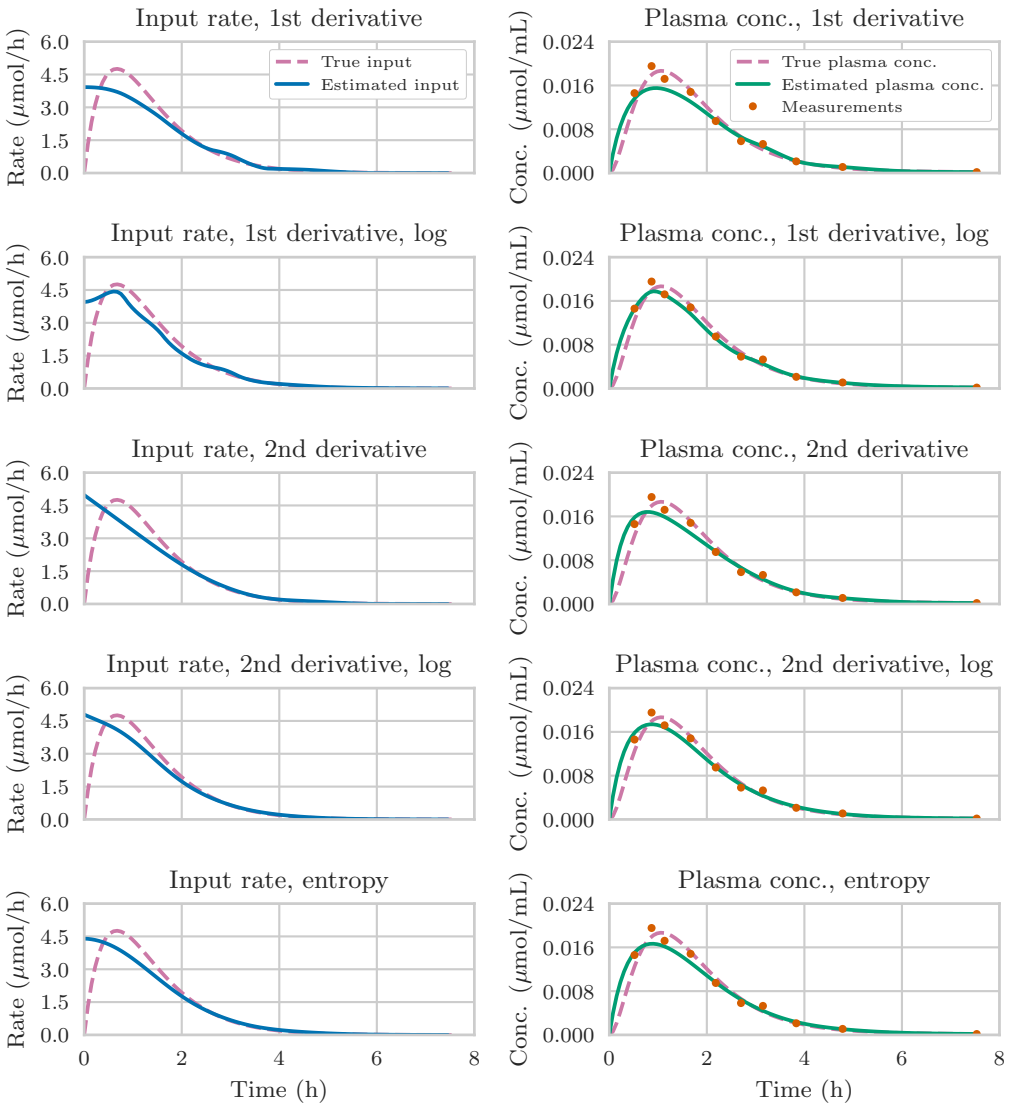

Figure 4.8: Optimal-control estimation results for sparse test data with 10% noise.

start at 0, it is possible to convey this to the method by an extra constraint. As with all other assumptions, this is a good idea only if the assumption is actually justified. In the case presented here, it is not obvious that the absorption process does not start immediately when the drug is administered. Note that in a traditional model using an absorption compartment, the absorption rate would be largest at $t = 0$. Another way of capturing the peak more accurately would be to use separate priors for the initial and later parts of the time series. However, this would require additional assumptions and modelling choices, such as determining how to partition the time

series. This could negate one of the major advantages of input estimation over parametric models — the reduced need to make strong assumptions. In Chapter 6, a case study is presented where such a partition was used, due to the vast differences in time scale between the initial and later parts of the time series.

In general, the estimated total absorbed amount is accurate to within a few percent of the true value. This result holds even when the estimated input function is significantly different from the true function. It can be seen that, in many cases, the estimate overpredicts the input initially, while the peak is underpredicted. Since the total amount is computed from the time integral, it is relatively insensitive to these variations, and the mispredictions partly cancel out. Any solution that would result in a substantially different amount would need to be consistently above or below the true function, in which case the measurements would not be predicted well. This is a valuable observation for the cases where the bioavailability is the main quantity of interest from the input-estimation procedure.

The computational performance of the methods was evaluated based on two criteria: the running time, and whether the method was able to find the optimum. Additionally, the number of iterations was reported. While this number is not of primary interest, it can be helpful in determining whether any difficulties are caused by the cost function and its derivatives being expensive to evaluate, or by the cost function having a form that requires many iterations. These criteria can obviously be applied to test data as well as real data, and a summary of the results are presented in Tables 4.7 and 4.8. A method was considered to be successful if the status code that was returned from IpOpt was one of:

Solve Succeeded. The method was able to find a local optimum within the desired tolerances. By default, IpOpt uses a relative tolerance of 10^{-8}, but this value can be changed by the user (Kawajir et al. 2016).

Solved To Acceptable Level. The method was able to find a local optimum within the acceptable, but not the desired, tolerances.

There are several reasons why the method may not find a solution. Examples of status codes for unsuccessful procedures include:

Infeasible Problem Detected. The method has converged to a local minimum of the constraint violation, but the point is infeasible.

Restoration Failed. When the optimiser fails to find an update step, it enters a *restoration phase*, where it attempts to reduce the constraint violation without

considering the cost value (Wächter 2009). If the restoration phase fails, the optimiser exists with this error message.

A full description of IpOpt status codes can be found in Kawajir et al. (2016).

	Median time (s)	Median iterations	Successful runs
Single shooting	9.5	29	29/30
Multiple shooting	6.1	31	30/30
Collocation	0.2	30	27/30

Table 4.7: Median running times, number of iterations, and the proportion of successful estimation procedures, organised by optimisation method.

	Median time (s)	Median iterations	Successful runs
1der	5.1	28	18/18
1der, log	7.9	39	16/18
2der	5.7	30	17/18
2der, log	4.7	31	17/18
entropy	6.0	28	18/18

Table 4.8: Median running times, number of iterations, and the proportion of successful estimation procedures, organised by choice of prior.

Error messages do not necessarily imply that the problem is infeasible — it merely means that the optimiser was not able to find a solution, as shown by the fact that in some cases, one method declared the problem infeasible while others did not. It is also important to note that a reported success merely means that a local optimum was found, which may or may not be the global optimum.

The performance depends on the optimisation method as well as on the dataset and the chosen prior. While more extensive data would be required in order to make definite conclusions, some trends can be observed. The most obvious result is that the collocation methods are consistently between one and two orders of magnitude faster than the other methods. The number of iterations is comparable across all methods, suggesting that the performance difference can be attributed to the time required to compute the cost function, the Jacobians, and the Hessians. Hence, this does not necessarily mean that the collocation formulation is inherently easier to solve. This could mean that decreased nonlinearity of the problem, which is one of the main reasons for using multiple shooting and collocation, does not influence the result. In the single shooting and multiple shooting case, the cost function and the constraints are computed by a sophisticated ODE solver, which includes adaptation of step sizes

and model orders, and which needs to solve a set of nonlinear equations at each time step (Serban and Hindmarsh 2005). In contrast, evaluating the cost function and constraints for the collocation methods largely involves evaluating low-order polynomials. While the adaptive ODE solvers ensure that the solution is accurate up to user-specified tolerance, the collocation methods require the user to select the discretisation step size and the order of the polynomial. If poor choices are made, the discretisation error could be large. Alternatively, too fine a discretisation may result in an unnecessarily expensive method. The shooting methods may therefore be a safer option. In this case, the chosen collocation scheme was sufficiently accurate, as the solutions were practically identical to the solutions for single and multiple shooting. The computation time was low enough to make fine-tuning unnecessary.

However, the collocation methods may suffer from a lack of robustness, as they failed in three cases for the test data. In all cases, this happened for densely sampled data, when modelling the input function in the log domain. The single shooting method also failed in a single case, when using real data. However, even though the method indicated an error, the input function estimate turned out to be extremely close to the estimates provided by the other optimisation methods. This could suggest that the problem in this case was numerical, where the method was not able to detect that it had converged to a solution that was practically correct.

In many cases, the optimisation algorithm attempted to evaluate the cost function or its derivatives for input functions that caused the ODE integrator to fail. This is especially likely to happen for penalisation of the second derivative in the log domain, where even a relatively minor change in a coefficient can result in large input function values. Failure of the ODE solver indicates that the proposed input function is not realistic, and should be assigned a negligible probability. It is important that the optimisation software is able to handle these kinds of failures gracefully, such as by treating the proposed coefficients as a high-cost or an infeasible solution. IpOpt does appear to handle these issues well. If other implementations are used, these aspects have to be considered.

Examples of estimated inputs and predicted plasma concentrations for the real data are shown in Fig. 4.9.

The L-curves for the optimal-control examples are shown in Fig. 4.10. The arrows show the values of τ that were chosen. Adjacent values of τ differ by approximately one order of magnitude. Estimation plots for these values are shown in Fig. 4.11. Clearly, τ has a large impact on the resulting functions. The result shows clear signs of oversmoothing.

The estimated bioavailabilities for the real data are shown in Table 4.9.

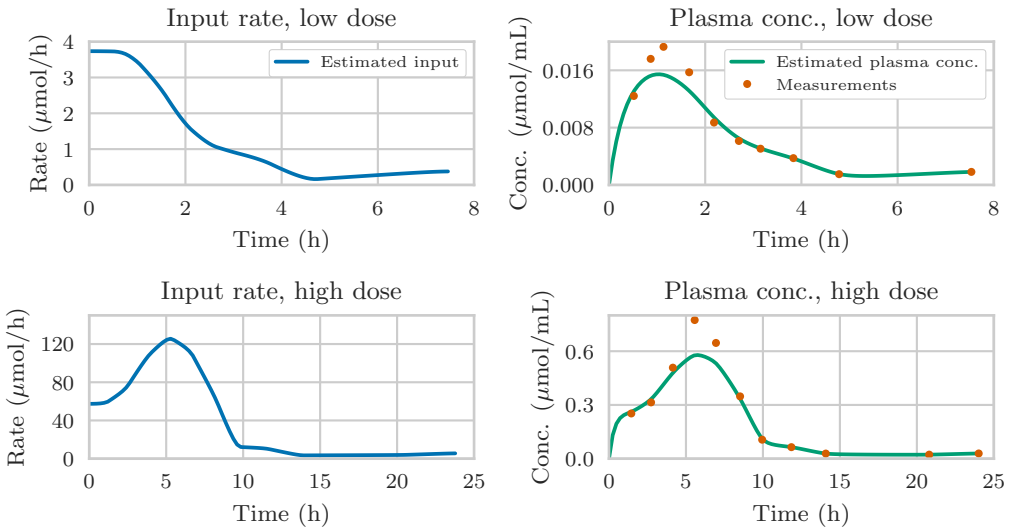

Figure 4.9: Example results from performing optimal-control input estimation on real data, using penalisation of the first derivative in the log domain.

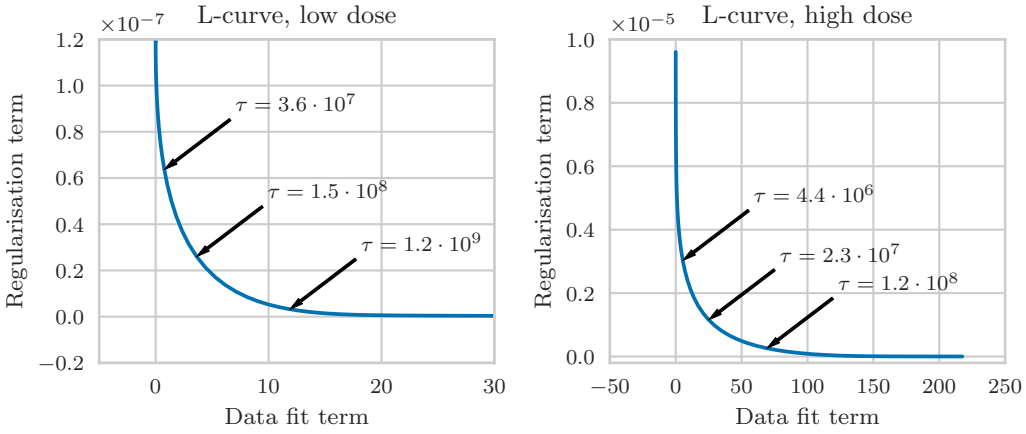

Figure 4.10: L-curves for the two real datasets, with a prior penalising the second derivative. The values of the regularisation parameter τ that were selected for the subsequent analysis are annotated.

The bioavailability estimate appears to be remarkably consistent across priors and estimation methods. The main exception to this is the estimate of the bioavailability of the high dose using the L-curve approach, whose underprediction is consistent with the behaviour shown in Fig. 4.11. The estimated values here are similar to the results obtained by Johansson et al. (2013). It can be noted that even though the

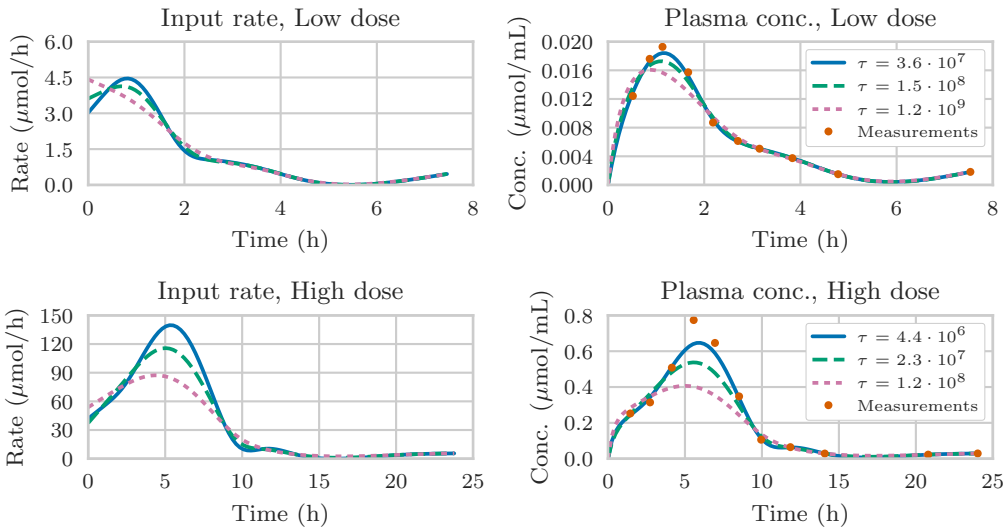

Figure 4.11: Estimation results for the two real datasets, with the regularisation parameter chosen from the L-curve criterion.

Dose	1der	1der, log	2der	2der, log	entropy	L-curve, low τ	L-curve, medium τ	L-curve, high τ
Low	0.32	0.33	0.32	0.33	0.32	0.32	0.32	0.32
High	0.39	0.40	0.38	0.41	0.40	0.40	0.36	0.32

Table 4.9: Estimated bioavailability from the real data, using optimal-control techniques. The first five columns were computed using the discrepancy criterion, while the last three columns were computed using the L-curve criterion, using three different values for the regularisation parameter.

discrepancy criterion is known to result in oversmoothing, the L-curve approach does not appear to work better in this regard in this particular case. These results are not conclusive, as there is a degree of subjectivity in the L-curve approach in that values are chosen manually. Considering the amount of manual work as well as the computational requirements of performing estimation for a large number of values of τ, the L-curve approach does not appear to be competitive.

In terms of statistical soundness, it can be noted that the choice of prior substantially influences the estimates (Fig. 4.8). This makes it important to consider which choices of prior are justified. Penalisation of the second derivative appears to make the function unrealistically smooth. If a large initial peak in the input function is expected, penalisation of the first derivative may therefore be a good option. For the same reason, penalisation in the log-domain may be preferred, as discussed in

Section 2.4.1.

4.5.2 MCMC methods

The performance of the MCMC methods was evaluated on the same criteria as the optimal-control methods. Assessing robustness is considerably more difficult for MCMC methods. Optimal-control methods aim to find a local maximum of the posterior, and the optimisation methods are generally able to detect failures to converge to a maximum. MCMC methods aim to generate a representative set of samples from the posterior. As discussed in Section 3.3.3, it is important to run diagnostics in order to determine the quality of the generated samples, and to estimate the number of samples required to obtain an accurate estimate. However, no diagnostics are able to conclusively confirm that this has been successful. Here, we will assume that methods that pass the Raftery-Lewis diagnostics (Section 3.3.3) were successful.

To assess performance, the following measures were evaluated:

ESS. For each estimation procedure, the ESS was calculated for each parameter, including τ and the basis function coefficients. The performance of the method is limited by the minimum of these values.

Raftery-Lewis. These diagnostics report the number of samples required to achieve the desired accuracy for five quantiles for every parameter. The performance of the method is limited by the maximum of these values.

Required time. This is an estimate of the computing time required, and is given by the maximum number of samples reported by the Raftery-Lewis diagnostics multiplied by the average computing time per sample.

Robustness was assessed by the number of successful runs. A run was considered to be unsuccessful if any of the following two conditions were met:

1. The sampling procedure failed because of numerical errors. In certain cases, MCMC sampling failed when a metric tensor that is not positive definite was encountered. This can happen for block RWMH and MALA, which use the metric tensor at the initial point to construct the proposal, and for SMMALA, where the metric tensor is recomputed at each iteration.

2. The required number of samples was impractically large. Here, a run was considered unsuccessful if the maximum number of samples reported by the Raftery-Lewis diagnostics exceeded the actual number of retained samples.

A large Raftery-Lewis value signifies that the Markov chain was not able to explore the parameter space efficiently. Figure 4.12 shows a comparison between a successful and an unsuccessful result. The unsuccessful example shows strong autocorrelations in the Markov chain, and the shown sample paths appear unrealistic.

All tested combinations of sampling method, function parameterisation, and prior were unsuccessful according to these criteria when applied to assumed noise-free data, regardless of measurement sampling frequency. The tight likelihood function concentrates all probability in a narrow region that is difficult for the methods to navigate. For this reason, many proposals end up in very low probability regions. This is true even for the differential geometric methods, which were invented to cope with difficult distributions. For the remainder of the analysis, only results for the noisy datasets are considered.

Robustness and performance results for the noisy datasets are shown in Table 4.10, reported by sampling method and function parameterisation. The median of each group is shown, as it was considered to be more robust to outliers than the mean. The medians could only be computed for the runs for which it was possible to compute the ESS and Raftery-Lewis values. This excludes all methods that failed because of an invalid metric tensor. Additionally, it was found that both the ESS and the Raftery-Lewis diagnostics could encounter numerical problems when the Markov chain was very strongly autocorrelated, resulting in nonsensical values. This means that the most problematic runs had to be excluded, which could bias the results in favour of methods that have many unsuccessful runs. In practice, however, the reported performance of the methods with many unsuccessful runs compared unfavourably even when the most problematic runs were excluded. For this reason, it is unlikely that the main conclusions will be affected by these exclusions.

		Median min ESS	Median max RL	Median required time	Successful runs
Single RWMH	KL	4	235669	4052	0/16
	spline	6	416390	4484	0/16
Block RWMH	KL	706	55268	80	13/16
	spline	62	417096	466	7/16
MALA	KL	450	8397	103	14/16
	spline	5	410579	4783	3/16
SMMALA	KL	1104	4326	59	16/16
	spline	15	218832	3816	5/16

Table 4.10: Performance and robustness results for MCMC methods on the noisy eflornithine datasets.

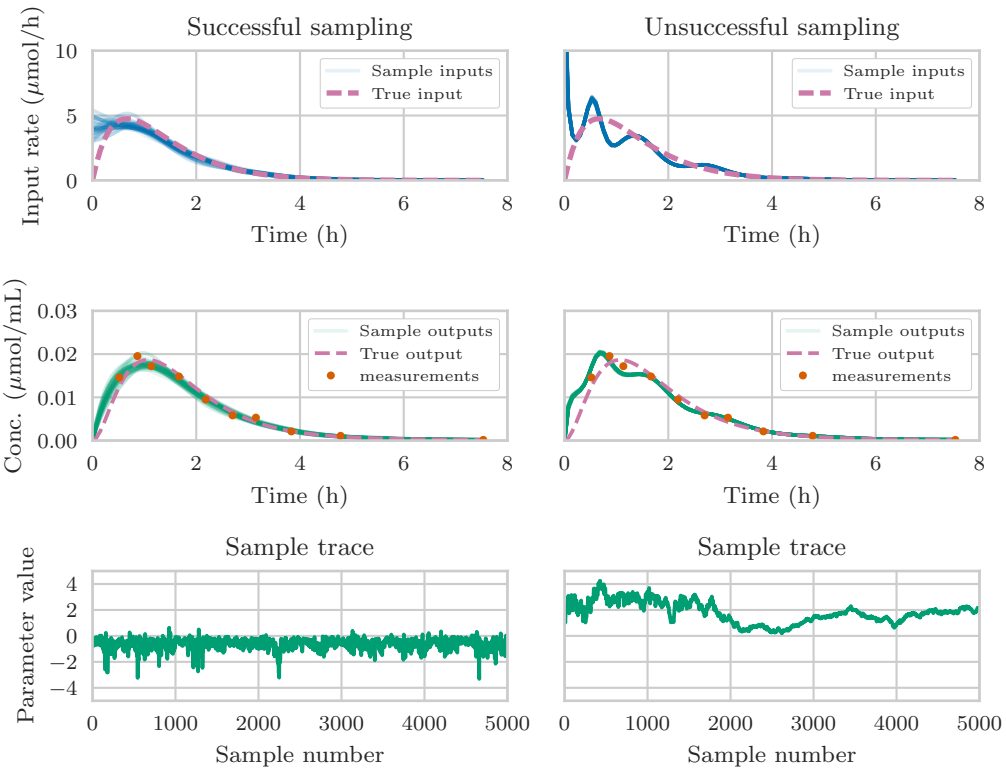

Figure 4.12: Comparison of results for successful and unsuccessful sampling. This is the sparse, noisy test data sampled using SMMALA, penalising the second derivative in the log domain. The left column shows the result when using Karhunen-Loève basis functions, while the right column shows the result for B-splines. The top plots show 50 consecutively sampled input functions, the mid plots show the 50 corresponding sampled output functions, and the bottom plots show the Markov chains for one of the parameters — the 3rd basis function coefficient. It can be seen that the B-spline result is stuck in an implausibly oscillatory function, and its Markov chain exhibits clear signs of strong autocorrelation. Note that the two Markov chains are not expected to have the same mean value, as the function parameterisations differ.

Table 4.11 shows measures of the accuracy of the estimates of the absorption profile and the total absorbed amount for the test data. As the MCMC methods return a large number of samples, the RMSE value shown in the table is the mean RMSE value, averaged over all samples. For the absorbed amount, the mean prediction, and the RMSE computed over all samples are reported. The latter value is an estimate of the standard deviation of the posterior distribution over the absorbed amount. In both cases, they are normalised by the true amount. Unsurprisingly, the densely sampled data allowed for a substantially more accurate estimate than

Dataset	Method	Basis	Prior	RMSE (μmol/h)	Total amount normalised mean	Total amount normalised RMSE
Dense, 10% noise	Block RWMH	KL	1der	0.17	0.98	0.021
	Block RWMH	KL	1der, log	0.25	0.99	0.016
	Block RWMH	KL	2der	0.15	0.99	0.020
	Block RWMH	KL	2der, log	0.21	0.99	0.016
	Block RWMH	spline	1der	0.21	0.99	0.019
	Block RWMH	spline	1der, log	0.26	0.99	0.016
	Block RWMH	spline	2der, log	0.27	1.00	0.016
	MALA	KL	1der	0.17	0.98	0.021
	MALA	KL	1der, log	0.25	0.99	0.016
	MALA	KL	2der	0.15	0.99	0.020
	MALA	KL	2der, log	0.21	1.00	0.016
	MALA	spline	1der	0.20	0.99	0.019
	SMMALA	KL	1der	0.17	0.98	0.021
	SMMALA	KL	1der, log	0.25	0.99	0.017
	SMMALA	KL	2der	0.15	0.99	0.019
	SMMALA	KL	2der, log	0.21	1.00	0.016
	SMMALA	spline	1der	0.20	0.99	0.019
	SMMALA	spline	2der	0.24	0.99	0.017
Sparse 10% noise	Block RWMH	KL	1der	0.77	1.06	0.165
	Block RWMH	KL	1der, log	0.66	1.03	0.056
	Block RWMH	KL	2der	0.77	1.06	0.110
	Block RWMH	KL	2der, log	0.67	1.04	0.054
	Block RWMH	spline	1der	1.52	0.98	0.656
	MALA	KL	1der	0.77	1.05	0.164
	MALA	KL	1der, log	0.68	1.03	0.059
	MALA	KL	2der	0.77	1.06	0.105
	MALA	KL	2der, log	0.65	1.04	0.053
	MALA	spline	1der	1.54	0.97	0.661
	SMMALA	KL	1der	0.76	1.05	0.159
	SMMALA	KL	1der, log	0.65	1.03	0.054
	SMMALA	KL	2der	0.77	1.06	0.111
	SMMALA	KL	2der, log	0.68	1.04	0.054
	SMMALA	spline	1der	1.58	0.98	0.689

Table 4.11: Accuracy measures of the MCMC methods on the noisy eflornithine test datasets.

the sparsely sampled data. For the sparse data, the log-domain priors appear to provide more accurate results. It is clear that spline models are less likely to result in a successful estimation procedure, especially for sparse data. Also note that the

spline models have larger RMSE values. This is an important point, since in the ideal case, the estimate should be determined by the prior and likelihood and not on the parameterisation. The results suggest that in practice the parameterisation can affect the estimate substantially. Additionally, it could be seen that the results are similar across sampling methods for any given combination of data, parameterisation and prior. This increases the confidence that the sampling methods were able to produce a representative set of samples.

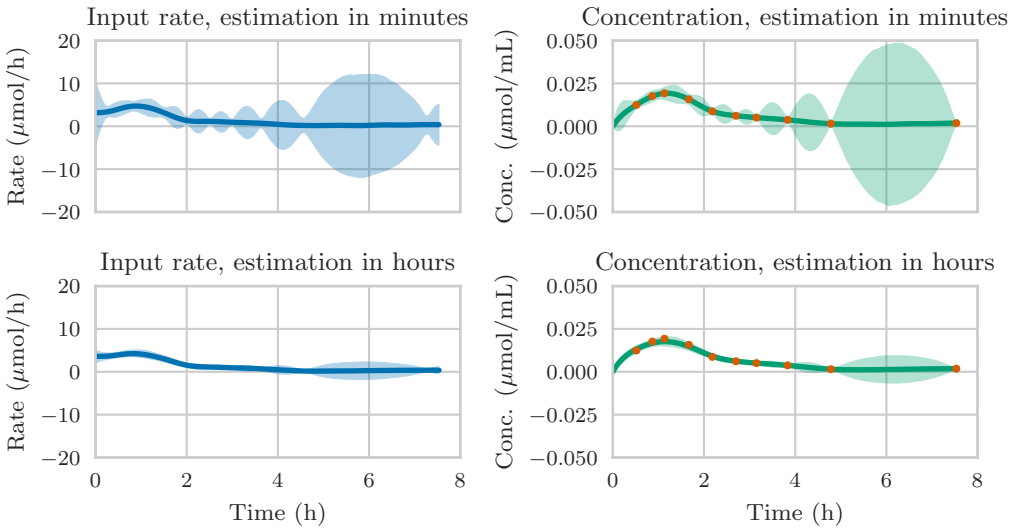

Figure 4.13: An illustration of the effects of letting the prior over τ overpower the data. These estimates were obtained from the low-dose real dataset, using Karhunen-Loève basis functions and penalising the first derivative. The top plots show the results when estimation is performed using minutes as time units, making the sum of squared coefficients smaller than the β parameter of the prior. The lower plots show the result when using time units in hours. Note that in both cases, the result is presented in hours to allow comparisons. The shaded regions show 95% credible intervals. While the mean predictions are similar, the top plots show much higher uncertainty.

Another important point is that the prior over τ needs to be checked to ensure that it does not influence the result in unintended ways. This can happen if the choice of physical units and the parameters of the prior are not well matched. It is evident from the form of the Gamma distribution (Eq. (3.80)) that α is unitless, while the units of β depend on the choice of units for the basis function coefficients. This can also be seen by studying the expressions for the parameters of the posterior distribution over τ. Recall that the β parameter in the posterior distribution is determined by the β parameter in the prior and the sum of squared function coefficients. If the

prior β is set to 10^{-3}, and the sum of squares is significantly smaller than this, then the posterior β will largely be determined by the prior. The parameter α, on the other hand, depends only on the prior α and the number of coefficients, and is therefore independent of the choice of units. Since the expected value of a Gamma distribution is α/β, a β value that is artificially inflated by the prior will drive τ down and decrease the amount of regularisation. This will increase the uncertainty in the estimate. Figure 4.13 shows that the eflornithine model would be vulnerable to this phenomenon if the time units were changed from hours to minutes. Although the mean predictions are similar, the credible intervals are much larger when using minutes. It was confirmed that using a vaguer prior with $\alpha = \beta = 10^{-6}$ resulted in estimates very similar to the estimates in hours. It is possible to rescale the basis function coefficients in order to make them unitless. However, the posterior distribution is still partly determined by the model and the data, and hence the choice of units will still influence the posterior.

Data	Prior	Median min ESS	Median max RL	Successful runs
Low dose	2der	11	296808	0/3
	2der, log	477	8688	3/3
High dose	2der	21	135531	0/3
	2der, log	325	14048	2/3

Table 4.12: Performance results for the procedures using the L-curve approach to estimate the regularisation parameter τ. Each prior and dataset was analysed using three values for τ: one at the "knee" of the L-curve, and one on either side of the "knee".

Table 4.12 shows the performance of the input-estimation procedures when using the L-curve method for determining τ. When log-domain priors are used, this method performs satisfactorily. This can be contrasted to the procedures where τ is estimated, where single-component RWMH performed very poorly on B-splines, only barely succeeding on the densely sampled test dataset, and not succeeding at all on more realistic datasets. Given the small number of tests, it is not possible to determine whether this is a general result. However, it is possible that including τ as a parameter in the estimation problem makes it more difficult for the sampling methods to perform well. Computing time was not reported for the L-curve examples, as these were made using an older version of the estimation software that lacked significant performance enhancements introduced in later versions. Therefore, no meaningful comparison of running time can be made. However, it is reasonable to assume that there is little difference in running time between the L-curve procedures

and the procedures which estimate τ, as the time required to update τ is insignificant compared to the time required to update the input-function coefficients.

The estimated mean bioavailabilities for the real data are shown in Table 4.13, reported by choice of prior and choice of method for estimating τ. Only results which passed the Raftery-Lewis diagnostics are shown. Hence, for the procedures where τ was treated as a parameter, all reported results used Karhunen-Loève basis functions. For the same reason, only log-domain priors are reported for the L-curve procedures. As the bioavailability is a single number, its distribution can be visualised using a one-dimensional histogram or a kernel-density estimate (Scott 2015). This provides more information than the mean values reported in the table. Figure 4.14 shows kernel-density estimates for the procedures where τ was estimated, corresponding to the first four columns of Table 4.13. In all cases, the estimates were obtained by using the samples from the sampling method that reported the lowest number of required samples according to the Raftery-Lewis diagnostics.

Dose	1der	1der, log	2der	2der, log	L-curve, low τ	L-curve, medium τ	L-curve, high τ
Low	0.34	0.34	0.32	0.33	0.33	0.33	0.33
High	0.42	0.43	0.41	0.42	0.42	0.41	0.38

Table 4.13: Mean estimated bioavailability for the real eflornithine datasets using MCMC. For the L-curve examples, only the log-domain results are shown, as the other L-curve procedures were not successful. All of the other shown results were obtained using Karhunen-Loève basis functions and the most successful sampling method, which was defined to be the method that required the lowest number of samples according to the Raftery-Lewis diagnostics.

Figure 4.15 shows estimation results for all choices of priors on the sparse noisy test dataset, using SMMALA and Karhunen-Loève basis functions. It can be noted that for the priors that do not enforce nonnegativity constraints, the uncertainty regions are large and can take on negative values in the later part of the curves. The reason for this is that when negative values are allowed, the estimation method is able to find a large range of trajectories that are consistent with the data, as sampling is very sparse in this region. Log-domain priors, which exclude negative solutions, help in making the uncertainty region smaller, while at the same time providing more plausible-looking results. When evaluating the methods in terms of statistical soundness, log-domain priors are therefore easier to justify. As in the case of optimal-control methods, penalisation of the second derivative appears to cause oversmoothing. Since a large initial peak can often be expected in PK models,

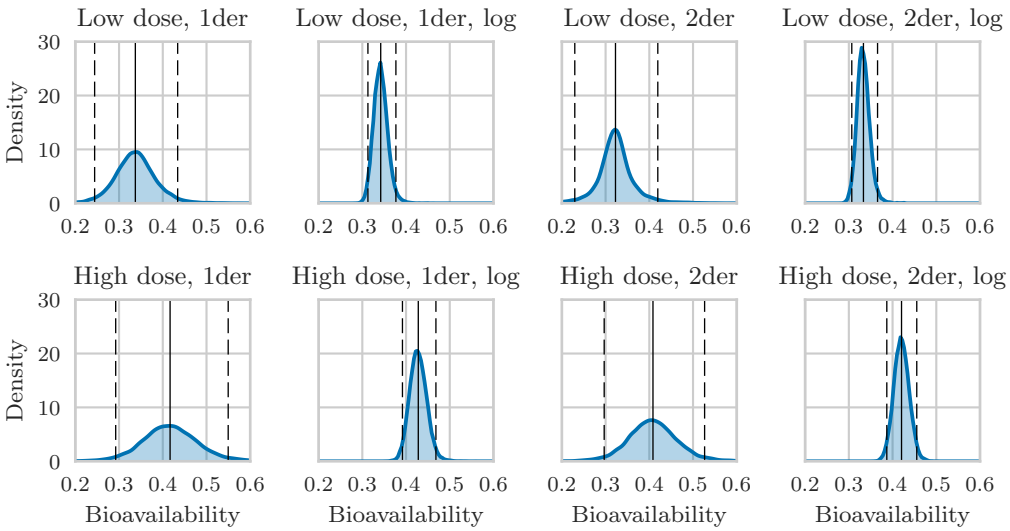

Figure 4.14: Kernel-density estimates of the bioavailability of the real eflornithine datasets, obtained from the MCMC analyses. In the displayed plots, estimation was performed using Karhunen-Loève basis functions, using the sampling method that provided the best performance. The solid lines denote means, and the dashed lines show the 2.5th and 97.5th percentiles.

this provides a reason for preferring the more conservative penalisation of the first derivative.

Because of the issues with priors that allow negative solutions, only log-domain results will be considered from here on. For sparsely sampled data with log-domain priors, estimation was consistently successful when block RWMH, MALA or SMMALA was used as the sampling method, and Karhunen-Loève functions were used as a basis. The computational speed is similar for all three methods.

Figure 4.16 shows example plots of successful estimation procedures for the noisy test data and the real data, using penalisation of the second derivative in the log domain and Karhunen-Loève basis functions. The test data plots also show the true functions. The dense noisy data have caused the estimate to show a double-peak that is not present in the true functions. Otherwise, the true input is recovered relatively accurately, and the initial input rate increase is followed closely. For sparse data, the initial rise is not captured at all. The concentration plot shows that the data can easily be explained without the initial rise. The prior, which discourages unnecessary oscillation, makes the procedure prefer the smoother estimate. As in the optimal-control case, it would be possible to constrain the estimated input to start at 0. For the Karhunen-Loève basis, this could be achieved by setting the offset

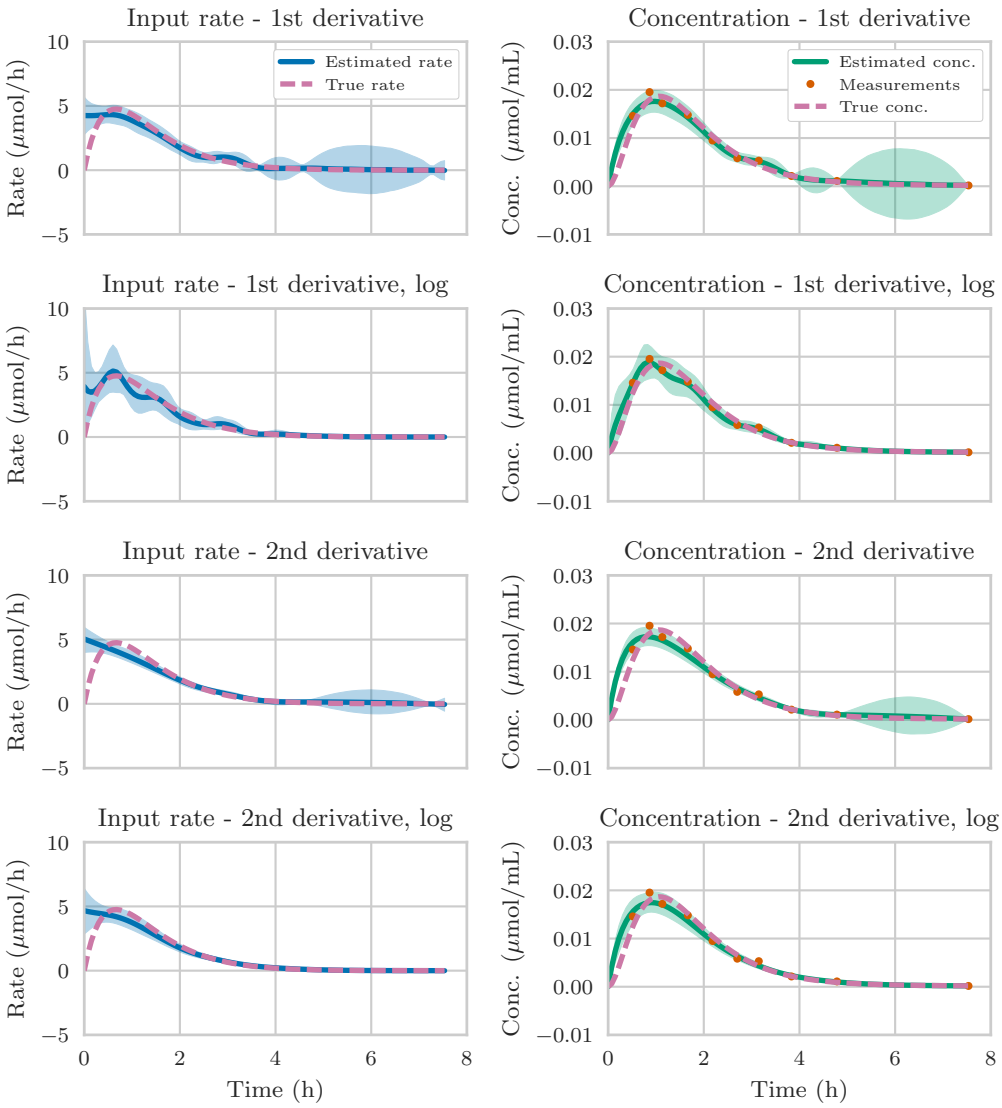

Figure 4.15: Examples of estimation results for the sparse noisy test data for various priors using Karhunen-Loève basis functions. All shown results were obtained by SMMALA sampling.

coefficient to 0, rather than treating it as an unknown parameter.

4.6 Summary

This chapter evaluated optimal-control and MCMC methods on the eflornithine dataset. For the optimal-control methods, underprediction of the initial peak is

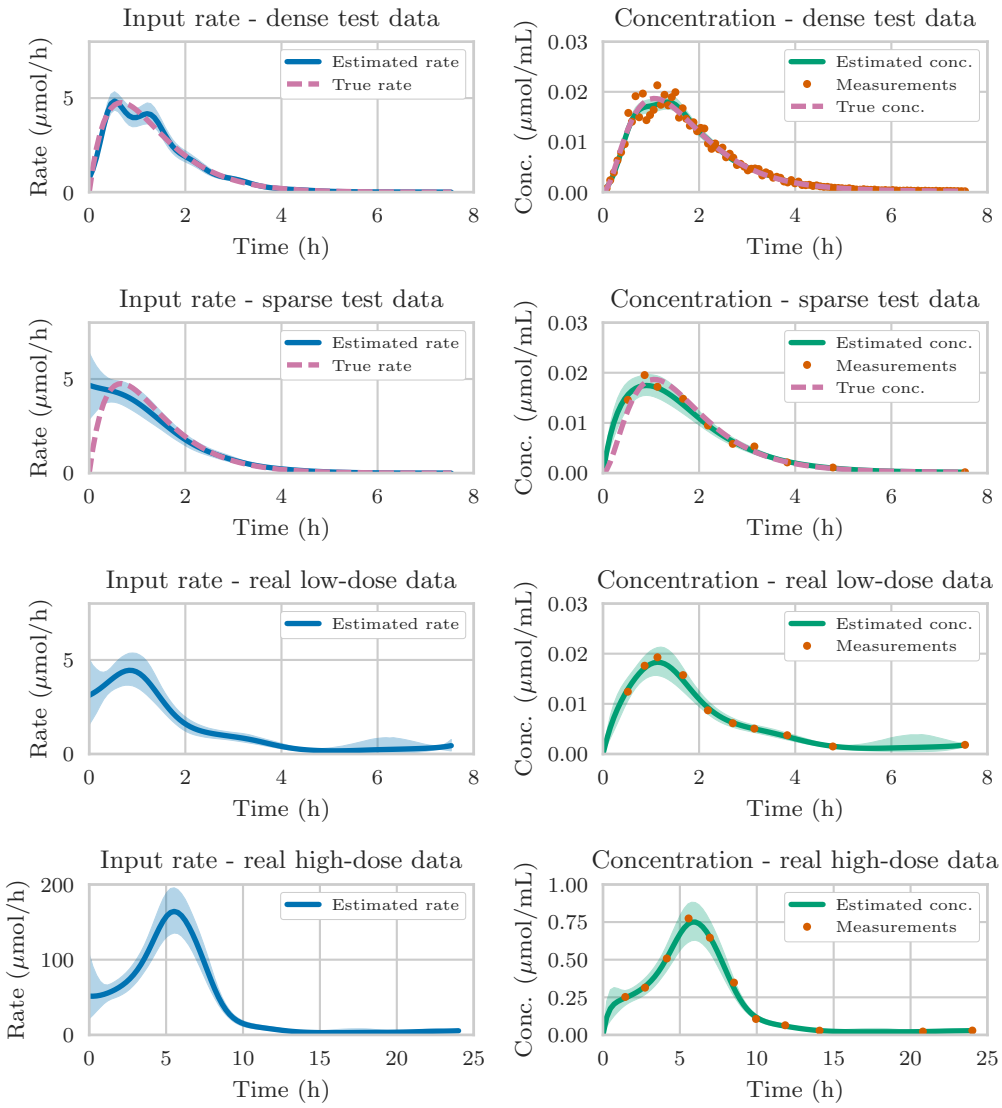

Figure 4.16: Estimates for the noisy test and real datasets, using penalisation of the second derivative in the log domain. Note that the last dataset has a different scale, since rates and concentrations are considerably higher for this dose, and measurements were taken over a longer time. These estimates were performed using MALA. The results for block RWMH and SMMALA are practically identical.

a common problem. However, this is difficult to avoid without making stronger modelling assumptions. In this and other PK examples, it may be reasonable to assume that the input function is 0 at time 0, which may help in capturing the peak. If it is important to accurately estimate the initial dynamics without additional assumptions, it appears necessary to have a higher initial sampling rate. However,

when the bioavailability is the main quantity of interest, the nonzero initial value and the lower peak partly offset each other, making the estimated bioavailability insensitive to these assumptions.

It is important to have an optimisation algorithm that handles evaluation errors gracefully. Such errors can occur when the optimiser attempts to evaluate the cost function for values that cause the ODE solver to fail. The optimiser used here, IpOpt, does perform well in this regard.

Collocation methods are one order of magnitude faster than the other methods, but they can have robustness issues in some situations, resulting in convergence failures. However, these failures can be automatically detected. Thus, one possible strategy is to first attempt to use collocation methods, and to use shooting methods as a fallback in case the collocation methods fail. For sparse data, which is the usual real-world case, multiple shooting is generally faster than single shooting.

While the shooting methods use an ODE solver that automatically selects suitable integration options, taking stiffness issues into account, the collocation methods as used here require the user to choose interpolating polynomials and discretisation step size. For future work, it would be useful to investigate how these choices affect the robustness. Better settings may potentially make the collocation methods more widely applicable.

The choice of prior greatly influences the result, although it is far from obvious which one should be preferred. The accuracy in terms of RMSE values is similar. Figure 4.8 suggests that penalisation of the first derivative in the log domain may be more likely to produce a peak similar to the one present in the true input function. However, these kinds of considerations are subjective.

The example of using the L-curve method to choose a value of τ shows that this method offers no clear advantage to the discrepancy criterion. Since this method requires the user to do a lot of work, it cannot be recommended in the form presented here.

All MCMC methods struggle with data that have almost zero measurement noise. When the data are noisy, as they usually are in real data, the block RWMH, MALA and SMMALA methods combined with Karhunen-Loève basis functions are generally able to sample from the posterior reasonably efficiently. This conclusion is strengthened by the observation that both sampling methods provide virtually identical results for the same prior and data. While the SMMALA compares favourably to MALA, in terms of median running time, the large variability of the results makes it difficult to make any definite conclusions. Since the proposal covariance matrix in block RWMH and the metric tensor in MALA are fixed from the start, it is possible

that they could be more sensitive to their initial conditions than SMMALA. For this reason, using SMMALA may be advisable in order to have enhanced robustness. The required running time per sample is similar for MALA and SMMALA. This may seem surprising, as SMMALA recomputes the metric tensor in each iteration. However, the most expensive step in the metric tensor computation is to compute the gradients of the plasma concentration predictions at the measurement time points. This step is also required in MALA in order to compute the gradient of the log-posterior. Total running time depends on the time per sample as well as the total number of samples required, which depends on the sampling method's ability to generate approximately uncorrelated samples.

For a given amount of time, the block RWMH method generated approximately one order of magnitude more samples than MALA or SMMALA. The main reason for this is that RWMH does not require gradient computations, which can be a computational bottleneck. Despite this, the ESS is similar for all three methods. This shows that the gradient-based methods do produce higher-quality samples, with lower autocorrelations in the Markov chains, at the expense of requiring more computation per sample.

In many cases, lack of nonnegativity constraints results in implausible results, and can also unnecessarily increase uncertainty. This is particularly pronounced in the sparsely sampled later parts of the time series, where concentrations are close to zero. With no constraints in place, there is a large number of candidate functions that could explain the data. When nonnegativity constraints are added, only functions that stay close to 0 can explain the data. For this reason, modelling the input function in the log domain should be preferred.

While the parameterisation should ideally not affect the results, in this case it did. Few sampling procedures involving B-splines were successful. However, for the ones that were successful, estimation errors were higher than for the corresponding procedures using Karhunen-Loève functions.

When starting with a new model and dataset, it is important to investigate typical values of the sum of squared residuals, so that the prior over τ does not influence the estimate in ways that were not intended. A poor choice can cause τ to be underestimated, allowing sample paths to make large excursions between measurements and causing the estimation uncertainty to increase.

Computing the likelihood, and especially the gradient of the likelihood, is expensive as it includes running an ODE solver. For this reason, methods that update all parameters influencing the likelihood simultaneously have an advantage in terms of speed. The single-component RWMH methods are much slower than block RWMH

for this reason — to update 20 coefficients, the ODE solver needs to run 20 times. In contrast, updating τ independently using Gibbs sampling is practically free, as updating τ does not require updating the likelihood.

Finally, it is important to stress that this should be considered to be an exploratory study. Many combinations of optimisation or sampling methods, priors, and function parameterisations are applied to a relatively small dataset. This allows for a multitude of possible comparisons, resulting in multiple hypothesis testing issues. For this reason, only major results are reported here. More extensive tests would be required to conclusively tell whether multiple shooting is generally faster than single shooting, or whether MALA is faster than SMMALA. Testing the methods on other problems, as done in the later chapters, helps increasing the confidence in the conclusions.

Chapter 5

Case study — body-weight modelling

5.1 Background

Obesity is a major problem worldwide. As of 2014, there were more than 600 million obese adults (BMI > 30), amounting to 13% of the total adult population (World Health Organization 2016). Obesity is linked to serious diseases such as diabetes, hypertension, cardiovascular disease, and premature death (Kumanyika et al. 2002; James 2008). Lifestyle changes, including reduced energy intake and increased exercise, are the main treatments. Additionally, drug therapy can be used. The drugs that are or have been available on the market usually act either by suppressing appetite, or by inhibiting uptake of fat in the intestine. Several of these drugs can have serious side effects, including cardiovascular diseases and psychiatric disorders (Kang and Park 2012). Hence, there is a market for developing safer drugs.

During trials of anti-obesity drugs, it is of interest to determine changes in energy intake and body weight. Measuring energy intake is possible, but can be expensive, especially for long-running studies (Göbel et al. 2014). In contrast, measuring body weight is cheap. One potential solution is to use models of body weight and body composition to infer the energy intake from body-weight measurements, rather than measuring the energy intake directly. Inferring the energy intake can be treated as an input-estimation problem. This assumes that the drug directly affects only the energy intake, and that any changes in energy expenditure are indirect effects that are accounted for in the body-weight models. Depending on the drug's mechanism of action, this may or may not be a plausible assumption.

In this chapter, input-estimation methods are evaluated on two body-weight

models: a simple two-state mouse model, and a more advanced five-state human model. The mouse body-weight model (Guo and Hall 2009; Guo and Hall 2011) was applied on a dataset where diet-induced obese (DIO) mice were administered two versions of a monoclonal antibody (R1c mAb opt1 and R1c mAb opt2) that targets the fibroblast growth factor receptor (FGFR) 1c. These are modifications of a parent antibody R1c mAb, which has previously been shown to cause body-weight loss in DIO mice (Lelliott et al. 2014), solely due to a decrease in energy intake. Other mAbs targeting FGFR1 or FGFR1c have also been shown to induce body-weight loss, either due to an effect only on energy intake, or due to a combined effect on energy intake and energy expenditure (Sun et al. 2007; Wu et al. 2011). Data for body weight as well as energy intake are available from the experiment. The aim of the analysis presented here is to investigate whether the input-estimation methods can recover the energy intake from body-weight measurements alone.

For the human body-weight model (Hall et al. 2011), 14 time series from 11 studies were used, with each time series describing the mean body-weight of a patient group. Descriptions of the drugs used in these studies can be found in Padwal and Majumdar (2007) and Kang and Park (2012). No energy-intake data are available. The original reason for compiling data from these studies was for use in Gennemark et al. (2017), which discusses strategies on using body-weight models together with biomarker data to aid study design and dose prediction of anti-obesity drugs in drug discovery.

5.2 Models

The two-state mouse model, describing the relationship between energy intake and body weight, is a semi-mechanistic model described in Guo and Hall (2009) and Guo and Hall (2011). The version of the model which is used in this work is reported by Gennemark et al. (2013). The model divides the total body mass into fat mass (FM) and fat-free mass (FFM). Mass balance requires the dynamics to satisfy

$$\rho_{FM} \cdot \frac{\mathrm{d}FM}{\mathrm{d}t} + \rho_{FFM} \cdot \frac{\mathrm{d}FFM}{\mathrm{d}t} = EI - EE \tag{5.1}$$

where ρ_{FM} and ρ_{FFM} is the energy density of fat mass and fat-free mass, EI is the energy intake and EE is the energy expenditure. Furthermore, FM and FFM are assumed to be related by the *Forbes curve*

$$\alpha = \frac{\mathrm{d}FFM}{\mathrm{d}FM} = q_1 + q_2 \cdot e^{q_3 \cdot FM} \tag{5.2}$$

where q_1, q_2, and q_3 are empirically determined constants. The energy expenditure EE is modelled as

$$\begin{aligned} EE =& K + \beta \Delta EI + (\gamma_{FFM} + \lambda) \cdot FFM + (\gamma_{FM} + \lambda) \cdot FM + \\ & \eta_{FFM} \cdot \frac{\mathrm{d}FFM}{\mathrm{d}t} + \eta_{FM} \cdot \frac{\mathrm{d}FM}{\mathrm{d}t} \end{aligned} \tag{5.3}$$

where K is a constant thermogenesis parameter, ΔEI is the change in energy intake from the baseline value EI_{stand}, β is a proportionality constant, λ is a physical activity parameter, γ_{FFM} and γ_{FM} are constants relating metabolic rate to body weight, and η_{FFM} and η_{FM} are the synthesis efficiencies of fat-free and fat mass. Table 5.1 shows the model parameter values, which are identical to the values used by Gennemark et al. (2013). The physical activity parameter λ, which is not to be confused with the eigenvalues of the covariance integral operator, was modelled as a time-varying quantity described by

$$\lambda = \begin{cases} \lambda_0 + \lambda_3 + \lambda_1 \cdot \lambda_2 \cdot t \cdot e^{-\lambda_2 (t - t_{treat})} & \text{if } (t - t_{treat}) \geq 0 \\ \lambda_0 & \text{otherwise,} \end{cases} \tag{5.4}$$

where λ_j, $j \in \{0, 1, 2, 3\}$ are empirical values estimated from the data, and t_{treat} is the time at the start of the treatment. Since the analysis is based on the assumption that the drug effect is acting only on the energy intake, these parameters should ideally be dose-independent. However, an inspection of the available energy-intake and body-weight data clearly showed that it was necessary to account for a drug-dependent effect on the physical activity that was not included in the model structure. Rather than explicitly modelling this dependence, values of λ_j were estimated separately for each dose group. First, a cubic smoothing spline was fitted to the energy-intake data. The λ_j parameters were then fitted to the body-weight data when using the fitted spline as input, keeping all other model parameters constant. Obviously, this requires the use of the very same energy intake data that are to be estimated, and which would not be available in the typical use-case. These methods would be more useful in situations where the drugs can be assumed to only affect energy intake. However, since the main purpose of this work is to evaluate input-estimation methods on known dynamic models, this procedure was deemed acceptable. The accuracy of the procedure is mainly evaluated on synthetic data, where the input function is not related to the change in physical activity. The fitted physical activity parameters are shown in Table 5.2.

The model equations can be rearranged to standard ODE form:

Parameter	Definition	Value	Unit
ρ_{FFM}	Energy density, fat-free mass	1.8	$\text{kcal} \cdot \text{g}^{-1}$
ρ_{FM}	Energy density, fat mass	9.4	$\text{kcal} \cdot \text{g}^{-1}$
γ_{FFM}	Metabolic rate, fat-free mass	0.15	$\text{kcal} \cdot \text{g}^{-1} \cdot \text{day}^{-1}$
γ_{FM}	Metabolic rate, fat mass	0.03	$\text{kcal} \cdot \text{g}^{-1} \cdot \text{day}^{-1}$
η_{FFM}	Synthesis efficiency, fat-free mass	0.23	$\text{kcal} \cdot \text{g}^{-1}$
η_{FM}	Synthesis efficiency, fat mass	0.18	$\text{kcal} \cdot \text{g}^{-1}$
K	Thermogenesis rate	2.1	$\text{kcal} \cdot \text{day}^{-1}$
β	Scaling factor, diet-induced thermogenesis	0.4	-
EI_{stand}	Standard energy intake	12	$\text{kcal} \cdot \text{day}^{-1}$
q_1	Forbes curve coefficient 1	0.13	-
q_2	Forbes curve coefficient 2	0.02	-
q_3	Forbes curve coefficient 3	0.09	g^{-1}

Table 5.1: Model parameters for the mouse body-weight model.

$$\frac{\mathrm{d}FFM}{\mathrm{d}t} = \frac{\alpha}{\alpha \cdot \rho_{FFM} + \rho_{FM}} (EI - EE) \tag{5.5a}$$

$$\frac{\mathrm{d}FM}{\mathrm{d}t} = \frac{1}{\alpha \cdot \rho_{FFM} + \rho_{FM}} (EI - EE), \tag{5.5b}$$

where the energy intake is given by

$$\begin{aligned} EE = \quad & (K + \beta \Delta EI + (\gamma_{FFM} + \lambda) \cdot FFM + (\gamma_{FM} + \lambda) \cdot FM \\ & + \eta_{FFM} \cdot \alpha \cdot g \cdot EI + \eta_{FM} \cdot g \cdot EI) / \\ & (1 + \eta_{FM} \cdot g + \eta_{FFM} \cdot \alpha \cdot g) \end{aligned} \tag{5.6}$$

where $g = 1/(\alpha \cdot \rho_{FFM} + \rho_{FM})$. The body-weight measurement error was assumed to be 0.5% proportional Gaussian noise. This was considered to be reasonable, as weight can be measured with good precision, and the reported body weights are averages.

The initial state of the ODE system could in principle have been included in the estimation problem as extra parameters. However, for simplicity, these were kept fixed. The total body mass was assumed to be equal to the first measurement. The initial fat mass in TM_{init} was assumed to be a linear function of the total mass TM_{init}:

$$FM_{init} = 0.81 \cdot TM_{init} - 12.2, \tag{5.7}$$

Group	Parameter	Value	Unit
Vehicle	λ_0	0.16	$\text{kcal} \cdot \text{day}^{-1}$
	λ_1	0.18	$\text{kcal} \cdot \text{day}^{-1}$
	λ_2	0.57	day^{-1}
	λ_3	-0.068	$\text{kcal} \cdot \text{day}^{-1}$
R1c mAb opt1 (0.3 mg/kg)	λ_0	0.15	$\text{kcal} \cdot \text{day}^{-1}$
	λ_1	0.25	$\text{kcal} \cdot \text{day}^{-1}$
	λ_2	0.95	day^{-1}
	λ_3	-0.051	$\text{kcal} \cdot \text{day}^{-1}$
R1c mAb opt1 (3 mg/kg)	λ_0	0.15	$\text{kcal} \cdot \text{day}^{-1}$
	λ_1	0.54	$\text{kcal} \cdot \text{day}^{-1}$
	λ_2	0.59	day^{-1}
	λ_3	-0.066	$\text{kcal} \cdot \text{day}^{-1}$
R1c mAb opt1 (10 mg/kg)	λ_0	0.14	$\text{kcal} \cdot \text{day}^{-1}$
	λ_1	0.52	$\text{kcal} \cdot \text{day}^{-1}$
	λ_2	0.27	day^{-1}
	λ_3	-0.063	$\text{kcal} \cdot \text{day}^{-1}$
R1c mAb opt2 (0.3 mg/kg)	λ_0	0.13	$\text{kcal} \cdot \text{day}^{-1}$
	λ_1	0.21	$\text{kcal} \cdot \text{day}^{-1}$
	λ_2	0.94	day^{-1}
	λ_3	-0.029	$\text{kcal} \cdot \text{day}^{-1}$
R1c mAb opt2 (3 mg/kg)	λ_0	0.14	$\text{kcal} \cdot \text{day}^{-1}$
	λ_1	0.33	$\text{kcal} \cdot \text{day}^{-1}$
	λ_2	0.69	day^{-1}
	λ_3	-0.044	$\text{kcal} \cdot \text{day}^{-1}$
R1c mAb opt2 (10 mg/kg)	λ_0	0.16	$\text{kcal} \cdot \text{day}^{-1}$
	λ_1	0.59	$\text{kcal} \cdot \text{day}^{-1}$
	λ_2	0.44	day^{-1}
	λ_3	-0.071	$\text{kcal} \cdot \text{day}^{-1}$

Table 5.2: Physical activity parameters for all of the mouse body-weight time series.

where units are in grams. This relationship was established empirically by Gennemark et al. (2013, Fig. 4).

The human body-weight model from Hall et al. (2011) is considerably more elaborate than the mouse model. The model has five state variables: glycogen mass G, extracellular fluid volume ECF, fat mass F, lean mass L, and adaptive thermogenesis AT.

The system dynamics are given by

$$\frac{\mathrm{d}G}{\mathrm{d}t} = \frac{1}{\rho_G}(CI - k_G \cdot G^2) \tag{5.8a}$$

$$\frac{\mathrm{d}ECF}{\mathrm{d}t} = \frac{1}{[Na]}\left(\Delta Na_{diet} - \xi_{Na}(ECF - ECF_{init}) - \xi_{CI}\left(1 - \frac{CI}{CI_{init}}\right)\right) \tag{5.8b}$$

$$\frac{\mathrm{d}F}{\mathrm{d}t} = \frac{1}{\rho_F}(1-p)\cdot(EI - EE - \rho_G\frac{\mathrm{d}G}{\mathrm{d}t}) \tag{5.8c}$$

$$\frac{\mathrm{d}L}{\mathrm{d}t} = \frac{1}{\rho_L}p\cdot(EI - EE - \rho_G\frac{\mathrm{d}G}{\mathrm{d}t}) \tag{5.8d}$$

$$\frac{\mathrm{d}AT}{\mathrm{d}t} = \frac{1}{\tau_{AT}}(\beta_{AT}(EI - EI_{init}) - AT). \tag{5.8e}$$

Equation (5.8a) describes the dynamics of the glycogen mass, where ρ_G is the energy density of glycogen and k_G is a parameter selected to ensure a given steady-state baseline value of glycogen mass. CI is the carbohydrate intake which is assumed to be a constant fraction of total energy intake with proportionality constant f_{CI}. In the extracellular fluid dynamics, $[Na]$ is the extracellular sodium concentration, ECF_{init} is the baseline extracellular fluid volume, ΔNa_{diet} is the change from baseline of the dietary sodium intake, and ξ_{Na} and ξ_{CI} describe the effect of the carbohydrate intake on renal sodium excretion. CI_{init} is the baseline carbohydrate intake. The variables ρ_F and ρ_L are the energy density of fat and lean tissue, and EE is the energy expenditure. The parameter p denotes the partition between fat and lean tissue, and is given by $p = C/(T+F)$, where $C = k_C \cdot \rho_L/\rho_F$ and k_C is a proportionality constant. The state AT describes energy intake-dependent adaptive thermogenesis, where τ_{AT} is the time scale of adaptation, β_{AT} is a proportionality constant, and EI_{init} is the baseline energy intake.

The total body weight, BW, which is the quantity that is measured, is given by

$$\rho_{ECF}\cdot ECF + F + L + (1 + 2.7)\cdot G, \tag{5.9}$$

where ρ_{ECF} is the density of the extracellular fluid, assumed to be equal to the density of water. The factor $(1 + 2.7)$ is included as each gram of glycogen stores 2.7 g of water. The energy expenditure EE is given by

$$EE = K + \gamma_F F + \gamma_L L + \delta \cdot BW + TEF + AT + \eta_L\frac{\mathrm{d}L}{\mathrm{d}t} + \eta_F\frac{\mathrm{d}F}{\mathrm{d}t} \tag{5.10}$$

where K is a constant which is determined on a per-dataset basis to ensure that the system is initially in steady state, γ_F and γ_L are regression coefficients relating energy expenditure with fat and lean mass, and η_L and η_F are proportionality constants relating energy expenditure with fat and protein synthesis. TEF is the thermic effect of feeding, $TEF = \beta_{TEF}(EI - EI_{init})$. The variable δ is a constant physical activity parameter, given by

$$\delta = ((1 - \beta_{TEF}) \cdot PAL - 1)\frac{RMR_{init}}{BW_{init}} \tag{5.11}$$

where PAL is the physical activity level and RMR_{init} is the initial resting metabolic rate.

Parameter	Definition	Value	Unit
ρ_G	Glycogen energy density	17.6	$\text{MJ} \cdot \text{kg}^{-1}$
ρ_F	Fat tissue energy density	39.5	$\text{MJ} \cdot \text{kg}^{-1}$
ρ_L	Lean tissue energy density	7.6	$\text{MJ} \cdot \text{kg}^{-1}$
$[Na]$	Extracellular sodium concentration	3.22	$\text{g} \cdot \text{L}^{-1}$
ξ_{Na}	Carbohydrate intake effect on renal Na secretion	3	$\text{g} \cdot \text{L}^{-1} \cdot \text{day}^{-1}$
ξ_{CI}	Carbohydrate intake effect on renal Na secretion	4	$\text{g} \cdot \text{day}^{-1}$
γ_F	Energy expenditure vs fat mass regression coefficient	0.031	$\text{MJ} \cdot \text{kg}^{-1} \cdot \text{day}^{-1}$
γ_L	Energy expenditure vs lean mass regression coefficient	0.092	$\text{MJ} \cdot \text{kg}^{-1} \cdot \text{day}^{-1}$
η_F	Efficiency of fat synthesis	0.75	$\text{MJ} \cdot \text{kg}^{-1}$
η_L	Efficiency of protein synthesis	0.96	$\text{MJ} \cdot \text{kg}^{-1}$
β_{TEF}	Thermic effect of feeding	0.1	-
β_{AT}	Adaptive thermogenesis scaling coefficient	0.14	-
t_{AT}	Adaptive thermogenesis time constant	14	day
k_C	Partition function coefficient	10.4	kg
ΔNa_{diet}	Change of dietary sodium intake	0	$\text{g} \cdot \text{day}^{-1}$
PAL	Physical activity level	1.5	-
f_{CI}	Fraction of carbohydrate intake	0.5	-

Table 5.3: Model parameters for the human body-weight model. The upper part of the table contains parameters obtained from Hall et al. (2011). The choice of values for the parameters in the lower part is discussed in Section 5.2. Parameters not shown here were determined on a per-dataset basis.

The model parameters are shown in Table 5.3. The majority of the parameters were obtained from Hall et al. (2011). As ΔNa_{diet} was unknown, it was assumed to

have a negligible effect and was set to zero. The physical activity level PAL was set to 1.5, which corresponds to a sedentary lifestyle (Hall et al. 2011), and f_{CI} was set to 0.5, which is a typical average value for obese individuals (Austin et al. 2011). The initial states were determined per dataset, as the patient populations had different baseline body weights and body-mass indices (BMI). The initial glycogen mass G_{init} was set to a baseline value of 0.5 kg (Hall et al. 2011). The initial ECF was obtained using the regression equations of Silva et al. (2007):

$$ECF_{init} = \begin{cases} 0.025 \cdot age + 9.57 \cdot H + 0.191 \cdot BW_{init} - 12.4 & \text{for males} \\ 5.98 \cdot H + 0.167 \cdot BW_{init} - 4 & \text{for females} \end{cases} \tag{5.12}$$

where age is in years, and the height H is in metres. All studies used reported the mean baseline body weight BW_{init} and BMI, from which H could be obtained. The Jackson regression equations (Jackson et al. 2002) were used to determine the initial fat mass:

$$F_{init} = \begin{cases} \frac{BW}{100}\left(0.14 \cdot age + 37.31 \cdot \log(BW_{init}/H^2) - 103.94\right) & \text{for males} \\ \frac{BW}{100}\left(0.14 \cdot age + 39.96 \cdot \log(BW_{init}/H^2) - 102.01\right) & \text{for females.} \end{cases} \tag{5.13}$$

The Mifflin-St Jeor regression equations (Mifflin et al. 1990) provided the initial resting metabolic rate RMR_{init} in kcal/day for males and females:

$$RMR_{init} = \begin{cases} 9.99 \cdot BW_{init} + 625 \cdot H - 4.92 \cdot age + 5 & \text{for males} \\ 9.99 \cdot BW_{init} + 625 \cdot H - 4.92 \cdot age - 161 & \text{for females.} \end{cases} \tag{5.14}$$

While the patient populations were of mixed sex and ages, the regression equations for female 40-year-olds were used for simplicity. The initial lean mass L_{init} was given by $BW_{init} - (1 + 2.7) \cdot G_{init} - \rho_{ECF} \cdot ECF_{init} - F_{init}$, and the initial value of AT was set to 0, as this state accounts for changes from baseline. For each dataset, values for the parameters k_G and K were chosen to ensure that the baseline is in steady state.

5.3 Data

5.3.1 Real data

The data from the mouse study have previously been published in Trägårdh et al. (2016). The study consisted of seven groups of four DIO mice each: one vehicle group and three dose groups each for opt1 and opt2, with doses of 0.3, 3 and 10 mg/kg. The drug was administered as a single subcutaneous injection at $t = 0$. The body weight and energy intake were measured during a period of 30 days after treatment. Additionally, body-weight measurements were performed 9 days and 1 day prior to the treatment. All data analysis was performed on the group means of the measurements. The energy-intake and body-weight data are shown in Fig. 5.1. The body-weight data are also shown in Table A.2.

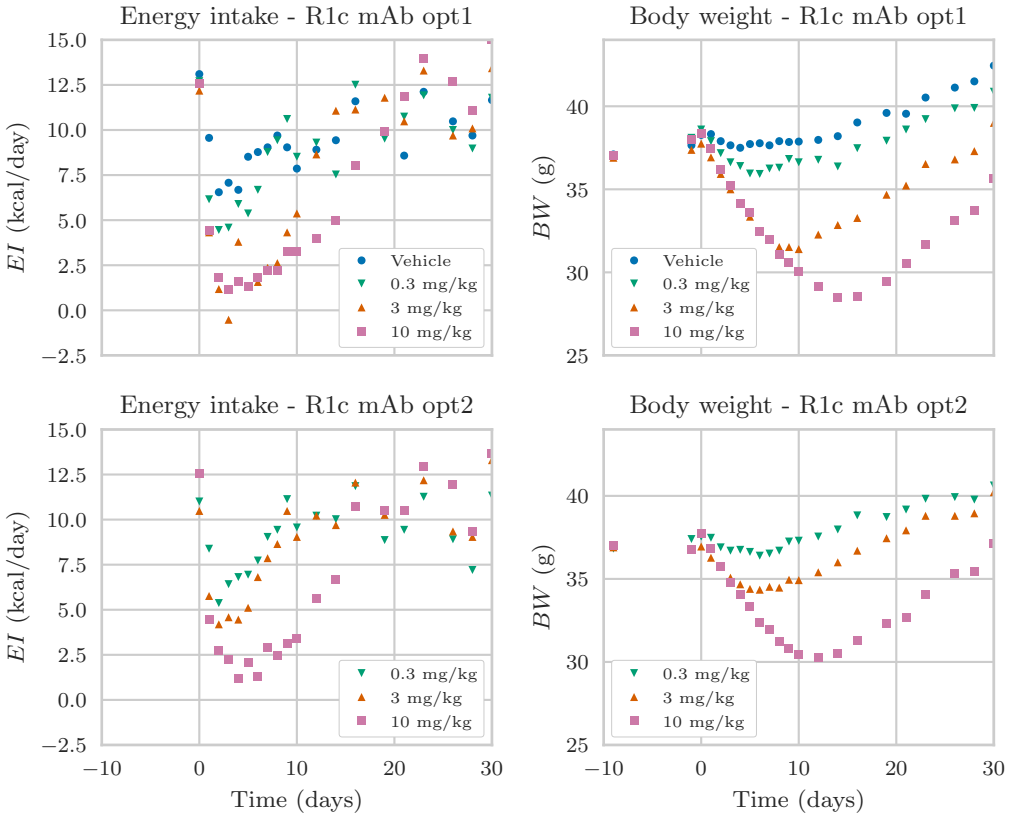

Figure 5.1: Group means of measured energy intake (left column) and body weight (right column) for the mouse dataset. The treatment was started at $t = 0$.

The human body-weight datasets collected from the literature were selected using the following criteria: less than 30% diabetic patients, the patients not being on

a very-low calorie diet, the first measurement before week 10, the second measurement before week 26, the last measurement after week 51, and a total number of measurements of at least 4. These criteria reflect the original purpose for collecting the data: guiding study design and dose prediction of anti-obesity drug trials. In total, 14 datasets from 11 studies were selected from Van Gaal et al. (2008), Torgerson et al. (2004), Pi-Sunyer et al. (2006), Smith and Goulder (2001), Hauner et al. (2004), Aronne et al. (2010), Wilding et al. (2004), Van Gaal et al. (2005), Després et al. (2005), Smith et al. (2010) and Allison et al. (2012). The data are shown in Fig. 5.2 and Tables A.4–A.6. The baseline body weights and BMI are shown in Table 5.4. In these studies, the drug was administered at regular intervals for the whole duration of the trial, in contrast to the mouse studies, where a single dose was administered.

Dataset	Body weight (kg)	BMI (kg/m^2)
Rimonabant, 20 mg, Van Gaal et al. 2008	101.1	36.3
Orlistat, 120 mg bds, Torgerson et al. 2004	110.4	37.3
Rimonabant, 20 mg, Pi-Sunyer et al. 2006	103.0	37.2
Sibutramine, 15 mg, Smith et al. 2001	87.0	32.7
Sibutramine, 15 mg, Hauner et al. 2004	99.5	35.1
Taranabant, 4 mg, Aronne et al. 2010	99.2	35.2
Topiramate, 96 mg, Wilding et al. 2004	105.3	37.3
Topiramate, 192 mg, Wilding et al. 2004	103.3	37.0
Topiramate, 256 mg, Wilding et al. 2004	106.3	37.9
Rimonabant, 20 mg, Van Gaal et al. 2005	101.7	36.2
Rimonabant, 20 mg, Despres et al. 2005	93.3	33.9
Lorcaserin, 10 mg bid, Smith et al. 2010	100.4	36.2
Phenterm.+Topiram., 3.75+23 mg, Allison et al. 2012	118.5	42.6
Phenterm.+Topiram., 15+92 mg, Allison et al. 2012	115.2	41.9

Table 5.4: Baseline body weight and BMI for the human body-weight datasets.

5.3.2 Test data

Although energy-intake data exist for the mouse data, these are only available at a small number of time points. These measurements could also potentially be noisy, as their accuracy is not known. Additionally, the physical activity parameters were estimated from the measured energy intake. Since the main purpose of performing estimation on synthetic data was to assess the ability of the methods to recover the true input function, it was not deemed suitable to use the same data that were used to fit the model, as this was likely to produce overly optimistic results. Therefore,

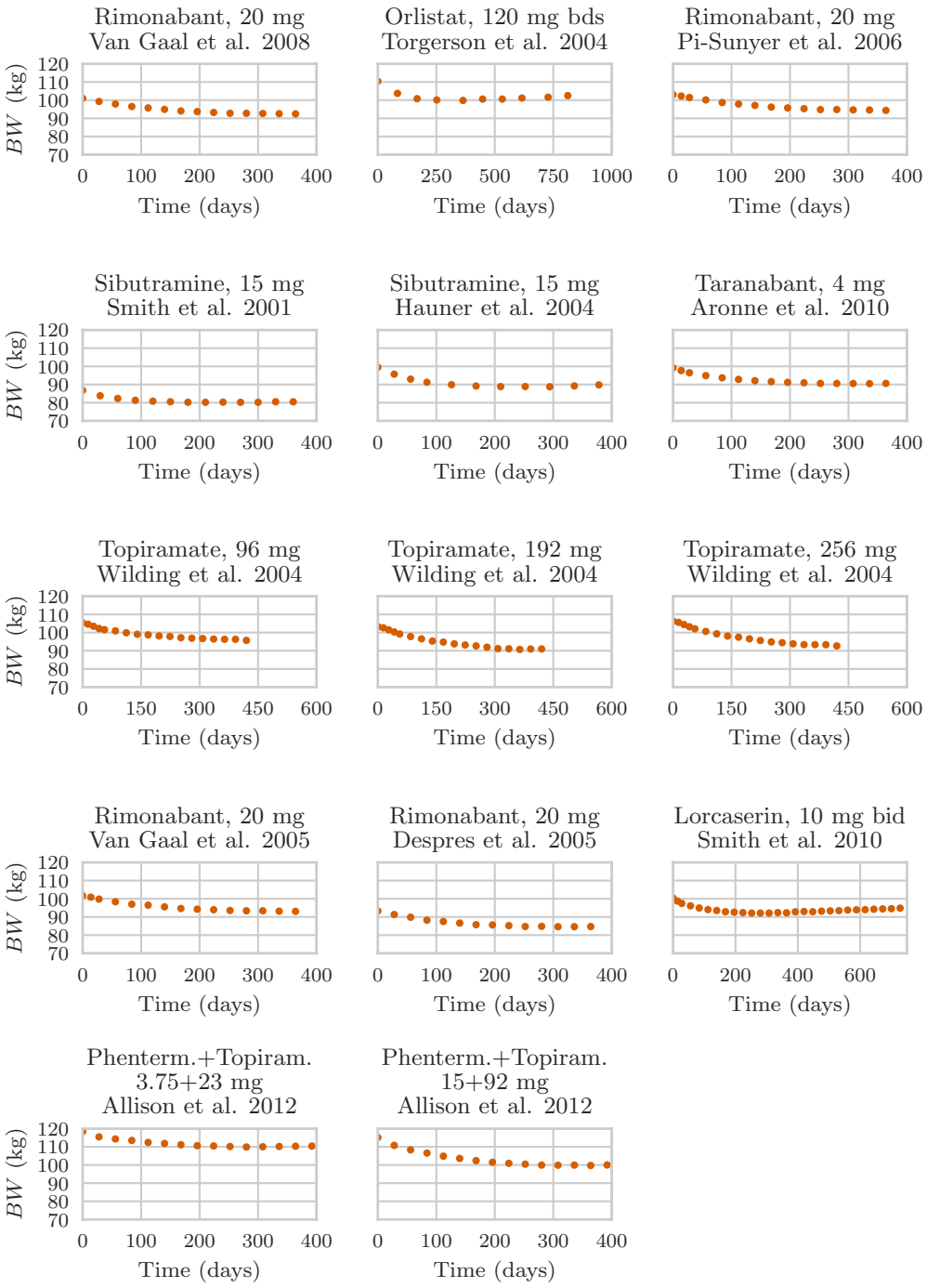

Figure 5.2: Group means of measured body weight for the human body-weight datasets.

synthetic input functions, which had no relationship to the assumed drug effect on the energy expenditure, were constructed for the mouse as well as for the human models.

The choice of synthetic input function was based on the observation that the human body-weight data are relatively smooth, and it is possible to fit the data using an input function of the form

$$EI = \begin{cases} EI_{init} & \text{if } t < 0 \\ EI_{init} - \Delta EI_{init} + (\Delta EI_{init} - \Delta EI_{final}) \frac{t^{\gamma EI}}{t^{\gamma EI} + t_h^{\gamma EI}} & \text{if } t \geq 0. \end{cases} \tag{5.15}$$

In this input model, the drug is assumed to have an effect on EI that is initially large, and gradually decreases to a long-term non-zero steady-state value. To simplify the notation, we assume that the start of treatment is at $t = 0$. This is the case for all datasets analysed here. If this were not the case, occurrences of t would have to be replaced by $t - t_{treat}$. The meaning of these parameters is illustrated in Fig. 5.3. The actual time for the drug to take effect in real experiments cannot be estimated from the available datasets, which typically contain one sample per week, starting at the time of intervention. Since the time series span over several weeks or years, modelling the drop in energy intake as being instantaneous was considered to be sufficiently accurate.

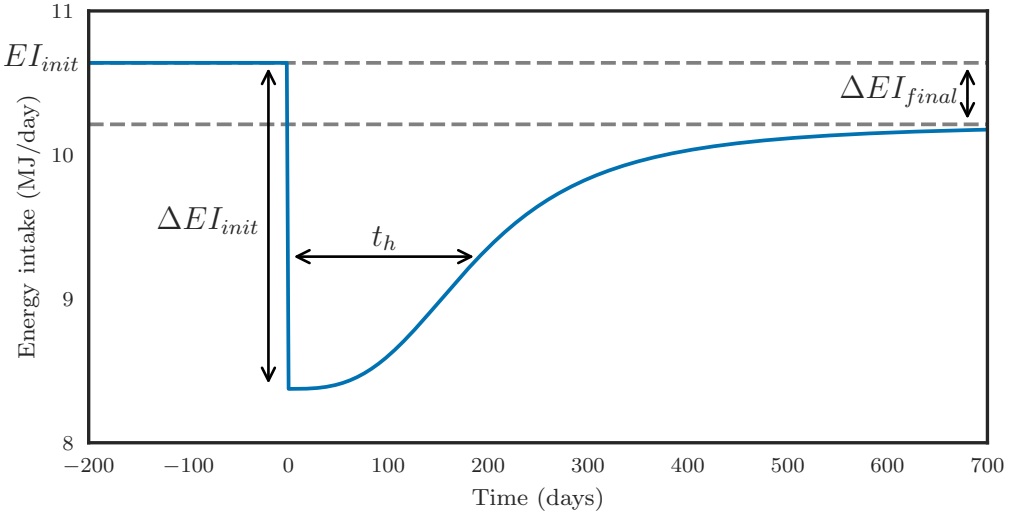

Figure 5.3: The synthetic input function for the human body-weight dataset. The meanings of the function parameters are annotated in the figure.

For the mouse datasets, an additional exponential function was introduced to make the input function smoother and avoid the instantaneous step at $t = 0$. The form of the modified function is

$$EI = \begin{cases} EI_{init} & \text{if } t < 0 \\ EI_{init} - \Delta EI_{init} + (\Delta EI_{init} - \Delta EI_{final}) \frac{t^{\gamma_{EI}}}{t^{\gamma_{EI}} + t_h^{\gamma_{EI}}} + A_{EI} e^{-\frac{t}{\tau_{EI}}} & \text{if } t \geq 0 \end{cases} \tag{5.16}$$

where A_{EI} and τ_{EI} are the amplitude and time constant for the additional exponential function. The parameter values that were used are shown in Table 5.5. Synthetic data were generated by using this function as input to the mouse body-weight model with the physical activity parameters for the R1c mAbs opt1 10 mg/kg dose group. Simulations were performed from $t = -9$ up to $t = 30$ days.

Parameter	Interpretation	Value	Unit
EI_{init}	Baseline energy intake	11.2	kcal $\cdot$ day^{-1}
ΔEI_{init}	Initial energy intake change	10.3	kcal $\cdot$ day^{-1}
ΔEI_{final}	Final energy intake change	-5.5	kcal $\cdot$ day^{-1}
t_h	Time to reach halfway to steady-state	18	day
γ_{EI}	Hill coefficient	3	-
A_{EI}	Exponential component amplitude	11.6	kcal $\cdot$ day^{-1}
τ_{EI}	Exponential component time constant	0.8	day^{-1}

Table 5.5: Parameters for the synthetic input function for the mouse body-weight model.

For the human body-weight model, Eq. (5.15) was used without any additional terms. The exponential term was not considered necessary, as the real data can be captured without such a term, and simulation does not start until the start of the treatment at $t = 0$. The parameters that were used are shown in Table 5.6. Synthetic data were produced by using this function as the input to the human body-weight model, using the baseline values from the Torgerson et al. (2004) time series, which set the initial body weight to 110.4 kg, and the initial BMI to 37.3 kg/m^2, making for a baseline energy intake of 11.4 MJ/day. The system was simulated from $t = 0$ up to $t = 812$ days.

It can be noted that for the mouse, ΔEI_{final} was negative, meaning that the final EI was higher than the EI prior to treatment. In contrast, the human model had a lower EI at the end of the study. This reflects the fact that while the mice were administered a single dose, whose effect may have disappeared by the end of the study, the humans were dosed repeatedly throughout the experiment.

Parameter	Interpretation	Value	Unit
EI_{init}	Baseline energy intake	11.4	$\text{MJ} \cdot \text{day}^{-1}$
ΔEI_{init}	Initial energy intake change	3.5	$\text{MJ} \cdot \text{day}^{-1}$
ΔEI_{final}	Final energy intake change	0.3	$\text{MJ} \cdot \text{day}^{-1}$
t_h	Time to reach halfway to steady-state	156	day
γ_{EI}	Hill coefficient	3	-

Table 5.6: Parameters for the synthetic input function for the human body-weight model.

Four datasets were created for each of the mouse and human models:

Densely sampled, no noise. Samples were obtained at 100 equispaced time points between the start and end time points.

Densely sampled, with noise. This dataset used the same sampling schedule as above, but zero-mean Gaussian noise was added to each measurement, with a proportional standard deviation of 0.5% of the true value.

Sparsely sampled, no noise. This dataset uses the same sampling schedule as the real data. For mice, the sampling schedule of the opt1 10 mg/kg dose was used. For humans, the sampling schedule of the Torgerson et al. (2004) study was used.

Sparsely sampled, with noise. This dataset used the same sampling schedules as above, with 0.5% proportional Gaussian noise. This noise level was chosen to agree with the assumed noise level for the real data.

5.4 Materials and methods

5.4.1 Optimal-control methods

The priors and optimisation methods were chosen to be the same as in Chapter 4, in order to facilitate comparisons. Hence, for every time series, optimal control-based input estimation was performed using the priors for penalisation of the first derivative, penalisation of the second derivative, and the maximum entropy-based prior. For derivative-based priors, estimation was also performed by penalising the logarithm of the input function. Nonnegativity inequality constraints were introduced where necessary to avoid unphysical solutions. Three direct optimal-control methods were evaluated: single shooting, multiple shooting, and collocation, as described in Section 3.2. The regularisation parameter was estimated using the discrepancy

Model	Method	Decision variables	Equality constraints	Inequality constraints
Mouse	Single shooting	100	0	1/101
	Multiple shooting	300	200	1/101
	Collocation	900	800	1/101
Human	Single shooting	100	0	1/101
	Multiple shooting	600	500	1/101
	Collocation	2100	2000	1/101

Table 5.7: Number of decision variables and constraints for the optimal-control problem formulations of the body-weight models. The number of inequality constraints is 101 if nonnegativity constraints are present, and 1 otherwise.

criterion. In all cases, the input function was represented by a piecewise constant function, discretised into $N_B = 100$ equispaced intervals. For the collocation methods, the state in each interval was represented by a Lagrange polynomial of degree $d = 3$, using Radau collocation points. In all cases, the state discretisation intervals were chosen to coincide with the input discretisation intervals. As the piecewise constant functions are not differentiable at the discretisation points, the derivative-based priors were based on a finite difference approximation, using Eqs. (4.3) and (4.4). The rationale for these choices was identical to that of Section 4.4.1. As the estimation methods do not allow zero measurement noise, estimation for the noise-free test data was performed assuming that the noise standard deviation was 0.005%, a factor 100 times lower than the noise standard deviation of the noisy data.

Table 5.7 shows the total number of decision variables in the resulting optimisation problems. There are N_B decision variables representing the input that are present in all problem formulations. Multiple shooting methods add an additional $N_B \cdot d_x$ variables for the states at the beginning of each shooting interval, while collocation methods add $N_B \cdot (d+1)d_x$ variables for the states at the beginning of each collocation interval and at the collocation points.

In total, 15 combinations of priors and optimisation methods were tested on each time series. The number of time series amounts to 4 synthetic and 7 real time series for the mouse model, and 4 synthetic and 14 real time series for the human model. This makes the number of optimal-control estimation procedures $15 \times (4 + 7 + 4 + 14) = 435$.

5.4.2 MCMC methods

To evaluate the MCMC methods, the same settings were used as in Chapter 4. Four priors were used: penalisation of the first and the second derivative, applied to the function and to the logarithm of the function. The regularisation parameter was in all cases treated as a Bayesian parameter, and assigned a Gamma prior with hyperparameters $\alpha = \beta = 10^{-3}$. Cubic B-splines and Karhunen-Loève functions were used as basis functions. The knots of the B-splines were placed at the measurement time points. Additional knots were placed at the first and last time points to ensure that all intervals were covered by four basis functions. For densely sampled test data, the knots were placed at the points of the measurements of the corresponding sparsely sampled test data. This was done in order to avoid an overly high-dimensional parameterisation. 20 basis functions were used for the Karhunen-Loève parameterisation. For noise-free data, 0.005% measurement noise was assumed, similar to the optimal-control methods. The basis function coefficients were updated either one at a time using RWMH, or jointly using either RWMH, MALA or SMMALA. In all cases, the regularisation parameter was updated using Gibbs sampling, where updates were computed as described in Section 4.4.2.

The initial parameter values were obtained from the results of the optimal control-based methods. The Hessian reported at the optimum was used to initialise the proposal densities and the metric tensor. During the initial part of the MCMC runs, the proposal densities were tuned to achieve a target acceptance ratio. After this was achieved, the estimation procedure was executed for 5 minutes of processing time. This procedure was identical to the procedure used for the eflornithine dataset, and is described in more detail in Section 4.4.2.

Each time series was analysed using all combinations of the two function parameterisations, four priors, and four sampling methods. As there were $4 + 7 + 4 + 14 = 29$ synthetic and real time series combined, in total $2 \times 4 \times 4 \times 29 = 928$ MCMC runs were performed.

5.5 Results and discussion

5.5.1 Optimal-control methods

The accuracy of the optimal-control methods was assessed by computing the RMSE, as defined in Eq. (4.13), between the estimated and the true input functions for the test data. Although energy-intake data were available for the real mouse data, they were not used to evaluate accuracy, as the same data were used for fitting some of

the model parameters. The results are shown in Table 5.8 for mouse data and in Table 5.9 for human data. The RMSE depends only on the prior, and not on the choice of optimisation method, as long as the optimisation procedure was successful. For Tables 5.8 and 5.9, collocation was used, as this method was successful for all test data time series.

Dataset	Prior	RMSE (kcal/day)	Dataset	Prior	RMSE (kcal/day)
Dense,	1der	0.25	Dense,	1der	0.75
0% noise	1der, log	0.23	0.5% noise	1der, log	0.76
	2der	0.25		2der	0.79
	2der, log	0.24		2der, log	0.74
	entropy	0.24		entropy	0.75
Sparse,	1der	0.30	Sparse,	1der	1.11
0% noise	1der, log	0.27	0.5% noise	1der, log	1.49
	2der	1.09		2der	1.70
	2der, log	0.63		2der, log	1.81
	entropy	0.66		entropy	1.49

Table 5.8: RMSE values for the synthetic mouse dataset.

Dataset	Prior	RMSE (MJ/day)	Dataset	Prior	RMSE (MJ/day)
Dense,	1der	0.03	Dense,	1der	0.10
0% noise	1der, log	0.03	0.5% noise	1der, log	0.11
	2der	0.03		2der	0.12
	2der, log	0.03		2der, log	0.15
	entropy	0.03		entropy	0.11
Sparse,	1der	0.03	Sparse,	1der	0.25
0% noise	1der, log	0.03	0.5% noise	1der, log	0.26
	2der	0.05		2der	0.18
	2der, log	0.05		2der, log	0.16
	entropy	0.03		entropy	0.25

Table 5.9: RMSE values for the synthetic human dataset.

For the densely sampled data, the effect of the prior on the RMSE value is very small. This holds for the mouse as well as the human model. It can also be seen that for the human data, accuracy remains good even in the presence of sparse sampling. On the other hand, the mouse data show a clear dependence on the sampling schedule. Even for sparse and noisy data, the prior has a relatively small

effect on the RMSE value.

Estimation results for sparse noisy test data for all priors are shown in Figs. 5.4 and 5.5. These time series were chosen as they are the test time series that are most similar to the real data. The figures show clear differences between the datasets. While the mouse time series were very densely sampled from $t = -1$ to $t = 30$, there is an initial period of 8 days where no measurements are available. Furthermore, the input function stays constant during this interval, after which a rapid decrease is initiated. This makes for a very challenging estimation problem, as there is a large range of energy-intake profiles that could explain the measurements. In particular, an energy-intake profile that starts at a high value at $t = -9$ and gradually decreases until $t = -1$ is able to explain the data very well, due to the lack of measurements in this interval. All of the tested priors reward smoothness, and thus favour such solutions over the true input. In particular, the priors based on penalisation of the second derivative make strong smoothness assumptions, causing them to completely miss the discontinuity. As Fig. 5.4 shows, an additional challenge is that energy-intake profiles that are substantially different can still result in similar body-weight profiles. This can be attributed to the low-pass characteristics of the system: rapid changes in energy intake do not cause equally rapid changes in body weight.

It can be argued that the statistical soundness of all the evaluated priors is limited in situations when the characteristics of the input function can be expected to change rapidly at the time of intervention. As in Chapter 4, penalisation of the first derivative appears to be a more conservative choice than penalisation of the second derivative. One possibility to make the estimate more realistic is to assume that the body weight and energy intake remain constant prior to intervention. However, this would also introduce additional assumptions which would need to be justified.

In contrast, the human body-weight data in Fig. 5.5 have no long intervals without measurements, even though the number of samples is less than for the mouse data. They also lack the rapid change in energy intake that is exhibited by the mouse data. This makes the estimates less sensitive to the prior. However, the methods still tend to underestimate the initial drop in energy intake. Similar to the mouse data example, the body-weight predictions are very similar for all priors, despite minor differences in the energy-intake profiles. The relative insensitivity to the choice of prior makes the estimation methods easier to justify in terms of statistical soundness, as the input profile appears to be determined mainly by the data.

Next, the performance and robustness of the methods were assessed. Tables 5.10 and 5.12 show running times and the number of successful optimisations organised by optimisation method, while Tables 5.11 and 5.13 show the same information organised

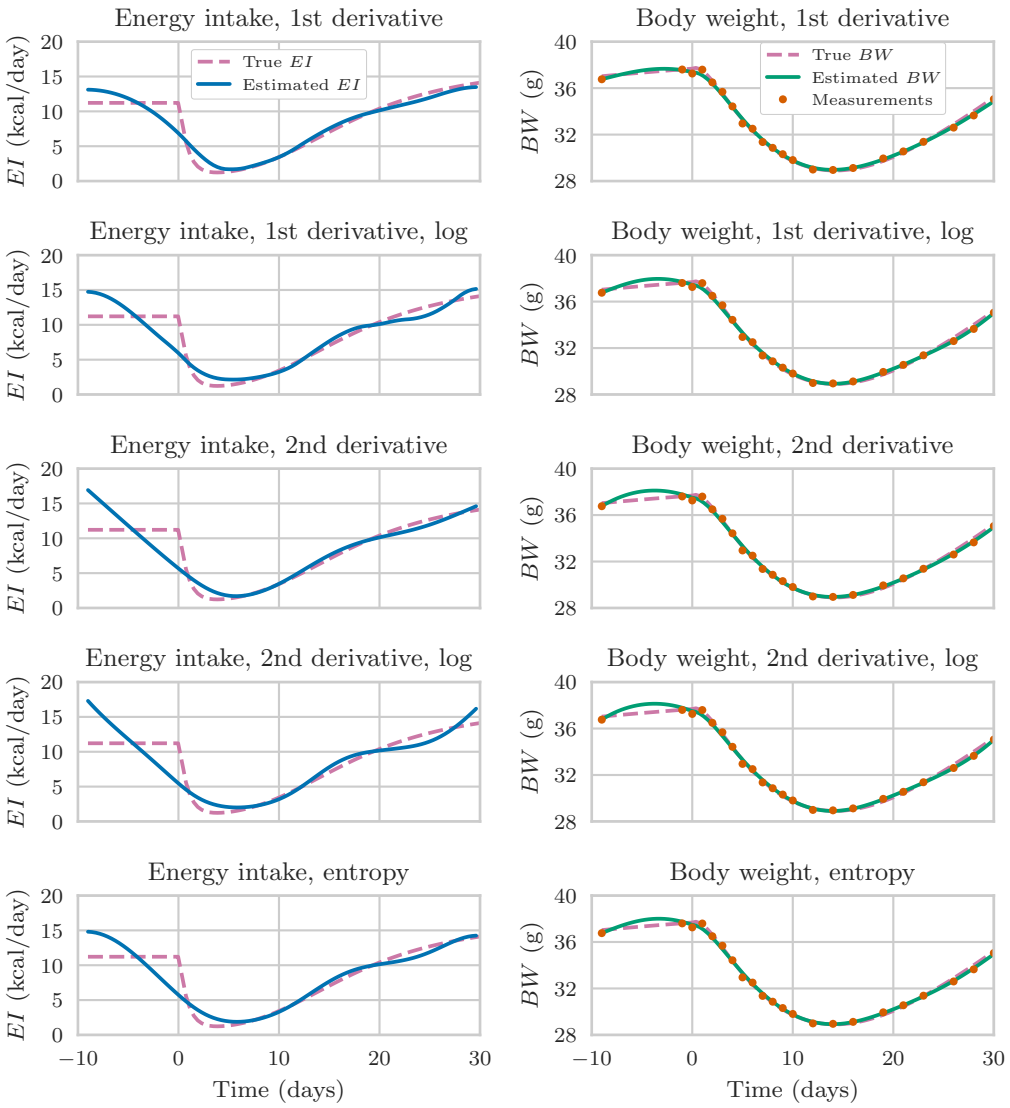

Figure 5.4: Optimal-control estimation results for the mouse body-weight model. Sparse test data with 0.5% noise.

by prior. Here, optimisation was considered to be successful if the return status from the optimisation software was either `Solve Succeeded` or `Solved To Acceptable Level`. In terms of robustness, there does not seem to be any major difference between the optimisation methods or priors. It can be seen that the collocation methods are much faster than the shooting methods. A major difference between the mouse and human models is that the running time for the shooting methods is one order of magnitude larger for the human model, although the number of iterations is similar.

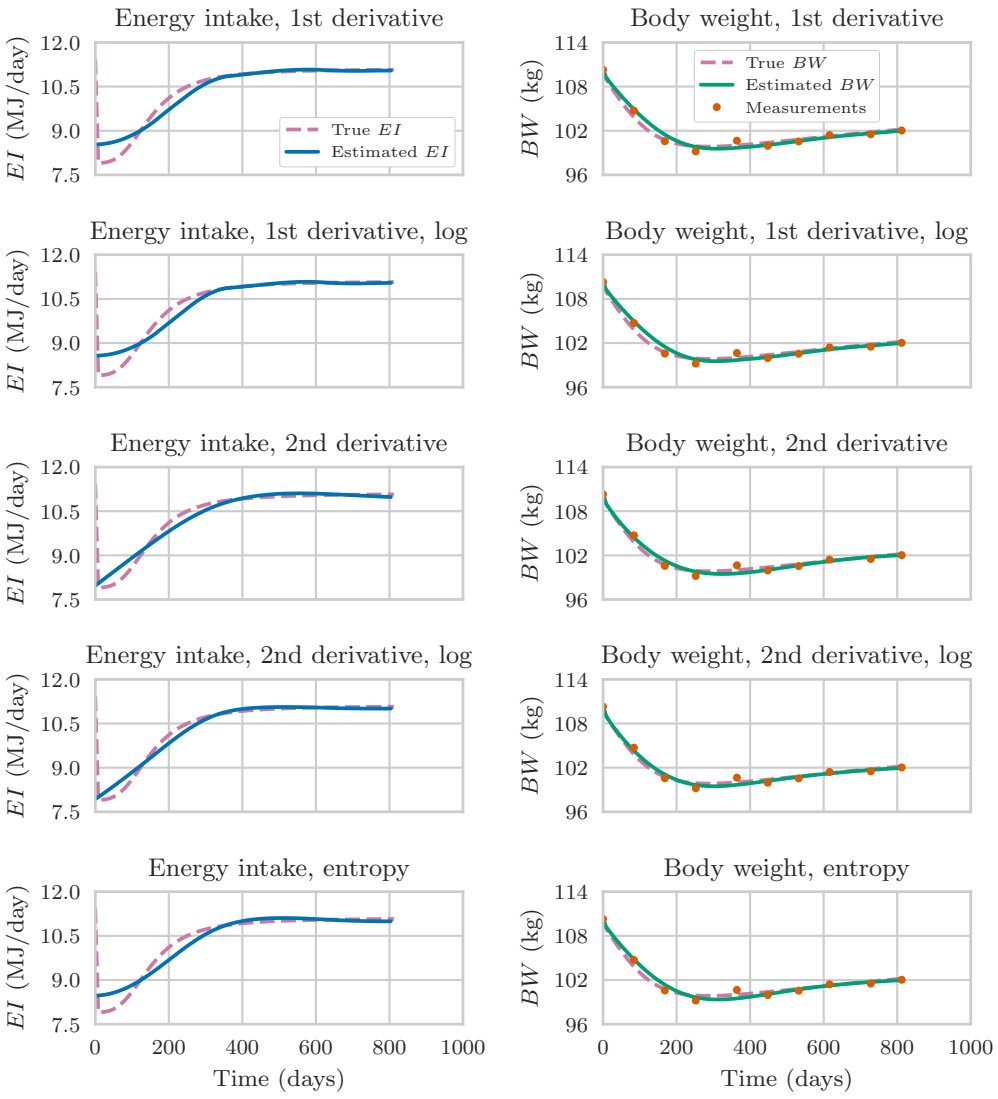

Figure 5.5: Optimal-control estimation results for the human body-weight model. Sparse test data with 0.5% noise.

This may be partly explained by the larger number of states in the human model. It is also possible that the human model has issues of stiffness. In particular, the dynamics of the ECF state are very fast compared to the overall time scale of the problem.

Figures 5.6 and 5.7 show examples of the estimation results for the real data for all mouse and human time series. In all examples shown, penalisation of the first derivative was used as a prior. For the mouse data, the measured energy intake is

	Median time (s)	Median iterations	Successful runs
Single shooting	8.1	24	50/55
Multiple shooting	3.2	24	50/55
Collocation	0.2	20	55/55

Table 5.10: Median running times, number of iterations, and the proportion of successful estimation procedures on the mouse body-weight dataset, organised by optimisation method.

	Median time (s)	Median iterations	Successful runs
1der	4.2	19	31/33
1der, log	3.5	26	30/33
2der	3.2	18	32/33
2der, log	3.3	26	32/33
entropy	2.5	19	30/33

Table 5.11: Median running times, number of iterations, and the proportion of successful estimation procedures on the mouse body-weight dataset, organised by choice of prior.

	Median time (s)	Median iterations	Successful runs
Single shooting	69.9	24	90/90
Multiple shooting	29.4	28	85/90
Collocation	0.5	27	88/90

Table 5.12: Median running times, number of iterations, and the proportion of successful estimation procedures on the human body-weight dataset, organised by optimisation method.

	Median time (s)	Median iterations	Successful runs
1der	23.9	21	54/54
1der, log	39.7	32	51/54
2der	19.5	19	53/54
2der, log	48.4	35	51/54
entropy	24.2	23	54/54

Table 5.13: Median running times, number of iterations, and the proportion of successful estimation procedures on the human body-weight dataset, organised by choice of prior.

also shown for comparison. Is is evident from the figure that the method is not able to capture the energy intake measurement at $t = 0$. This is similar to the results for simulated data, as discussed on Page 134.

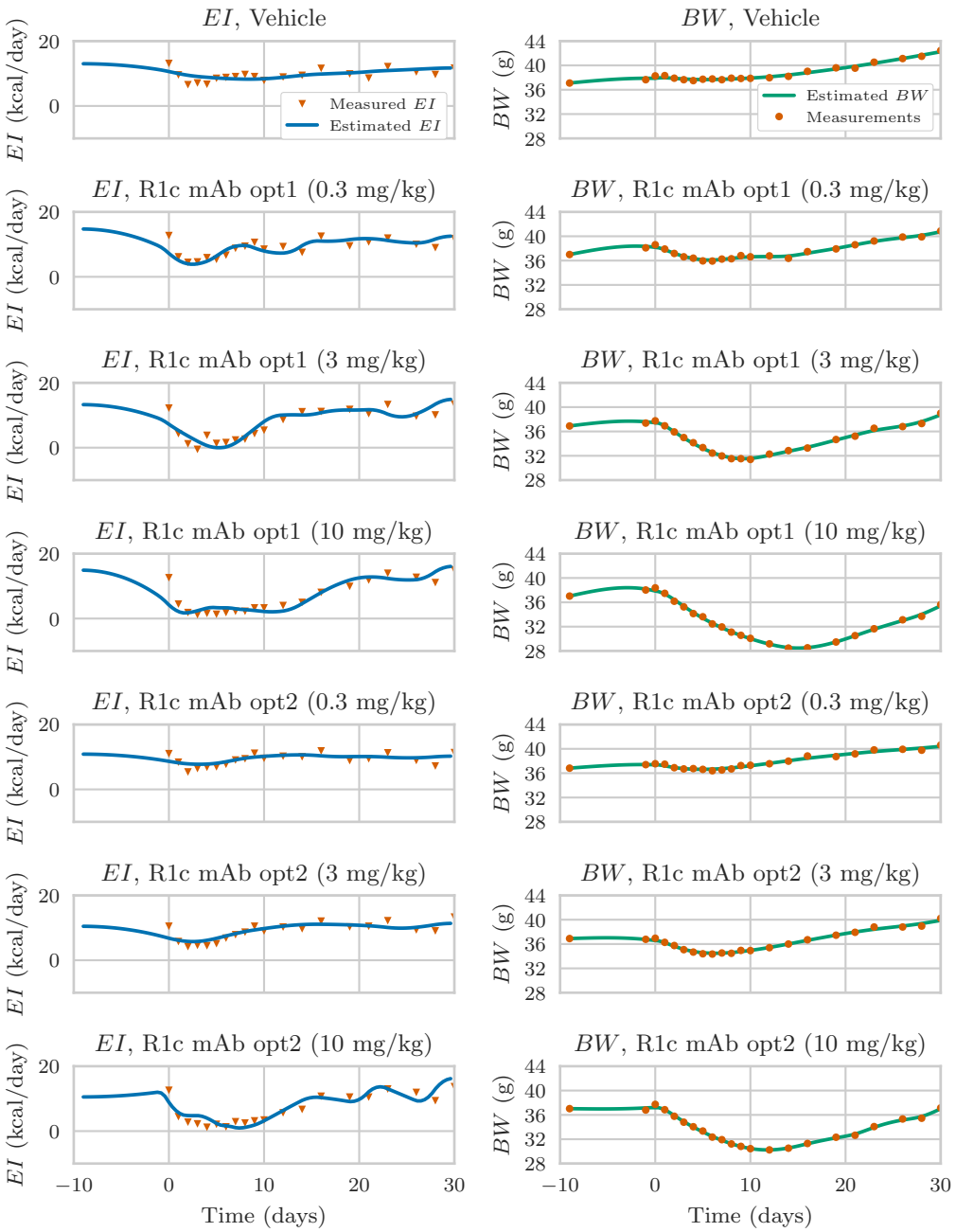

Figure 5.6: Examples of the estimated energy-intake profiles for the real mouse data, obtained using optimal-control methods with penalisation of the first derivative as the prior.

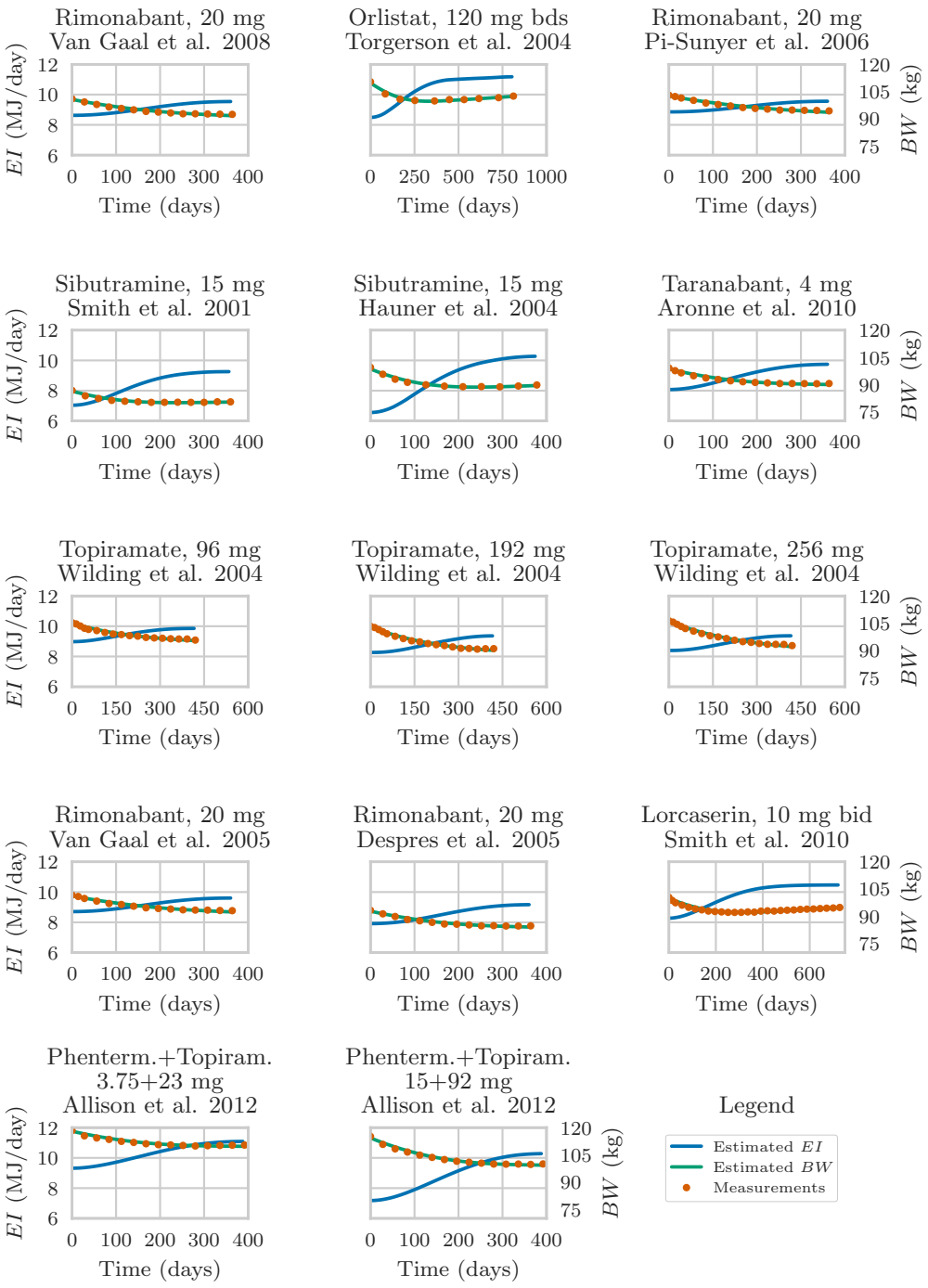

Figure 5.7: Examples of the estimated energy-intake profiles for the real human data, obtained using optimal-control methods with penalisation of the first derivative as the prior.

5.5.2 MCMC methods

As a first step in evaluating the MCMC methods, the estimation accuracy was assessed for the test datasets, using the mean RMSE defined in Eq. (4.13), averaged over all MCMC samples. For the human dataset, the samples at time $t = 0$ were excluded when computing the RMSE. This was done as the true input starts at the baseline energy intake EI_{init}, and instantaneously drops by ΔEI_{init} (see Fig. 5.3). Since this initial high energy intake is not visible in the simulated data, no estimation algorithm would be able to detect it, and its inclusion would cause the RMSE value to be dominated by the contribution from that single time point.

All generated sets of samples were diagnosed using the Raftery-Lewis diagnostics as well as the effective sample size (Section 3.3.3). An estimation procedure was considered to be successful if

1. The estimation procedure exited normally. A number of procedures exited due to errors such as obtaining a non-positive definite metric tensor.

2. The required number of samples, as reported by the Raftery-Lewis criterion, was at least as large as the actual number of generated samples.

The RMSE values are shown in Table 5.14 for the mouse, and in Table 5.15 for the human dataset. Unsuccessful estimation procedures were excluded from these tables, as the generated MCMC samples from these are not an accurate representation of the true posterior. The estimation procedures were unsuccessful in all cases where data were noise-free. This is likely attributable to the fact that a highly peaked likelihood function causes the posterior to be concentrated in a small area which is difficult for the MCMC methods to explore. In the sequel, only noisy data are considered, and all reported values refer only to such data.

Recall that the RMSE measure should depend only on the prior and parameterisation, and not on the sampling method. While small discrepancies are to be expected due to the statistical nature of Monte Carlo methods, larger differences between the sampling methods are a sign that at least one of the methods failed to obtain representative samples of the posterior distribution. In Table 5.14, it can be seen that for the mouse model, the RMSE values are similar across all sampling methods for any given prior and parameterisation. However, Table 5.15 shows that for the human model, larger than expected differences between the sampling methods were obtained for first derivative penalisation when using B-splines. A visual inspection confirms that the block RWMH and MALA methods failed to explore the posterior well (Fig. 5.8), despite the ESS and Raftery-Lewis diagnostics showing

satisfactory values. This is a reminder that, although such diagnostics can help to find problems in the sampling method, successful tests are not a guarantee that the method performs well, as discussed in Section 3.3.3.

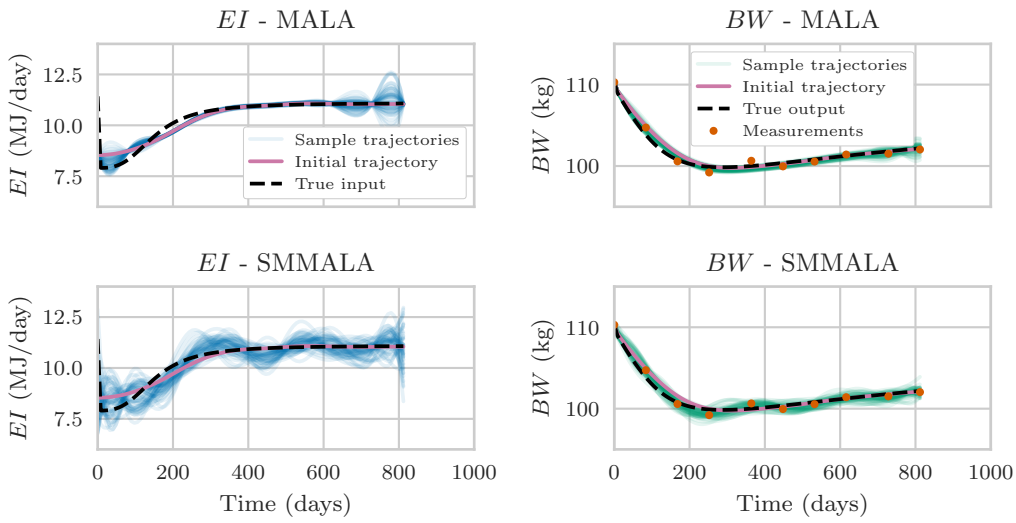

Figure 5.8: Sample trajectories for the sparse noisy human test data using B-splines and penalisation of the first derivative. Top: MALA sampling. Except for the initial and final portion of the curve, all samples are closely clustered around the initial trajectory. Bottom: SMMALA sampling. Here, the sampling algorithm generates a wider range of trajectories. This suggests that MALA fails to explore the whole posterior.

In general, the choice of prior and parameterisation does not appear to have a major effect on the accuracy. This may be attributable to the relatively dense sampling compared to typical PK datasets. However, a general trend is that, for any given dataset and parameterisation, the log-domain RMSE values are slightly higher than their linear-domain counterparts. Figures 5.9 and 5.10 show examples of the estimation results for the sparse noisy test data using various priors. For the mouse, it can be clearly seen how a log-domain prior can increase the uncertainty region. In order to fit the data in the region where the energy intake drops close to zero, the estimated regularisation parameter has to be set to a relatively small value, causing large uncertainties for time points where the estimated energy intake is high. It can be argued that in body-weight applications, a log-domain prior is a less natural choice than in PK applications. Daily energy intake values are unlikely to vary by orders of magnitude over the course of the experiment, in contrast to plasma concentrations of drugs.

One disadvantage of the linear-domain priors is that nonnegativity constraints cannot be enforced. However, in most of the investigated cases, such constraints were not necessary. For the mouse dataset, the estimated input did occasionally drop slightly below zero. However, this is not likely to significantly influence the prediction, in contrast to the example in Chapter 4, where the lack of nonnegativity constraints caused dramatic increases in the uncertainty regions. If such constraints are desired, a pragmatic solution would be to assign a zero probability to any input function which drops below zero. Hence, any such proposed function would be rejected by the Metropolis-Hastings accept-reject step. This may be sufficiently efficient, provided that only a small part of the unconstrained posterior contains function values below zero. Otherwise, the rejection rate may become excessive.

Penalisation of the second derivative results in smoother estimates and smaller uncertainty regions than penalisation of the first derivative. However, this can also result in overconfident predictions when the actual function is not smooth. This can be seen in Fig. 5.9, where the sharp decrease in energy intake at the start of the treatment is challenging for all tested priors. When penalisation of the first derivative is employed, the uncertainty region is relatively large, and the 95% credible interval covers the true input. In contrast, penalisation of the second derivative results in a narrow credible interval that completely misses the initial part of the input function.

To summarise the findings with respect to statistical soundness, the considerations for the MCMC algorithms are similar to the considerations for the optimal-control algorithms. Any prior which does not explicitly account for the change in characteristics at the start of intervention is likely to miss important features if data are sparsely sampled. If it cannot be assumed that the energy intake and body weight are constant before intervention, it may be necessary to collect more data.

In terms of speed and robustness of the MCMC methods, the main results are presented in Tables 5.16 and 5.17. These tables show the results for sparse noisy time series, including simulated as well as real data. Speed and robustness tend to be related, as a common reason for failure is that a set of samples of sufficiently high quality could not be generated in the allocated time. In terms of the sampling methods, the results differ substantially for the two models. For the mouse, by far the most successful option was block RWMH with Karhunen-Loève basis functions. Although the required number of samples was higher for RWMH than for the gradient-based methods, block RWMH was able to generate the samples at a higher rate, since no gradient computations are required. In contrast, SMMALA with Karhunen-Loève basis functions is the superior choice for the human model. However, even in this case there was a relatively high number of failures. For both models, single-component

Dataset	Method	Basis	Prior	RMSE kcal/day
Dense,	Block RWMH	KL	1der	0.98
0.5% noise	Block RWMH	KL	1der, log	1.12
	Block RWMH	KL	2der	0.93
	Block RWMH	KL	2der, log	1.00
	Block RWMH	spline	1der	1.09
	Block RWMH	spline	1der, log	1.20
Sparse	Block RWMH	KL	1der	1.53
0.5% noise	Block RWMH	KL	1der, log	2.10
	Block RWMH	KL	2der	1.64
	Block RWMH	spline	1der	2.25
	Block RWMH	spline	1der, log	2.59
	MALA	KL	1der	1.55
	MALA	KL	2der	1.65
	MALA	spline	1der	2.23
	SMMALA	KL	1der	1.53

Table 5.14: Accuracy measures for the MCMC methods on the sparse noisy mouse body-weight test datasets.

RWMH was found to be completely unsuitable.

One possible reason why SMMALA is more successful than the other methods for the human data is that this method is more tolerant to a poor initial state for the Markov chain. As Fig. 5.8 shows, the initial trajectory obtained from the optimal-control analysis differs considerably from the true input function. All methods except SMMALA use the Hessian at the initial point to construct the proposal distributions for all steps of the chain. Therefore, if the initial point is poorly chosen, the sampling performance is likely to suffer at every iteration. In contrast, the proposal distribution in SMMALA depends only on the current state of the chain. For this reason, the chain will eventually "forget" a poor initial state. Figure 5.8 shows that the MALA estimates in most cases do not deviate much from the initial trajectory, which suggests that the chosen proposal is ill-suited for trajectories far away from the starting point.

In general, estimation using Karhunen-Loève basis functions achieved considerably higher performance in terms of speed and robustness than estimation using B-splines. As the estimation accuracy does not depend strongly on the choice of basis functions, the performance considerations alone may provide sufficient justification to prefer Karhunen-Loève functions. Additionally, a parameterisation that is known to result in good sampling performance increases the confidence in the estimation results.

Dataset	Method	Basis	Prior	RMSE MJ/day
Dense,	Block RWMH	KL	1der, log	0.33
0.5% noise	Block RWMH	KL	2der, log	0.51
	Block RWMH	spline	1der	0.13
	Block RWMH	spline	1der, log	0.26
	Block RWMH	spline	2der, log	0.28
	MALA	KL	1der, log	0.33
	MALA	KL	2der, log	0.51
	MALA	spline	1der	0.13
	MALA	spline	1der, log	0.25
	MALA	spline	2der, log	0.28
	SMMALA	KL	1der	0.24
	SMMALA	KL	1der, log	0.33
	SMMALA	KL	2der	0.15
	SMMALA	spline	1der	0.21
	SMMALA	spline	1der, log	0.25
	SMMALA	spline	2der	0.24
	SMMALA	spline	2der, log	0.29
Sparse	Block RWMH	spline	1der	0.25
0.5% noise	MALA	spline	1der	0.25
	MALA	spline	2der	0.34
	SMMALA	KL	1der	0.41
	SMMALA	KL	1der, log	0.72
	SMMALA	KL	2der	0.22
	SMMALA	spline	1der	0.48

Table 5.15: Accuracy measures for the MCMC methods on the sparse noisy human body-weight test datasets.

For the mouse model, the sparse test data resulted in moderately higher RMSE values than the dense test data, while sparsity of measurements appeared to have no effect for the human model. This is expected, as even the sparse data are considerably denser than is the case for the PK applications in Chapters 4 and 6, with the exception of the days prior to treatment in the mouse model. For the human model, it is also clear that sparsity of data causes many sampling methods to perform poorly. No comparisons between noise-free and noisy data can be performed for either model, as no sampling methods were able to produce results for the noise-free data.

Figures 5.11 and 5.12 show examples of estimation results for the real data, obtained using the settings that were deemed most appropriate. In both cases,

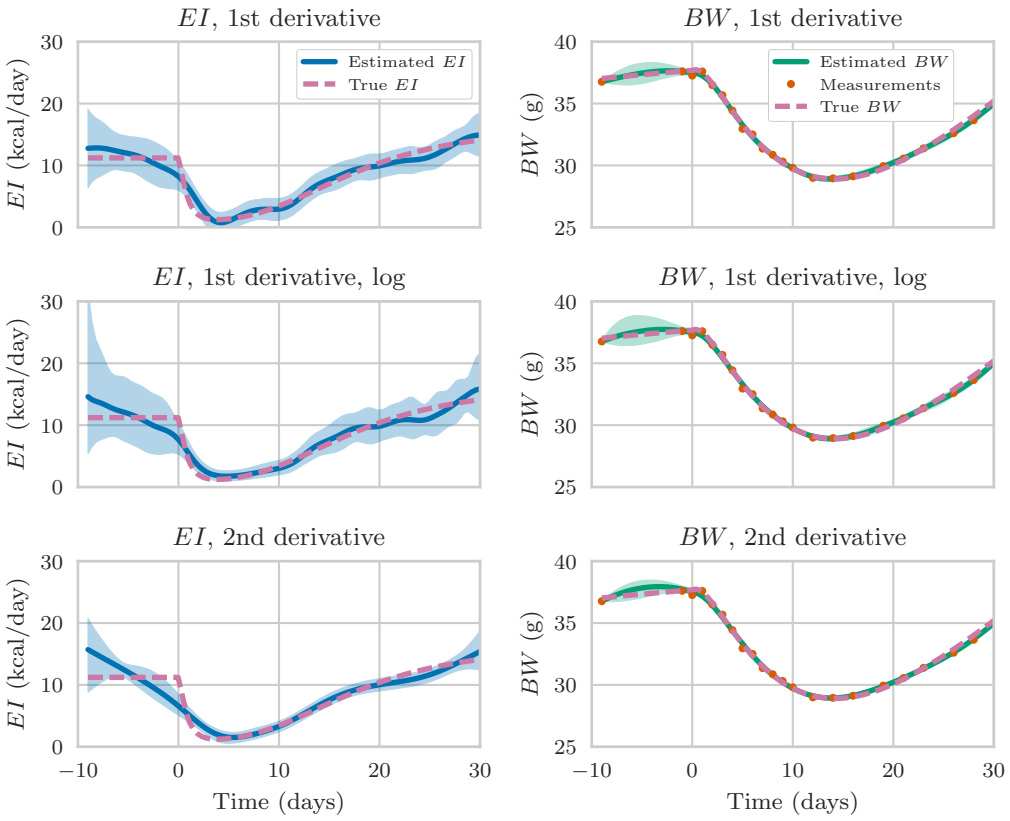

Figure 5.9: Examples of MCMC estimation results for the sparse noisy test data for the mouse model for various priors using Karhunen-Loève basis functions. Penalisation of the second derivative in the log domain is not shown, as no successful results are available. All shown results were obtained by block RWMH sampling.

		Median min ESS	Median max RL	Median required time	Successful runs
Single RWMH	KL	3	512546	6512	0/36
	spline	3	247472	5141	0/36
Block RWMH	KL	438	90805	87	30/36
	spline	15	658478	1424	12/36
MALA	KL	74	44809	412	18/36
	spline	4	316894	4387	5/36
SMMALA	KL	40	70557	669	15/36
	spline	14	209684	3428	5/36

Table 5.16: Performance and robustness results for MCMC methods on the noisy mouse body-weight datasets.

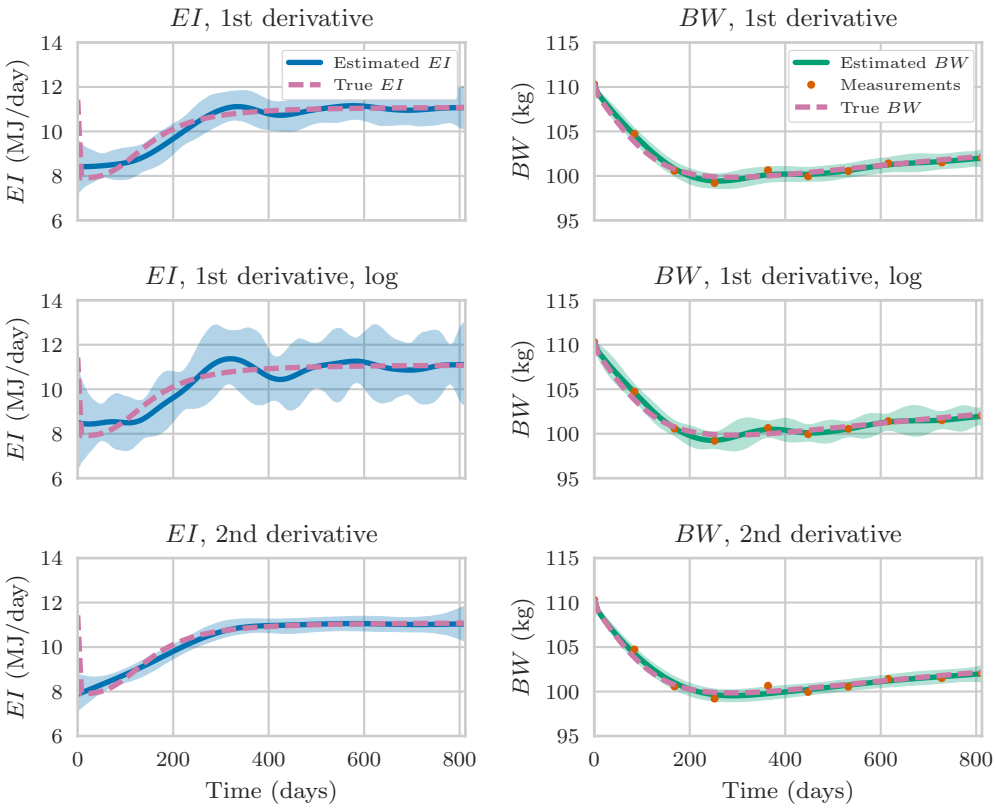

Figure 5.10: Examples of MCMC estimation results for the sparse noisy test data for the human model for various priors using Karhunen-Loève basis functions. Penalisation of the second derivative in the log domain is not shown, as no successful results are available. All shown results were obtained by SMMALA sampling.

		Median min ESS	Median max RL	Median required time	Successful runs
Single RWMH	KL	3	538748	8185	0/64
	spline	9	111755	2362	0/64
Block RWMH	KL	3	5585945	6108	4/64
	spline	18	597056	770	19/64
MALA	KL	7	183413	2479	11/64
	spline	17	162456	2359	22/64
SMMALA	KL	1103	4820	61	38/64
	spline	52	25725	571	19/64

Table 5.17: Performance and robustness results for MCMC methods on the noisy human body-weight datasets.

penalisation of the first derivative and Karhunen-Loève basis functions were used. The sampling method is block RWMH for the mouse and SMMALA for the human data.

5.6 Summary

The datasets that have been analysed in this chapter differ from the PK examples in Chapters 4 and 6 in several respects. First, the available data are much more densely sampled. Second, the assumed noise level is considerably lower. It can be noted that the sparse measurements were one of the main arguments for favouring batch methods over sequential methods such as particle filters, and this argument is less valid here. On the other hand, the low noise variance in the model may still cause problems for such methods. Evaluating sequential estimation methods could be interesting for future work.

In terms of RMSE, the choice of prior and function parameterisation does not appear to make a large difference. This may partly be explained by the fact that the measurements are relatively densely sampled. There is some indication that log-domain priors are less suitable here, which may be explained by the fact that the energy intake is not expected to vary by orders of magnitude. For MCMC estimation, this results in an inability to enforce nonnegativity constraints, but this does not appear to be an issue for the evaluated datasets.

Although the RMSE measures are similar across priors, the estimated functions are qualitatively different for different priors. Therefore, there may exist subjective reasons for preferring one prior over the others. If the energy intake is expected to be very smooth, a second-derivative prior may be appropriate. However, if this condition does not hold, such a prior can fail to track the input function. One possibility is to run estimation for several priors, in order to assess whether the results are sensitive to this choice.

All of the optimal-control methods performed well in terms of robustness. The collocation methods were considerably faster than the shooting-based methods, and should therefore be preferred. For the MCMC methods, Karhunen-Loève basis functions generally outperformed B-splines in terms of sampling performance. For the mouse model, block RWMH achieved the highest performance, while SMMALA performed best for the human model.

This chapter also shows examples of cases where the MCMC diagnostics, including ESS and the Raftery-Lewis diagnostics, failed to detect problems with the sampling method, highlighting the fact that seemingly satisfactory diagnostics

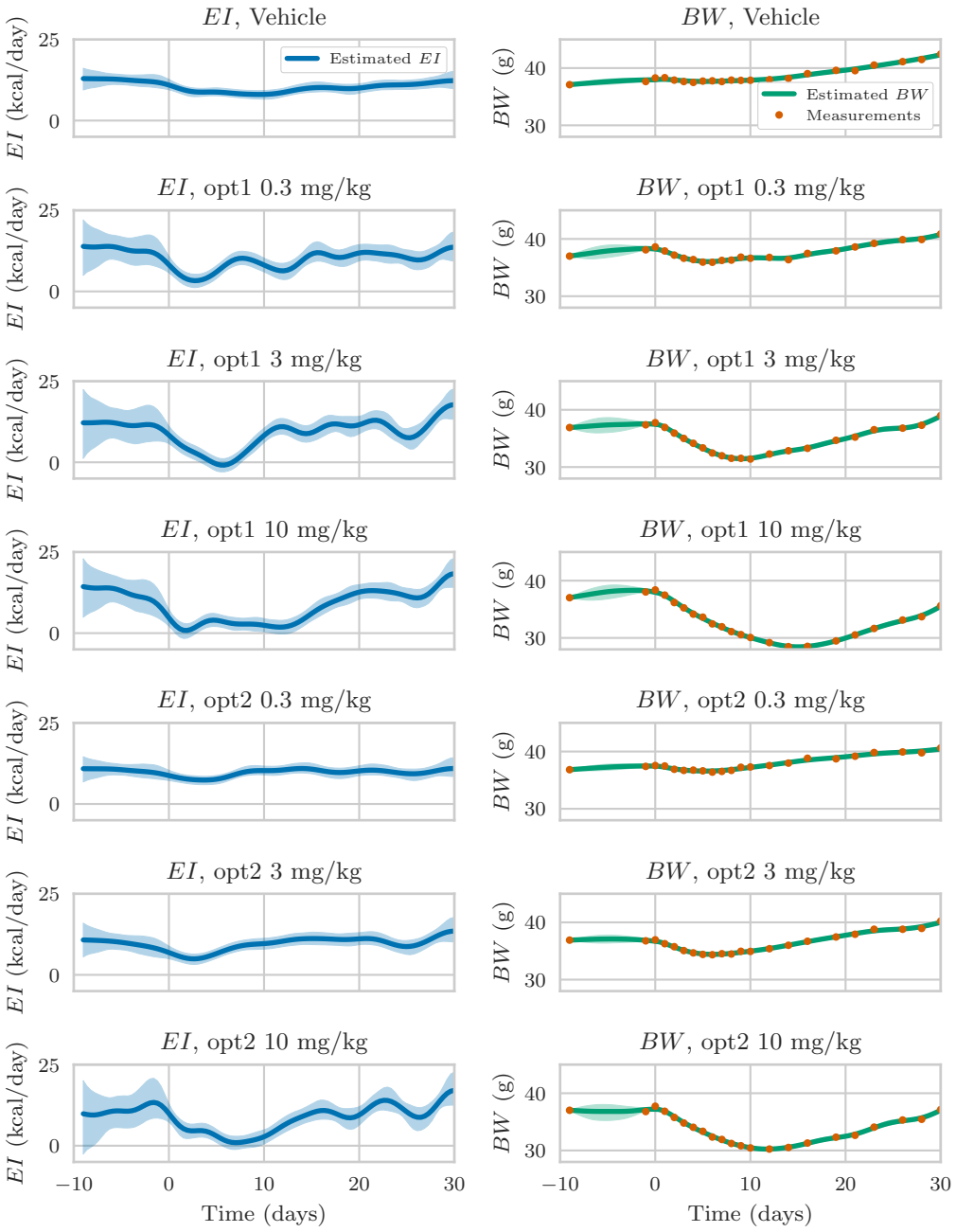

Figure 5.11: MCMC estimation results for the real mouse data using Karhunen-Loève basis functions and a prior based on the first derivative. In all cases, block RWMH was used as the sampling method.

results are no guarantee that sampling has been successful. Visual inspection of sample trajectories can help to detect obviously unsuccessful procedures. It can also

help if more than one sampling method is used, and the results compared. It would also be possible to use more than one kind of MCMC update in a single estimation procedure.

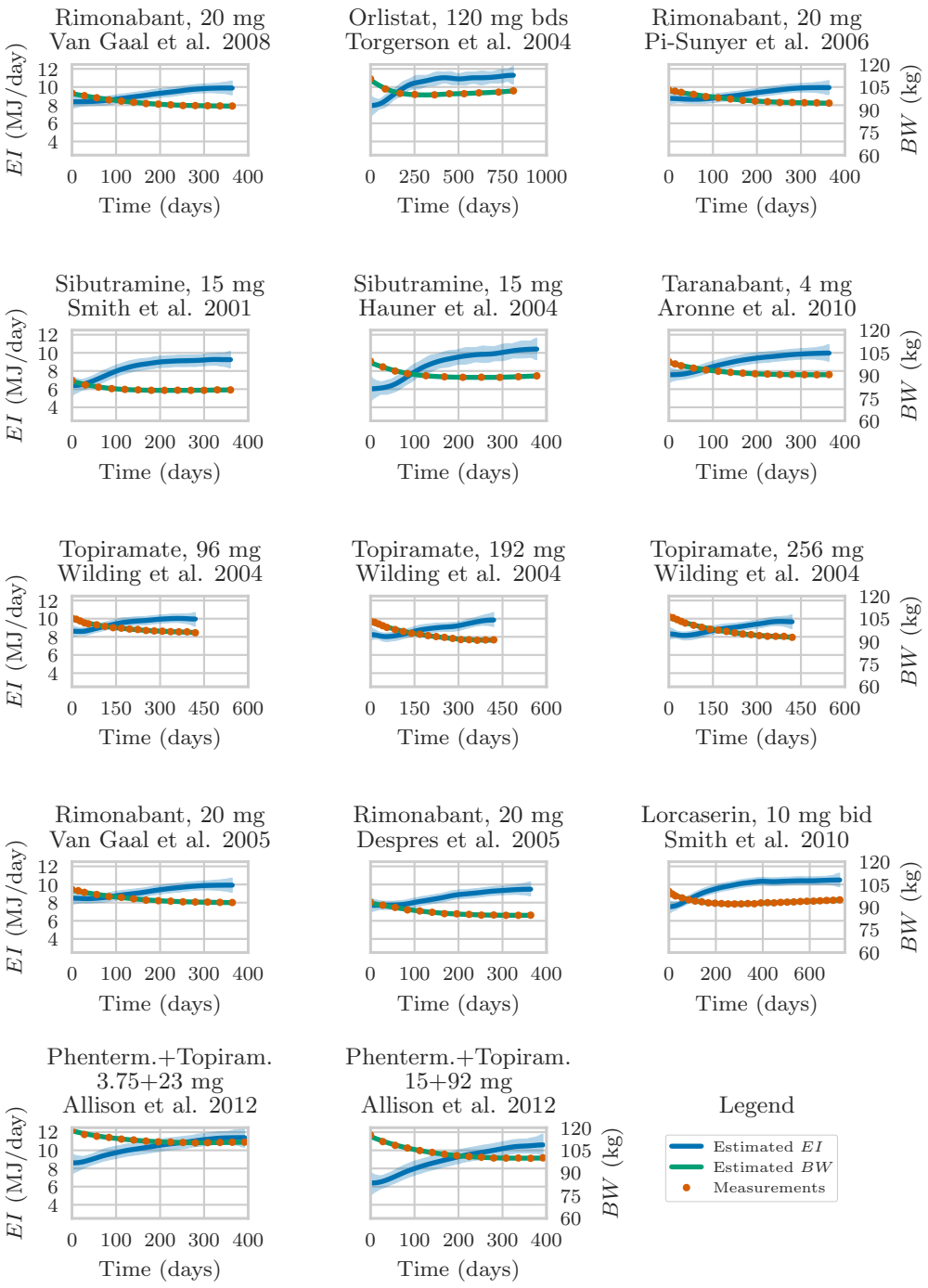

Figure 5.12: MCMC estimation results for the real human data using Karhunen-Loève basis functions and a prior based on the first derivative. In all cases, SMMALA was used as the sampling method.

Chapter 6

Case study — exenatide absorption

6.1 Background

Extended release (ER) formulations are commonly used to enhance the PK properties of drugs. They can increase half-life, allow less frequent drug administration, and increase compliance and quality of life for the patient. They can also increase safety by lowering the maximum required plasma concentration, and allow the use of compounds whose PK properties would otherwise be unsuitable. The formulation is designed to achieve a specified release and absorption rate to optimise the PK. Typically, ER medications are administered orally using either tablets or capsules, or injected as intramuscular or subcutaneous depot formulations.

Oral ER formulations are usually polymeric (Yang and Pierstorff 2012; Arafat 2015) and can be classified into matrix, reservoir (or membrane controlled) and osmotic systems (Khalane et al. 2016; Ratnaparkhi and Gupta Jyoti 2013). The drug-release mechanisms involve drug diffusion, system swelling, erosion and dissolution, or osmotic pressure-induced release (Siepmann and Göpferich 2001; Arifin et al. 2006). For long-term release, over the time scale of several days, intramuscular or subcutaneous administration can be used and include formulation types such as oil-based solutions, drug suspensions, polymer-based microspheres and polymer-based or lipid liquid crystal *in-situ* formings (Rhee et al. 2010; Schwendeman et al. 2014; Gulati and Gupta 2011). Biodegradable microsphere systems, such as those made of poly(lactic-co-glycolic acid) (PLGA) copolymer, have proved to be a successful approach to deliver macromolecular drugs (Mitragotri et al. 2014).

In any ER formulation development process, it is fundamental to determine the *in vivo* drug release/absorption profile of each candidate formulation. This is done routinely in drug discovery and development. Measuring the release profile *in vivo* is

generally difficult and expensive. Typically, the data that are available are plasma concentration profiles following extravascular administration. If a model of the PK is available, it is possible to infer the release profile from such plasma concentration data. The total amount of drug absorbed, and therefore the bioavailability, can be computed by integrating the release profile.

When predicted *in vivo* input profiles are available, it may be possible to validate or invalidate the translatability of the *in vitro* system. Given data for several candidate formulations, an *in vitro-in vivo* correlation (ivivc) can be established, relating the *in vitro* drug dissolution or release to the *in vivo* drug absorption or release (Lu et al. 2011; Cardot and Davit 2012). Ideally, one can then predict the *in vivo* performance based on the *in vitro* release profile and optimise the formulation by *in vitro* testing at low cost. In addition, knowledge of the release profile in an animal model can help in predicting, and hence optimising, the human PK profile. To achieve this, a human intravenous PK model is required, either from real data or predicted from cellular or animal data. The release profile obtained from animal data is fed to the human model, resulting in human PK trajectories. These types of human predictions are always desired in drug discovery to assess feasibility. Naturally, prediction reliability increases with the amount and quality of data.

One way to estimate the release profile is to build a parametric model of the drug release and absorption process. For the drug release process, various models have been proposed, ranging from simple empirical models to detailed mechanistic models that account for various processes such as degradation and erosion (Siepmann and Peppas 2001; Versypt et al. 2013). However, if the release profile is complicated, it may be difficult to create a model that is able to capture the observed plasma concentration (Shen and Burgess 2015). One example is long acting biodegradable particles for subcutaneous injection. The model may also need to be tailored to the particular type of drug and formulation used. For sparse data, such models may also have practical identifiability issues. An alternative is to use nonparametric methods such as the input-estimation methods evaluated in this thesis. This way, the release profile is allowed to take any functional form as long as it matches the data and does not exhibit any unrealistic behaviour, such as taking negative values.

When the time scale of the release and absorption process is significantly larger than the time scale of the PK, is it possible to assume that the system is in steady state, such that the plasma concentration at any time point is only a function of the release rate at that time point, independent of previous history. Replacing a dynamic model with a computationally cheap algebraic model can lead to substantial savings in computational time, especially for non-linear systems which otherwise

would require the use of numerical ODE solvers.

In this chapter, the input-estimation methods are benchmarked on a dataset of plasma concentrations following administration of Bydureon (Buse et al. 2010; Buse et al. 2013), an extended-release microsphere formulation of the GLP-1 receptor agonist exenatide (Buse et al. 2004; DeFronzo et al. 2005). The Bydureon formulation consists of exenatide encapsulated within PLGA microspheres that are designed to release exenatide over an extended period of time, which allows once-weekly patient-administered subcutaneous injections (European Medicines Agency 2011). Typical *in vitro* release curves for Bydureon are given in Fig. 3 in DeYoung et al. 2011. Such curves can be used, together with predicted input profiles from *in vivo* data, to establish an ivivc. In humans, Bydureon exhibits a multiphasic concentration-time profile over approximately 10 weeks consistent with the known mechanism of release from PLGA microspheres. This is characterised by a limited initial rapid release of loosely bound surface exenatide (<1% released in the first few hours) followed by two additional phases corresponding to diffusion and erosion release with peak plasma concentrations at around week 2 and week 7 following intervention (DeYoung et al. 2011). The reason for choosing Bydureon as an example is that it is a drug that is already on the market, and data (Fineman et al. 2011; Li et al. 2015) as well as PK models (Gao and Jusko 2012) are available in the literature. The complicated profile of Bydureon (Fig. 6.1) cannot be easily captured by a simple parametric model. A compartmental model of the release and absorption process has been proposed (Li et al. 2015), where the extended release process is modelled by a cascade of transition compartments, where the initial amount of several compartments is nonzero. However, this model was designed to fit data from multiple-dosing experiments, where the multiple peaks are not as noticeable.

6.2 Model

The model used was developed by Gao and Jusko (2012). Similar models have been developed by Chen et al. (2013) and Li et al. (2015). The main difference between these model structures is in the release and absorption processes. Gao and Jusko (2012) consider an immediate-release formulation with an absorption compartment, while Chen et al. (2013) model the absorption with a Michaelis-Menten function. Li et al. (2015) model the release and absorption of an ER formulation using a linear compartmental model. However, it is not clear whether this model would be able to capture the multiple peaks, as it was only applied to data from experiments with repeated dosing. In this work, the absorption compartments were removed from the

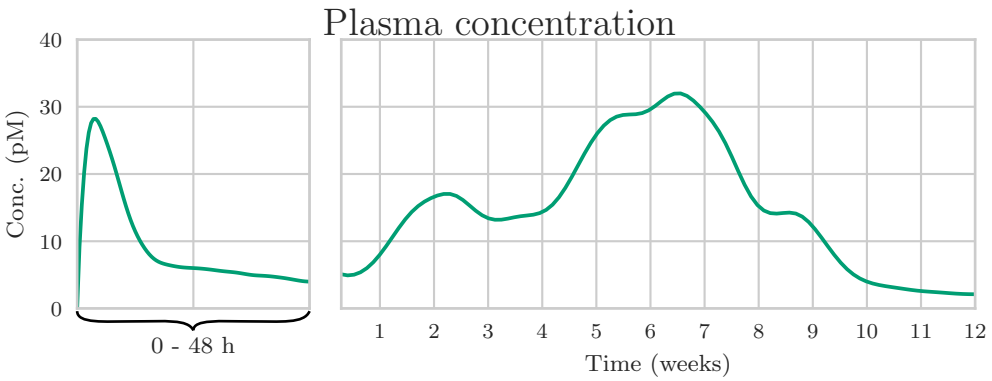

Figure 6.1: Example of a plasma concentration profile after a 10 mg dose of Bydureon. This is a mean prediction of the data in Fineman et al. (2011). The profile shows multiple peaks, making it non-trivial to model using conventional compartmental models.

model, as the purpose of the analysis is to estimate the release and absorption using non-parametric methods. The remaining PK model is a three-compartment model featuring target-mediated drug disposition, where target binding affects the PK. The model is given by:

$$\begin{aligned}\frac{\mathrm{d}C}{\mathrm{d}t} =& \frac{u}{V_c} - (k_{el} + k_{pt}) \cdot C + k_{tp} \cdot \frac{A_T}{V_c} \\ & - k_{on} \cdot (R_{tot} - RC) \cdot C + k_{off} \cdot RC \end{aligned} \tag{6.1a}$$

$$\frac{\mathrm{d}A_T}{\mathrm{d}t} = k_{pt} \cdot C \cdot V_c - k_{tp} \cdot A_T \tag{6.1b}$$

$$\frac{\mathrm{d}RC}{\mathrm{d}t} = k_{on} \cdot (R_{tot} - RC) \cdot C - (k_{off} + k_{int}) \cdot RC \tag{6.1c}$$

where C is the drug molar concentration in the central compartment, A_T is the drug amount in the peripheral compartment, and RC is the molar concentration of the drug-receptor complex. The model parameters are from Gao and Jusko (2012) and are shown in Table 6.1. The model structure is shown in Fig. 6.2.

It can be noted that over sufficiently long time scales, this model can be approximated by a steady-state model, where the plasma concentration is a function only of the current release rate. This is possible as the ER formulation considered here has a release rate over time scales of weeks, while the PK model has considerably faster time constants. This is illustrated in Fig. 6.3. The steady-state approximation can be obtained by setting all time derivatives to zero, and solving for the plasma concentration in the central compartment as a function of the release rate. The

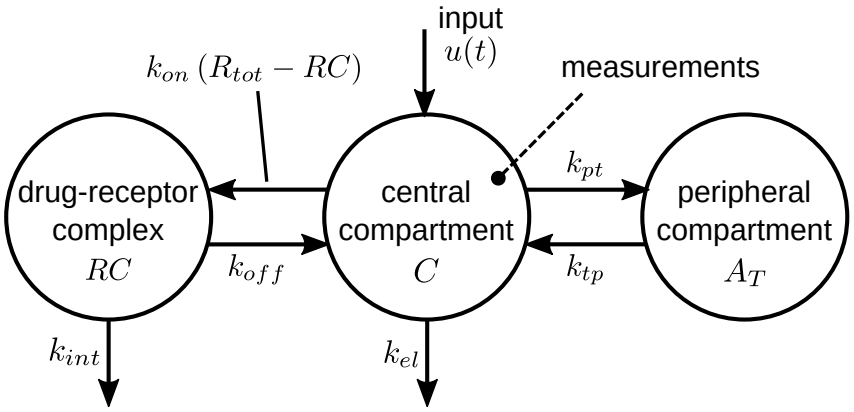

Figure 6.2: The nonlinear three-compartment exenatide model from Gao and Jusko (2012).

Parameter	Definition	Value	Unit
k_{el}	Elimination rate constant	0.013	min^{-1}
k_{pt}	Intercompartmental rate constant	0.0685	min^{-1}
k_{tp}	Intercompartmental rate constant	0.0846	min^{-1}
V_c	Central volume of distribution	111	$\text{mL} \cdot \text{kg}^{-1}$
k_{on}	Second-order binding constant	0.000411	$\text{pM}^{-1} \cdot \text{min}^{-1}$
k_{off}	First-order dissociation constant	0.566	min^{-1}
k_{int}	Internalization rate constant	0.00342	min^{-1}
R_{tot}	Total receptor concentration	1240	pM

Table 6.1: Pharmacokinetic parameters of the exenatide model.

resulting equation is given by:

$$\begin{aligned} C(t) = \frac{1}{2V_c k_{el} k_{on}} \big(&-R_{tot}V_c k_{int} k_{on} - V_c k_{el} k_{int} - V_c k_{el} k_{off} \\ &+ u(t)k_{on} + (R_{tot}^2 V_c^2 k_{int}^2 k_{on}^2 + 2R_{tot}V_c^2 k_{el} k_{int}^2 k_{on} \\ &+ 2R_{tot}V_c^2 k_{el} k_{int} k_{off} k_{on} - 2R_{tot}V_c u(t) k_{int} k_{on}^2 \\ &+ V_c^2 k_{el}^2 k_{int}^2 + 2V_c^2 k_{el}^2 k_{int} k_{off} + V_c^2 k_{el}^2 k_{off}^2 \\ &+ 2V_c u(t) k_{el} k_{int} k_{on} + 2V_c u(t) k_{el} k_{off} k_{on} + u(t)^2 k_{on}^2\big)^{\frac{1}{2}}\big). \end{aligned} \tag{6.2}$$

While this expression may appear complicated, it is a purely algebraic expression that does not require expensive numerical ODE solvers to compute. The steady-state approximation was not used in this analysis, in the interest of being able to compare the performance of the input-estimation methods on the other datasets.

Still, it is worth noting that it may be possible to speed up these computations considerably.

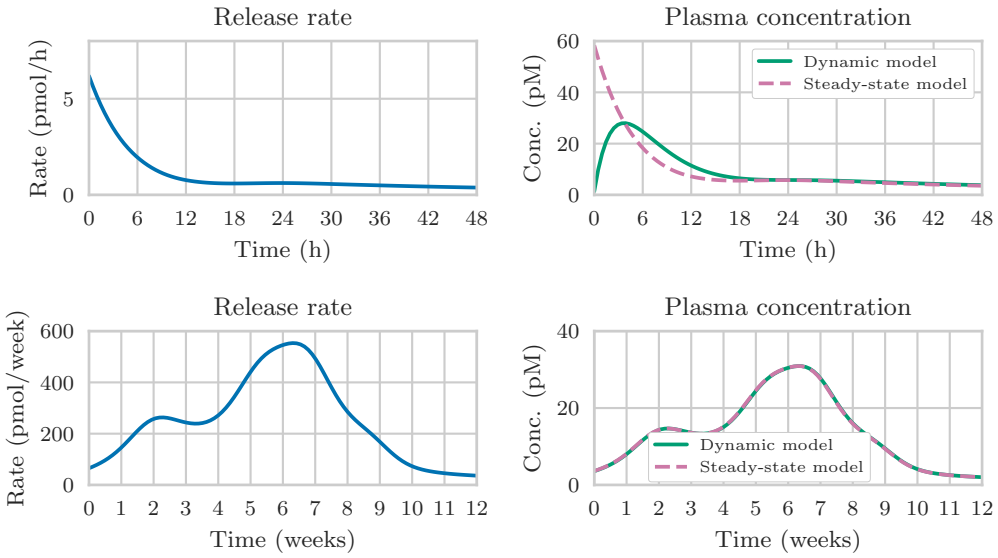

Figure 6.3: Comparison of the dynamical model and the steady-state approximation for short (top row) and long (bottom row) time scales. On short time scales, the model predictions are substantially different, while the predictions are essentially identical for long time scales. This suggests that for long time scales, computation time could be reduced by replacing the expensive ODE-based dynamical model by the computationally cheaper algebraic steady-state model.

6.3 Data

6.3.1 Real data

The data for this analysis are from a study by Fineman et al. (2011). In the study, 54 subjects were administered a single dose of exenatide, in doses of 2.5, 5, 7, 10 mg, or placebo. Plasma concentrations were measured over 12 weeks. The plasma concentration was sampled relatively densely during the initial 48 hours, followed by sampling once a week for the remainder of the study. The group means, which were used for this analysis, are shown in Fig. 6.4 and Tables A.7 and A.8. Note that data for the placebo group were not reported during the initial 48 hours.

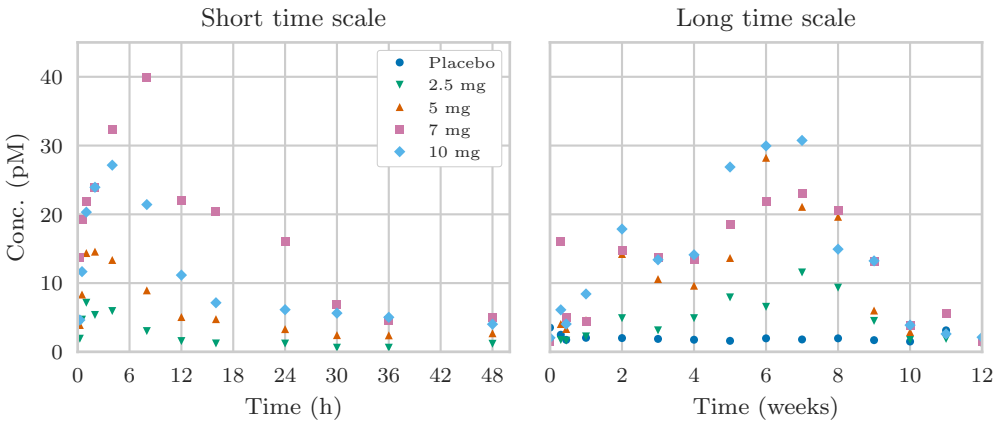

Figure 6.4: Plasma concentration measurements for the exenatide dataset.

6.3.2 Test data

In order to assess the accuracy of the input-estimation methods on a known input, a synthetic input function was generated and used to produce simulated data. The input function was designed to produce data that have similar characteristics to the real data. On longer time scales, the real data have two major peaks, one around 2 and one around 7 weeks. Additionally, there is a fast peak during the first few hours after administration (Fig. 6.4). The long-term characteristics could be captured by the sum of two Erlang distributions, each described by the function

$$u_i(t) = a_i \frac{k_{tri}^{n_i} \cdot t^{n_i - 1} \cdot e^{-k_{tri} \cdot t}}{(n_i - 1)!}, \qquad i = \{1, 2\} \tag{6.3}$$

where k_{tri} is a rate constant that controls the rise and fall times of the peaks, n_i controls the time delay, and a_i is a multiplicative scaling factor. The fast initial peak was modelled by adding a third component $u_3(t)$, described by a biexponential function:

$$u_3(t) = r_1 e^{-k_1 t} + r_2 e^{-k_2 t}. \tag{6.4}$$

The final input function was given by

$$u(t) = u_1(t) + u_2(t) + u_3(t) \tag{6.5}$$

The synthetic input function can be interpreted as the output from a compartmental model, shown in Fig. 6.5. Each Erlang distribution represents a series

of n_i compartments, connected with the rate constant k_{tri}, where the first compartment in the series is initialised with the amount a_i, while the other compartments have zero initial values (Jacquez 1985, Ch. 8). The biexponential part $u_3(t)$ can be interpreted as an absorption compartment combined with a peripheral compartment. The parameters k_{12}, k_{21} and k_a in Fig. 6.5 are related to r_1, r_2, k_1 and k_2 by

$$q = \sqrt{(k_{12} + k_{21} + k_a)^2 - 4k_{21}k_a} \tag{6.6}$$

$$r_1 = \frac{k_a a}{2q}(q + k_{12} - k_{21} + k_a) \tag{6.7}$$

$$r_2 = \frac{k_a a}{2q}(q - k_{12} + k_{21} - k_a) \tag{6.8}$$

$$k_1 = \frac{1}{2}(q + k_{12} + k_{21} + k_a) \tag{6.9}$$

$$k_2 = \frac{1}{2}(-q + k_{12} + k_{21} + k_a) \tag{6.10}$$

where a is the initial amount in compartment 1, and q has been introduced to simplify the expressions. This relationship can be determined by solving the system of differential equations describing the dynamics for compartments 1 and 2, and matching the coefficients.

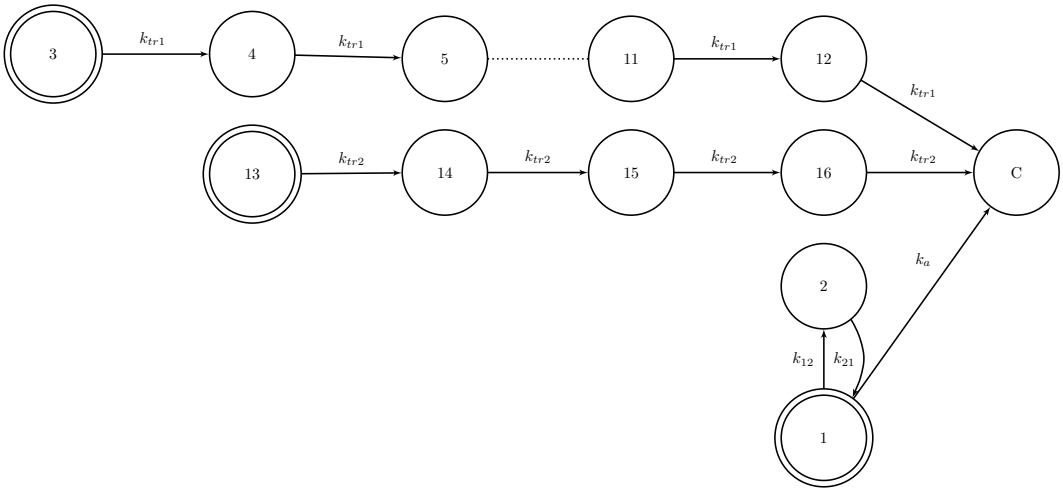

Figure 6.5: Compartmental model interpretation of the test input function. The upper two rows represent the functions $u_1(t)$ and $u_2(t)$, and compartments 1 and 2 represent the function $u_3(t)$. Double circles denote compartments with non-zero initial values. The compartment C is the central compartment of Fig. 6.2.

The input function parameters were chosen to result in simulated plasma concentrations that were similar to the real plasma concentrations following a 10 mg

dose, and are shown in Table 6.2. Separate time series were generated for a short time scale of 48 hours and for a longer time scale of 12 weeks. For each time scale, four synthetic time series were generated:

Densely sampled, no noise The simulated plasma concentration was sampled at 100 time points equispaced between $t = 0$ and $t = 48$ h (short time scale) or $t = 12$ weeks (long time scale).

Densely sampled, with noise The simulated plasma concentration was sampled as above, and Student's t-distributed noise was added to the samples.

Sparsely sampled, no noise The simulated plasma concentration was sampled at the same time points as the real data.

Sparsely sampled, with noise The simulated plasma concentration was sampled as above, and Student's t-distributed noise was added to the samples.

The Student's t-distributed noise had 4 degrees of freedom, and a scale parameter equal to 10% of the true plasma concentration value. This type of noise was used because the MCMC analysis methods use a Student's t likelihood with these parameters. The rationale for this choice of likelihood is described in Section 6.4.2. The test datasets are shown in Figs. 6.6 and 6.7.

Parameter	Interpretation	Value	Unit
n_1	Number of compartments, first chain	10	-
a_1	Initial amount, first chain	2700	pmol
k_{tr1}	Rate constant, first chain	$1.5 \cdot 10^{-4}$	min^{-1}
n_2	Number of compartments, second chain	4	-
a_2	Initial amount, second chain	800	pmol
k_{tr2}	Rate constant, second chain	$1.8 \cdot 10^{-4}$	min^{-1}
r_1	Amplitude, first exponential	0.12	$\text{pmol} \cdot \text{min}^{-1}$
k_1	Rate constant, first exponential	$7.64 \cdot 10^{-3}$	min^{-1}
r_2	Amplitude, second exponential	0.02	$\text{pmol} \cdot \text{min}^{-1}$
k_2	Rate constant, second exponential	$4.76 \cdot 10^{-3}$	min^{-1}

Table 6.2: Synthetic input model parameters for the exenatide model.

6.4 Materials and methods

All analysis was performed separately for the time series over short time scales (48 h) and long time scales (12 weeks). This was deemed necessary, as the different

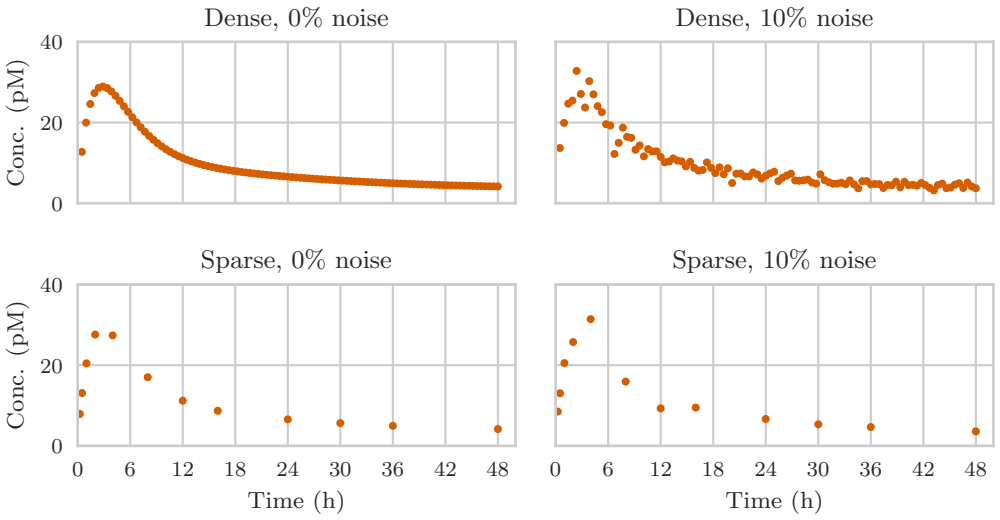

Figure 6.6: The short time scale synthetic test dataset generated for the exenatide model.

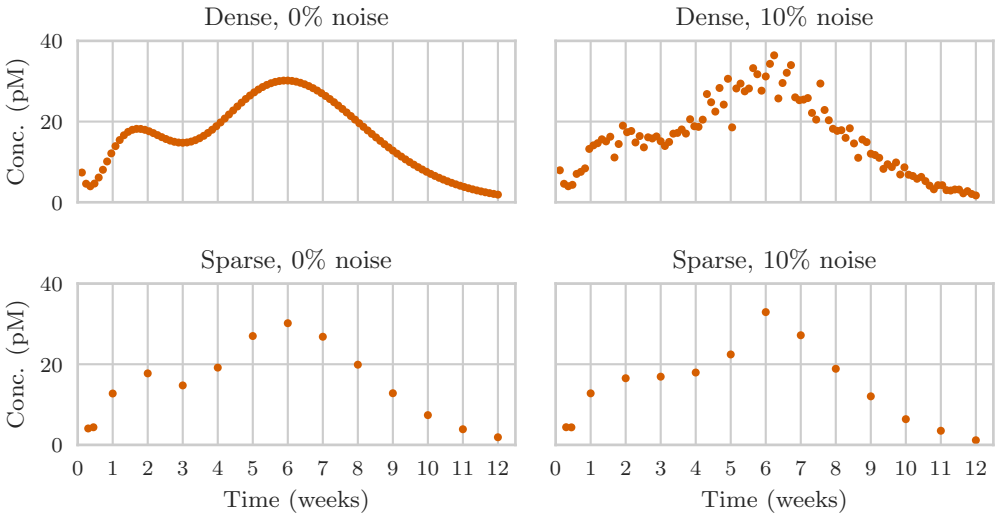

Figure 6.7: The long time scale synthetic test dataset generated for the exenatide model.

time scales involved may have caused computational issues otherwise. For example, when Karhunen-Loève basis functions are used, a large number of functions would be required to accurately capture peaks at time scales that are short compared to the overall time scale. Estimating the initial and later parts of the time series separately

also results in separate priors and regularisation coefficients being assigned to these parts. This can be an advantage when the characteristics of the input function are expected to change during the course of the experiment, as discussed in Section 4.5.1. When performing input estimation on noise-free test data, a proportional noise scaling factor of 0.1% was assumed, as the estimation methods do not permit setting the assumed measurement noise to zero.

6.4.1 Optimal-control methods

Optimal control-based input estimation was performed on all 8 synthetic and 9 real time series. In all cases, a basis of 100 equispaced piecewise constant functions was used. Five priors were used: penalisation of the first and the second derivative of the function as well as of the logarithm of the function, and the maximum entropy-based prior. When penalisation of a derivative of the function was employed, nonnegativity constraints were added. Penalisation was performed on on a finite-difference approximation of the derivatives, which is

$$E_R = \int_{t_i}^{t_f} \left(\frac{\mathrm{d}u(t)}{\mathrm{d}t} \right)^2 \mathrm{d}t \approx \sum_{k=0}^{N_B-2} \frac{(u_{k+1} - u_k)^2}{\Delta t} \tag{6.11}$$

for the first derivative, and

$$E_R = \int_{t_i}^{t_f} \left(\frac{\mathrm{d}^2u(t)}{\mathrm{d}t^2} \right)^2 \mathrm{d}t \approx \sum_{k=1}^{N_B-2} \frac{(u_{k+1} - 2u_k + u_{k-1})^2}{(\Delta t)^3} \tag{6.12}$$

for the second derivative. Here, Δt is the length of each discretisation interval. Three optimal-control methods were used: single shooting, multiple shooting and collocation. The shooting and collocation intervals were chosen to coincide with the basis function intervals. For collocation, polynomials of degree 3 with Radau collocation points were used. The amount of regularisation was determined by using the discrepancy criterion and treating the optimal-control problem as a constrained optimisation

Method	Decision variables	Equality constraints	Inequality constraints
Single shooting	100	0	1/101
Multiple shooting	400	300	1/101
Collocation	1300	1200	1/101

Table 6.3: Number of decision variables and constraints for the optimal-control problem formulations of the exenatide model. The number of inequality constraints are either 1 or 101, depending on whether nonnegativity of the input function is enforced.

problem of the same type as as Chapter 4:

$$\begin{aligned} \underset{\mathbf{a}}{\text{minimise}} \quad & E_R(\mathbf{a}) && \text{(6.13a)} \\ \text{such that} \quad & \chi^2 \leq n && \text{(6.13b)} \\ & \mathbf{x}(t_i) = \mathbf{x}^{(0)} && \text{(6.13c)} \\ \text{and} \quad & \mathbf{h}(\mathbf{a}, \mathbf{x}(t)) \leq \mathbf{0} && \text{(6.13d)} \\ \text{where} \quad & \mathbf{x}(t) = \Phi(t_i, t, \mathbf{x}^{(0)}, \mathbf{a}). \end{aligned}$$

In all cases, the initial value of all compartments was assumed to be zero.

Table 6.3 shows the number of variables and constraints for the optimisation methods. For single shooting, the number of variables is equal to the number of basis functions. For multiple shooting, variables and constraints representing the state variables at the beginning of each time interval are added to the optimisation problem. When collocation is used, additional variables and constraints are added for the state variables at the collocation points.

In summary, $8 + 9$ time series were analysed using 3 optimisation methods and 5 priors, making for a total of $(8 + 9) \times 3 \times 5 = 255$ analyses.

6.4.2 MCMC methods

For the MCMC evaluation, four priors were used: penalisation of the first and second derivative of the function as well as of its logarithm. Two function parameterisations were tested: Karhunen-Loève functions with 20 basis functions, and cubic B-splines. One B-spline knot was placed at each data point. Additional knots were added at the start and end points to ensure that each interval was covered by 4 basis functions. For densely sampled test data, knots were placed at the measurement points of the real data, as the dimensionality of the problem would be excessive otherwise.

The regularisation parameter τ was assigned a Gamma prior with hyperparameters $\alpha = 10^{-3}$, $\beta = 10^{-3}$, for consistency with the other chapters. Four MCMC sampling methods were tested to update the basis function coefficients: single-component RWMH, block RWMH, MALA and SMMALA. The regularisation parameter was updated using Gibbs sampling. For each combination of prior, parameterisation, sampling method, and data, the estimation procedure was allowed to run for 5 minutes of processing time.

Initialisation of the Markov chain was performed using the results from the optimal-control methods, using the same methodology as in Section 4.4.2. For the sparse noise-free test data, no successful results from the optimal-control methods were available. In these cases, initial values were provided by optimal-control procedures which assumed a measurement noise standard deviation of 10%.

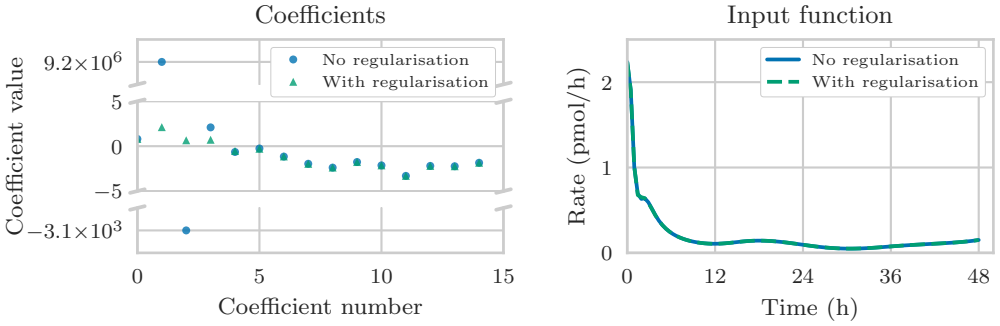

Figure 6.8: The effect of regularisation when converting piecewise constant basis functions to B-spline coefficients. Left: The function coefficients for the regularised solution attain absolute values which are no larger than 5, while the function coefficients for the unregularised solution have two values that are orders of magnitude larger. Right: The resulting input functions are almost identical.

The piecewise-constant parameterisations of the optimal-control methods were transformed to the parameterisations of the MCMC methods by least squares-fitting, similar to the procedures of Chapters 4 and 5. For B-splines, it was found that this reparameterisation was ill-conditioned in certain cases. This resulted in parameters attaining values that are several orders of magnitude larger than is necessary for accurately reproducing the original function. This behaviour was considered undesirable, as the differences in magnitude may cause problems for the sampling methods. It was found that this could be remedied by adding a quadratic regularisation term to the least-squares cost, with a small regularisation parameter on the order of 10^{-10}. This is illustrated in Fig. 6.8. Note that this regularisation

term has a different purpose than the regularisation term in the input estimation. In the former case, it is used to combat numerical issues, while it in the latter case is used to avoid unrealistic oscillatory solutions.

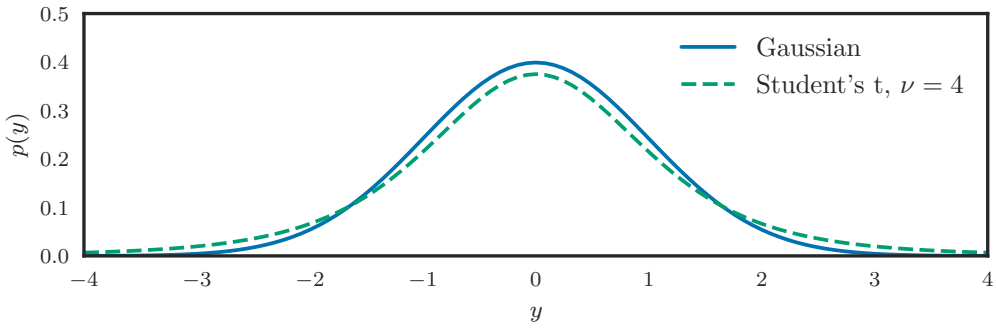

Figure 6.9: Comparison of a Gaussian and a Student's t-distribution with 4 degrees of freedom. While the distributions are visually similar, the Student's t-distribution has heavier tails, and will assign a higher probability density to outliers than the Gaussian distribution.

Originally, the likelihood was assumed to be Gaussian. However, it was found that this choice rendered the estimation methods sensitive to outliers in the data. The light tails of the Gaussian distribution make measurements far from the prediction very improbable under the model. Therefore, the estimated input function needs to result in predictions which are close to the measurements at every measurement time point in order to have a non-negligible probability density. When an outlier is present, the input function may need to be highly oscillatory in order to make the predictions fit the data, which will lead to the basis function coefficients having a distribution with a large standard deviation. Since the regularisation parameter is estimated from the distribution of the basis function coefficients, this will cause the estimated regularisation parameter to be set to a very low value. This low regularisation parameter will make the model assign non-negligible probability densities to a wide range of oscillatory functions. As a result, a single outlier can cause the uncertainty regions to become unrealistically large over the whole time series. For this reason, it was decided to use a Student's t likelihood with a small number of degrees of freedom, which has heavier tails (Fig. 6.9) and thus allows larger deviations from the prediction (Gelman et al. 2013). This is illustrated in Fig. 6.10. The likelihood for the jth measurement y_j is given by:

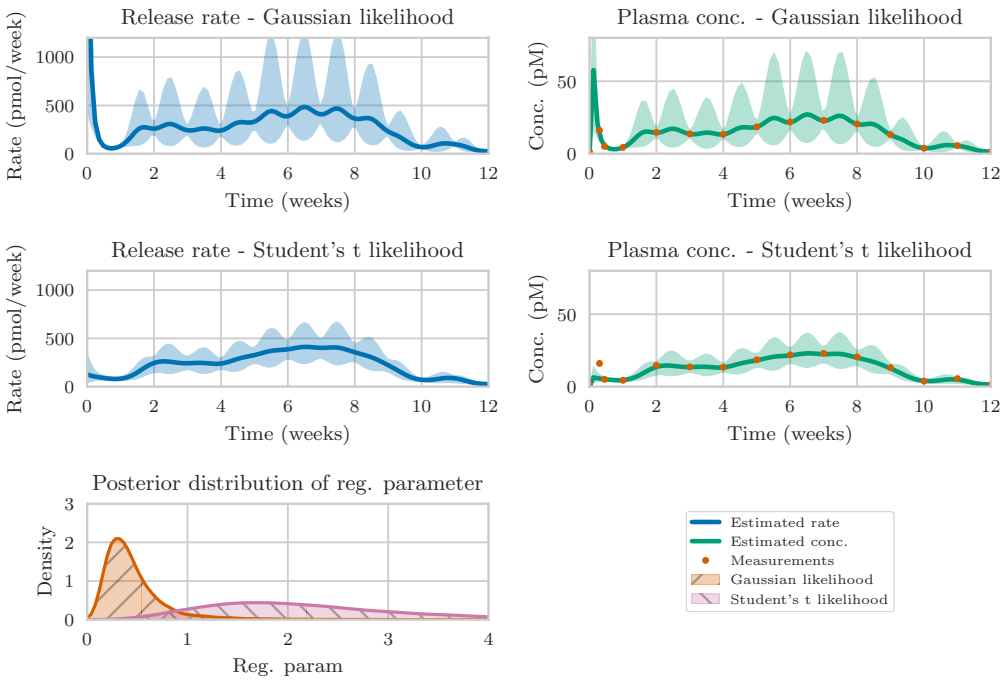

Figure 6.10: Comparison of MCMC-based input estimation using a Gaussian likelihood (top row) and a Student's t likelihood with 4 degrees of freedom (middle row). For the Gaussian likelihood, the measurement at time $t = 0.3$ weeks $= 50$ hours drives down the value of the regularisation parameter, and causes large uncertainties in the prediction. When a Student's t likelihood is used, the prediction uncertainty is decreased, at the expense of the measurement at $t = 0.3$ being ignored. The kernel density estimates show how different likelihoods result in different estimates of the regularisation parameter. The example shows data for the 7 mg dose, analysed by SMMALA, using Karhunen-Loève basis functions and penalisation of the first derivative of the logarithm of the input.

$$p(y_j|\nu, y_j^{(pred)}, \sigma_j) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \cdot \frac{1}{\sqrt{\nu\pi\sigma_j^2}} \cdot \left(1 + \frac{1}{\nu} \cdot \frac{\left(y_j^{(pred)} - y_j\right)^2}{\sigma_j^2}\right)^{\frac{\nu+1}{2}} \tag{6.14}$$

where ν is the number of degrees of freedom, $y_j^{(pred)}$ is the predicted plasma concentration and σ_j is a scaling factor for the standard deviation that was set to 10% of the predicted plasma concentration in order to make the model comparable to the eflornithine model. A number of estimation procedures was performed using $\nu = 4$ and $\nu = 6$ in order to assess the sensitivity of the model to the number of degrees of

freedom. As the estimation results showed only minor differences, $\nu = 4$ was chosen for all reported results.

In summary, 8 synthetic and 9 time series were analysed using 4 choices of priors, 2 choices of basis functions, and 4 choices of sampling methods, making for $(8 + 9) \times 4 \times 2 \times 4 = 544$ estimation procedures.

6.5 Results and discussion

6.5.1 Optimal-control methods

The accuracies of the optimal-control based methods were assessed on the test data by computing RMSE values of the estimates in relation to the true input. Additionally, the estimated total absorbed amount was compared to the true amount, as this number is often of primary interest in practical applications, where an estimate of the bioavailability is desired. The true total amount was obtained by integrating Eq. (6.5) from $t = 0$ up to the time of the last measurement. The accuracy measures are shown in Tables 6.4 and 6.5.

For the long-term data, the initial peak occurs on such short time scales that it is not visible in the data. Evidently, no estimation method was able to capture this peak. As a result, the contribution from this peak dominated the RMSE values, making any meaningful comparisons between methods impossible. For this reason, the results for the first day were removed before the RMSE was computed. Although this makes for a fairer comparison, it is important to point out that in the presence of such peaks, no method is likely to perform well.

For both short- and long-term data, the choice of prior had little or no effect on the result when sampling was dense, regardless of the noise level. For sparse data, the choices of prior did influence the results, although more data would be required in order to draw definite conclusions. In the case of short-term data, log-domain modelling may result in higher accuracy, although the difference in accuracy may not be large enough to allow any strong conclusions. It can be noted that the short-term test data were generated by a sum of exponential functions, which translate to straight lines in the log domain. As a consequence, the true solution was assigned a low cost under the log priors. Since little can be assumed about the release process of the drug, it is difficult to assess which priors are justified from a statistical soundness perspective. However, the estimates for different priors are relatively similar. Hence, any prior might be able to produce sufficiently reliable results.

Robustness and performance summaries are shown in Tables 6.6–6.9. In terms of robustness, the optimisation methods performed similarly on the short-term

Dataset	Prior	RMSE (pmol/h)	Normalised total amount
Dense, 0% noise	1der	0.11	1.00
	1der, log	0.11	1.00
	2der	0.11	1.00
	2der, log	0.11	1.00
	entropy	0.11	1.00
Dense, 10% noise	1der	0.21	0.95
	1der, log	0.22	0.96
	2der	0.22	0.96
	2der, log	0.25	0.96
	entropy	0.23	0.96
Sparse, 0% noise	1der	-	-
	1der, log	-	-
	2der	-	-
	2der, log	-	-
	entropy	-	-
Sparse, 10% noise	1der	0.29	1.00
	1der, log	0.15	1.00
	2der	0.30	1.01
	2der, log	0.18	0.99
	entropy	0.33	1.02

Table 6.4: Accuracy measures of the optimal-control methods for the short-term exenatide test data series. Missing entries mean that no estimation procedures were successful for the given time series and prior.

data, with a success rate of 80–85%. For the long-term data, both shooting methods performed poorly, failing in half of the cases, while collocation was relatively successful. In all cases, collocation was significantly faster than the shooting methods.

Figures 6.11 and 6.12 show estimation results for the sparse noisy test data, which was deemed to be the most realistic test case. The short-term plots show how the log-domain priors are able to capture the peak relatively accurately, while the other priors tend to have too low initial release rates. For the long-term data, it can be seen that when penalising the first derivative, the optimal solution is essentially a piecewise linear function, whose knots are placed at the measurement times. This can be explained by the steady-state nature of the model. The plasma concentration at a time point is solely determined by the release rate at that time point, making the likelihood a function of only the input at the measurement time points. The cheapest way under this prior of connecting any two points is by a straight line (Kimeldorf

Dataset	Prior	RMSE (pmol/week)	Normalised total amount
Dense, 0% noise	1der	1.40	0.99
	1der, log	1.85	0.99
	2der	1.37	0.99
	2der, log	1.37	0.99
	entropy	1.36	0.99
Dense, 10% noise	1der	22.85	0.95
	1der, log	22.20	0.97
	2der	20.45	0.96
	2der, log	22.29	0.97
	entropy	21.35	0.97
Sparse, 0% noise	1der	14.43	0.98
	1der, log	15.62	0.97
	2der	9.26	0.99
	2der, log	9.90	0.99
	entropy	8.62	0.99
Sparse, 10% noise	1der	55.73	0.88
	1der, log	40.55	0.91
	2der	40.64	0.92
	2der, log	37.54	0.94
	entropy	37.74	0.93

Table 6.5: Accuracy measures of the optimal-control methods for the long-term exenatide test data series.

and Wahba 1970).

	Median time (s)	Median iterations	Successful runs
Single shooting	29.1	24	34/40
Multiple shooting	12.2	23	33/40
Collocation	0.1	20	33/40

Table 6.6: Median running times, number of iterations, and the proportion of successful estimation procedures for the short-term exenatide time series, organised by optimisation method.

Figures 6.13 and 6.14 show examples of optimal-control-based input estimation for the entropic prior. It can be seen that the amount of smoothing is relatively small, with the predictions adhering closely to the measurements, except for intervals of dense sampling. As a consequence, some of the predictions exhibit large oscillations, which is most clearly seen for the short-term 7 mg time series.

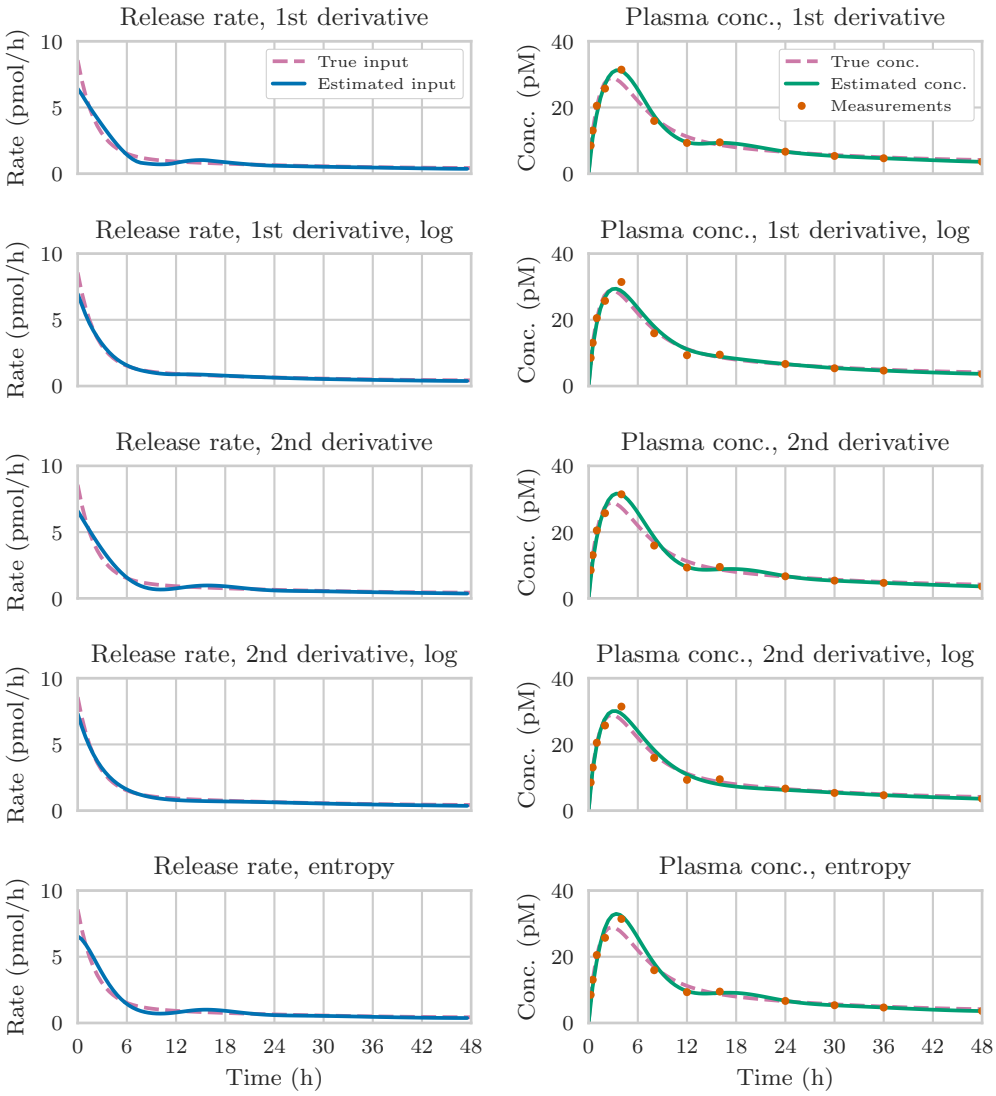

Figure 6.11: Comparison of optimal-control results for various priors using the sparse, noisy short-term test time series.

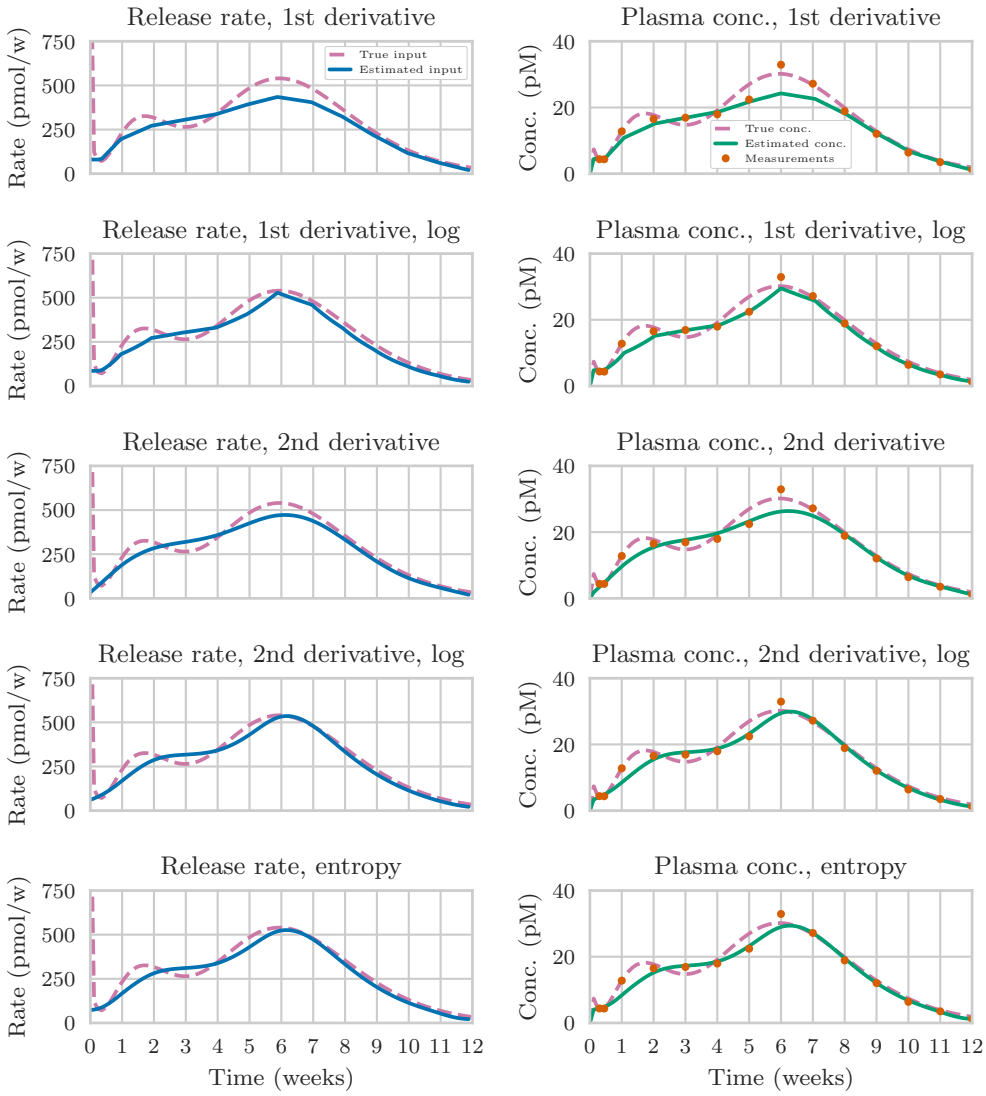

Figure 6.12: Comparison of optimal-control results for various priors using the sparse, noisy long-term test time series.

	Median time (s)	Median iterations	Successful runs
1der	12.6	21	21/24
1der, log	8.0	21	19/24
2der	15.9	22	21/24
2der, log	17.9	30	18/24
entropy	10.3	21	21/24

Table 6.7: Median running times, number of iterations, and the proportion of successful estimation procedures for the short-term exenatide time series, organised by choice of prior.

	Median time (s)	Median iterations	Successful runs
Single shooting	85.0	35	20/45
Multiple shooting	57.9	47	23/45
Collocation	0.2	23	41/45

Table 6.8: Median running times, number of iterations, and the proportion of successful estimation procedures for the long-term exenatide time series, organised by optimisation method.

	Median time (s)	Median iterations	Successful runs
1der	70.2	37	10/27
1der, log	60.0	43	17/27
2der	72.8	44	9/27
2der, log	38.4	35	21/27
entropy	30.2	25	27/27

Table 6.9: Median running times, number of iterations, and the proportion of successful estimation procedures for the long-term exenatide time series, organised by choice of prior.

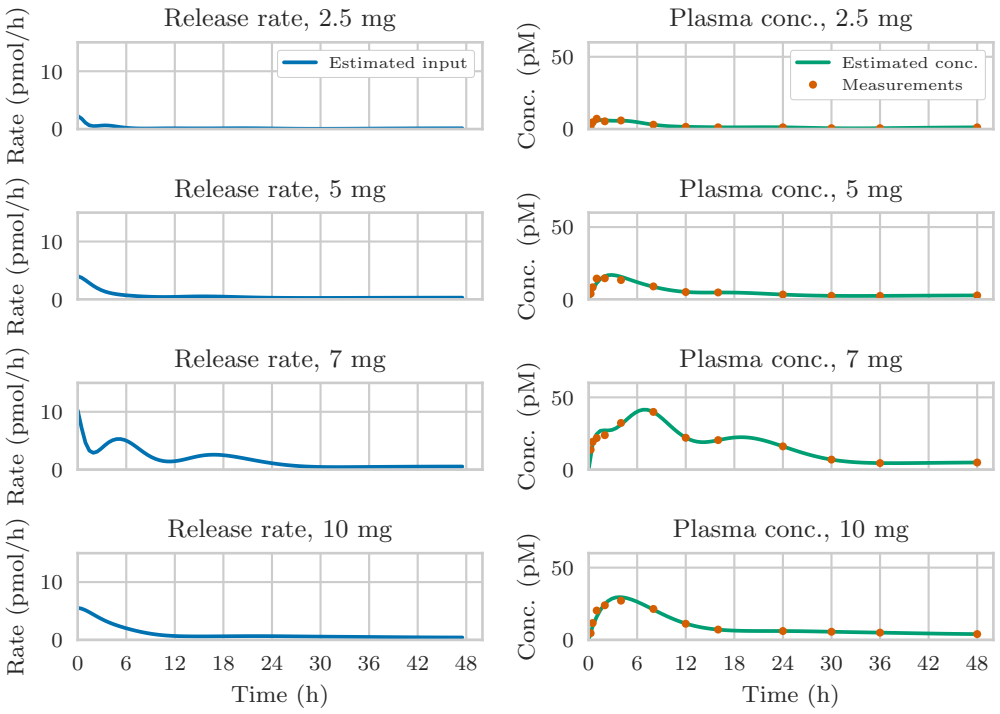

Figure 6.13: Example results from performing input estimation on the real short-term data, using an entropic prior.

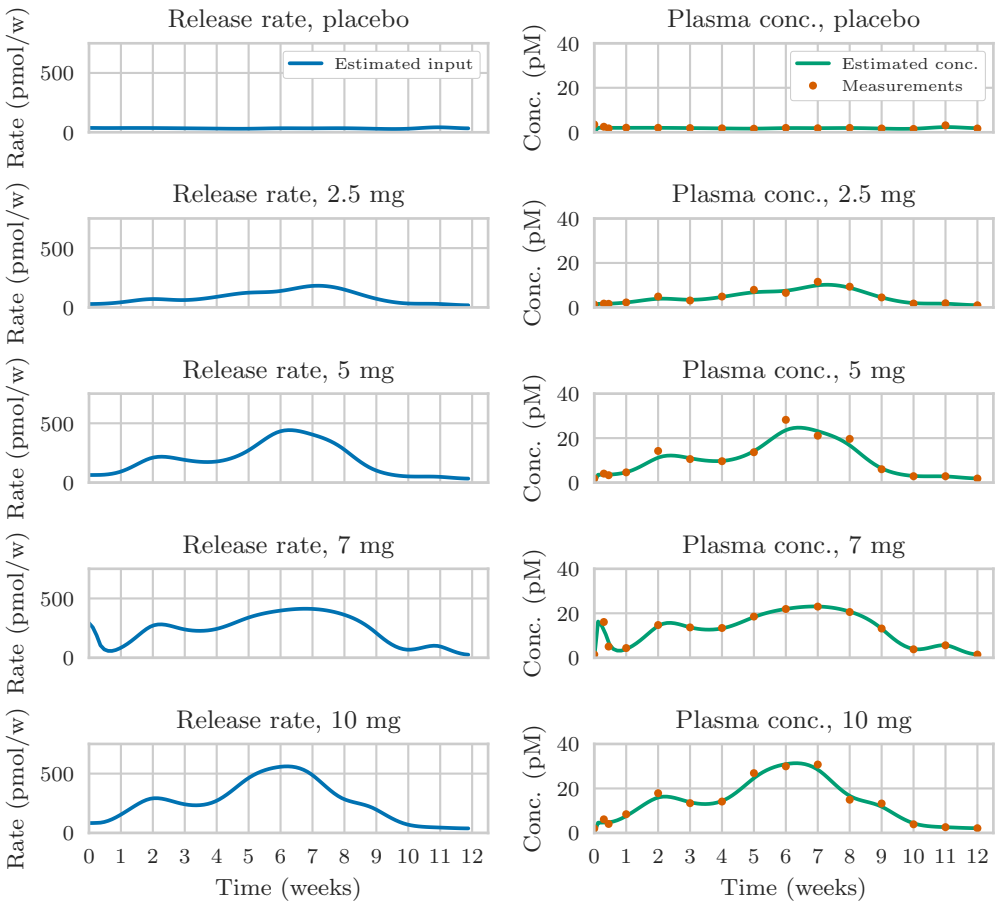

Figure 6.14: Example results from performing input estimation on the real long-term data, using an entropic prior.

6.5.2 MCMC methods

Summaries of the accuracy measures of the MCMC estimation procedures for the test datasets are shown in Table 6.10 for the short-term data and in Table 6.11 for the long-term data. The summaries show the mean RMSE values of the trajectories, as well as mean and RMSE values for the estimates of the total amount. As in the case of evaluating the optimal-control results, the initial time points for the long-term data were omitted from the RMSE computations in order to ensure that differences between the priors were visible.

Dataset	Method	Basis	Prior	RMSE pmol/h	Total amount normalised mean	Total amount normalised RMSE
Dense,	Block RWMH	KL	1der	0.30	1.00	0.017
10% noise	Block RWMH	KL	1der, log	0.17	1.01	0.020
	Block RWMH	KL	2der	0.25	1.00	0.017
	Block RWMH	KL	2der, log	0.11	1.02	0.021
	Block RWMH	spline	1der	0.55	0.95	0.059
	MALA	KL	1der	0.30	1.00	0.017
	MALA	KL	1der, log	0.17	1.01	0.021
	MALA	spline	1der	0.55	0.95	0.058
	SMMALA	KL	1der	0.31	1.00	0.017
	SMMALA	KL	1der, log	0.17	1.01	0.020
	SMMALA	KL	2der	0.25	1.00	0.017
	SMMALA	KL	2der, log	0.12	1.02	0.022
	SMMALA	spline	1der	0.55	0.95	0.058
	SMMALA	spline	2der	0.31	1.02	0.031
Sparse	Single RWMH	KL	1der, log	0.28	1.03	0.057
10% noise	Block RWMH	KL	1der	0.85	1.04	0.184
	Block RWMH	KL	1der, log	0.28	1.03	0.057
	Block RWMH	spline	1der	0.85	0.99	0.220
	MALA	KL	1der	0.85	1.04	0.176
	MALA	KL	1der, log	0.27	1.03	0.056
	MALA	spline	1der	0.86	1.00	0.227
	SMMALA	KL	1der	0.86	1.03	0.181
	SMMALA	KL	1der, log	0.28	1.03	0.054
	SMMALA	KL	2der	0.49	1.04	0.128
	SMMALA	KL	2der, log	0.18	1.02	0.046
	SMMALA	spline	1der	0.85	1.00	0.222

Table 6.10: Accuracy measures of the MCMC methods on the short-term exenatide test datasets.

Similar to Chapters 4 and 5, no estimation procedures were successful for the

Dataset	Method	Basis	Prior	RMSE pmol/week	Total amount normalised mean	Total amount normalised RMSE
Dense,	Block RWMH	KL	1der, log	25.11	0.98	0.028
10% noise	Block RWMH	KL	2der, log	21.94	0.97	0.032
	MALA	KL	1der, log	25.20	0.98	0.028
	MALA	KL	2der, log	21.79	0.97	0.031
	SMMALA	KL	1der	22.46	0.98	0.028
	SMMALA	KL	1der, log	25.07	0.98	0.028
	SMMALA	KL	2der	20.42	0.97	0.033
	SMMALA	KL	2der, log	21.83	0.97	0.031
	SMMALA	spline	2der	20.56	0.98	0.025
Sparse	Block RWMH	KL	1der, log	69.88	0.95	0.064
10% noise	Block RWMH	KL	2der, log	48.80	0.93	0.083
	MALA	KL	1der, log	70.49	0.95	0.066
	MALA	KL	2der, log	47.32	0.93	0.077
	SMMALA	KL	1der	49.96	0.94	0.076
	SMMALA	KL	1der, log	69.46	0.96	0.063
	SMMALA	KL	2der	46.21	0.92	0.090
	SMMALA	KL	2der, log	49.16	0.93	0.083

Table 6.11: Accuracy measures of the MCMC methods on the long-term exenatide test datasets.

assumed noise-free test data. All of the following discussion refers to noisy data.

For the short time scale data, it is evident that log-domain priors consistently performed somewhat better than their linear-domain counterparts in terms of RMSE, regardless of the function parameterisation. The mean estimated total amount appears to be relatively insensitive to the choice of prior. However, the RMSE of the total amount estimate was considerably smaller for log priors than linear priors when the data were sparse. The differences between penalisation of the first and second derivative were relatively minor. There do not appear to be any major differences between the Karhunen-Loève and B-spline parameterisations. However, the number of comparisons that can be made is not very large, as few B-spline estimation procedures were successful. In particular, no log-domain prior B-spline results are available.

For the long time scale data, the effect of the prior on the estimation accuracy is inconclusive. For dense data, all priors achieved similar results. For sparse data, differences did exist, but were relatively minor. It can also be noted that the prior that achieved the lowest RMSE did not always achieve the best estimate of the total amount. In all cases, the total amount was underpredicted to some extent.

Example plots showing estimation results for sparse noisy short-term test data are shown in Fig. 6.15. From these plots, it is clear that when the prior allows negative values of the input function, the uncertainty regions can become relatively large. These considerations are similar to those presented in Chapter 4. In contrast, log-domain priors appear to work well. In particular, penalisation of the second derivative in the log domain produces a credible interval which covers the true input function, despite being very narrow. It is clear that log-domain priors should be recommended for analysis of the short-term data.

Similar plots are shown for the long-term test data in Fig. 6.16. For these data, nonnegativity constraints are generally not necessary — the data do not support negative input values in any case. This may be an explanation for why log-domain priors provided better accuracy for the short-term, but not for the long-term, data. All of the priors cover the true input function by their credible intervals, excluding the initial release peak, which is not visible in the data. The mean predictions appear to be relatively insensitive to the choice of prior. However, the estimated uncertainty is significantly larger for certain priors. This is particularly visible in the credible interval resulting from penalising the first derivative in the log-domain, which appears to be overly cautious. It can be noted that this prior had a considerably higher mean RMSE value than the other priors, which can be seen as an indication that mean RMSE is an appropriate measure of estimation accuracy. When a conservative estimate is desired, one option is to perform estimation using several priors, and report the largest obtained uncertainty. This may be the most satisfactory option from a statistical soundness perspective.

For the cases where multiple sampling methods provided results for the same data, prior, and parameterisations, the accuracy measures agree very well, increasing confidence that the sampling was able to explore the posterior. However, in some cases no comparisons are available, such as linear-domain long-term test data.

Note that the mean predictions from the MCMC estimation methods can be noticeably different from the predictions of the optimal-control methods. This can be clearly seen when comparing the results for penalisation of the first derivative for the long-term data. While the optimal-control estimates were piecewise linear functions, the mean predictions from the MCMC methods are much smoother. One reason for this discrepancy is that the optimal-control methods provide MAP estimates, which can be expected to be different from the pointwise mean predictions of the MCMC methods. Additionally, neither the Karhunen-Loève nor the B-spline basis functions are able to represent functions whose derivatives are discontinuous.

Measures of speed and robustness are shown in Table 6.12 for the short-term

		Median min ESS	Median max RL	Median required time	Successful runs
Single RWMH	KL	4	238721	5304	3/24
	spline	6	255575	3502	0/24
Block RWMH	KL	229	141600	196	12/24
	spline	11	1284669	2151	5/22
MALA	KL	66	40268	674	9/24
	spline	4	296641	3960	5/24
SMMALA	KL	701	5904	82	21/24
	spline	17	158546	2744	7/24

Table 6.12: Performance and robustness results for MCMC methods on the noisy exenatide short-term datasets.

		Median min ESS	Median max RL	Median required time	Successful runs
Single RWMH	KL	7	156353	3633	1/28
	spline	4	282334	5323	0/28
Block RWMH	KL	17	627699	1041	11/28
	spline	6	1394684	2226	3/28
MALA	KL	34	48739	763	11/28
	spline	5	328016	5224	1/28
SMMALA	KL	659	4895	92	25/28
	spline	12	127647	3401	4/28

Table 6.13: Performance and robustness results for MCMC methods on the noisy exenatide long-term datasets.

and Table 6.13 for the long-term data. The quality of the generated samples was assessed using the Raftery-Lewis diagnostics and the effective sample size. In both tables, results are reported for sparse, noisy data, including test as well as real data. It is evident that B-spline parameterisations do not result in a satisfactory performance. For Karhunen-Loève basis functions, it is clear that SMMALA sampling outperforms all of the other sampling methods. It is interesting to note that these results are similar for short- and long-term data. This holds despite the fact that the time scales and the input functions are very different. As the Karhunen-Loève functions also appear to provide more accurate estimates than B-splines, this combination of parameterisation and sampling method should be preferred.

Comparing the densely and sparsely sampled test data, it can be seen that sparse data result in lower estimation accuracy, but not to a very large extent. RMSE values typically differ by a factor of 2–3. It appears that sparsity does not affect the

sampling performance — the number of successful estimation procedures are similar for dense and sparse sampling. This result holds for both the short- and long-term data.

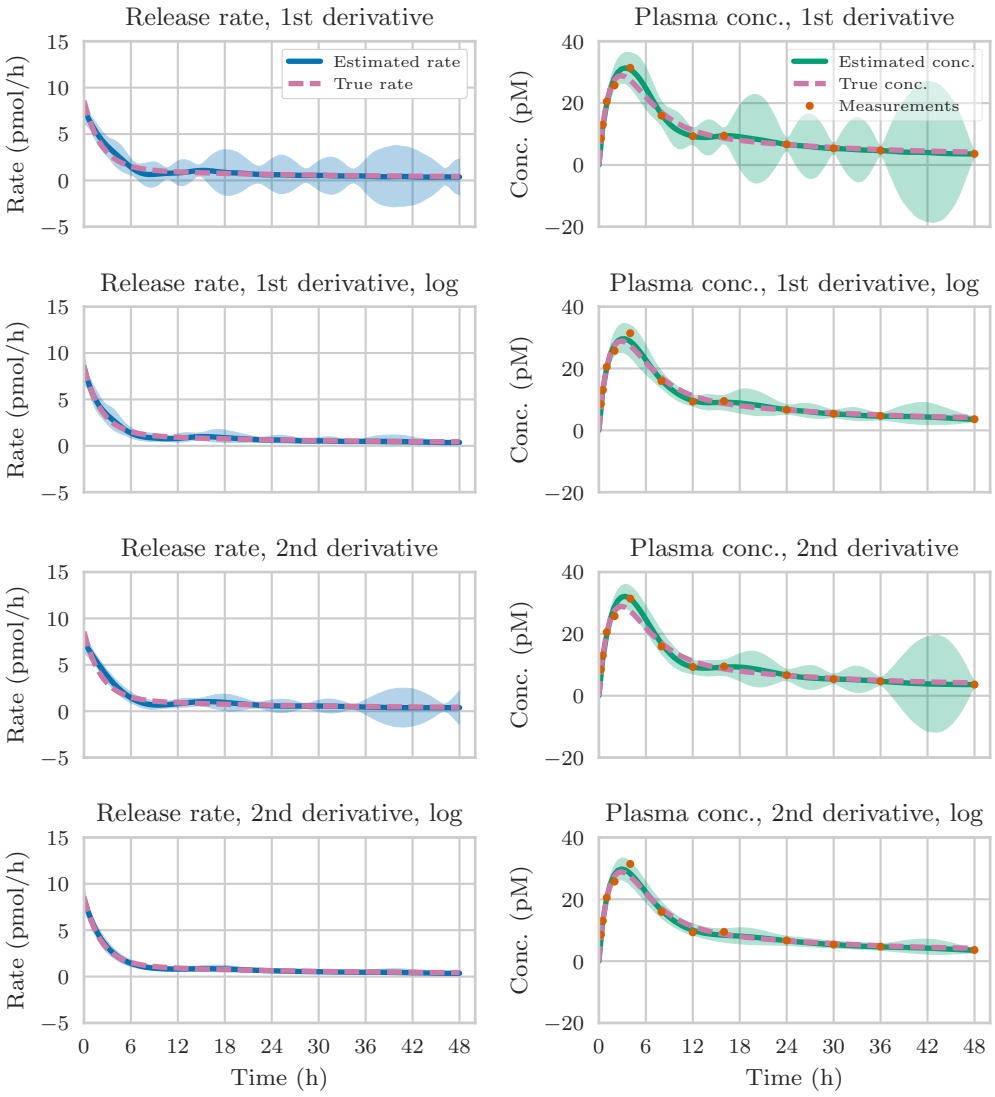

Figure 6.15: Examples of MCMC estimation results for the sparse noisy short-term test data for various priors using Karhunen-Loève basis functions. All results shown were obtained by SMMALA sampling.

Finally, examples of estimation results for the real data are shown in Figs. 6.17 and 6.18. Kernel density estimates of the posterior distribution of the bioavailability are presented in Fig. 6.19. It can be noted that the estimated bioavailabilities,

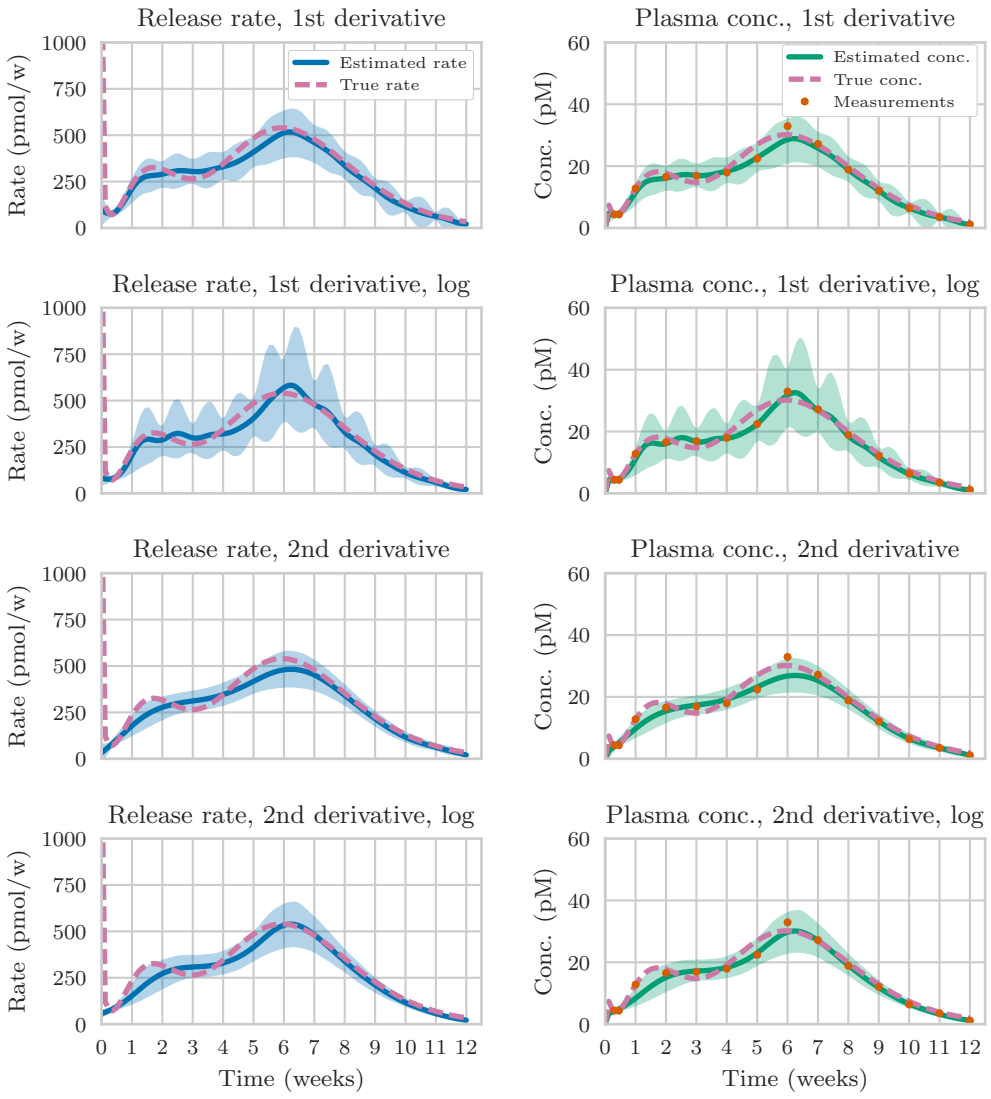

Figure 6.16: Examples of MCMC estimation results for the sparse noisy long-term test data for various priors using Karhunen-Loève basis functions. All results shown were obtained by SMMALA sampling.

ranging between 0.1 and 0.15, are significantly lower than previously reported values of 0.22–0.25 (European Medicines Agency 2011). There are several possible explanations for this. The previously reported figures refer to bioavailability relative to a subcutaneous administration of an immediate-release formulation, rather than absolute bioavailability. Additionally, there may be large uncertainties in the model parameter values. The models of Gao and Jusko (2012), Chen et al. (2013), and

Li et al. (2015) are structurally similar, while the estimated parameter values are substantially different. These parameters can clearly affect estimated bioavailability, as a drug with a higher clearance requires higher release rates in order to maintain the same plasma concentration.

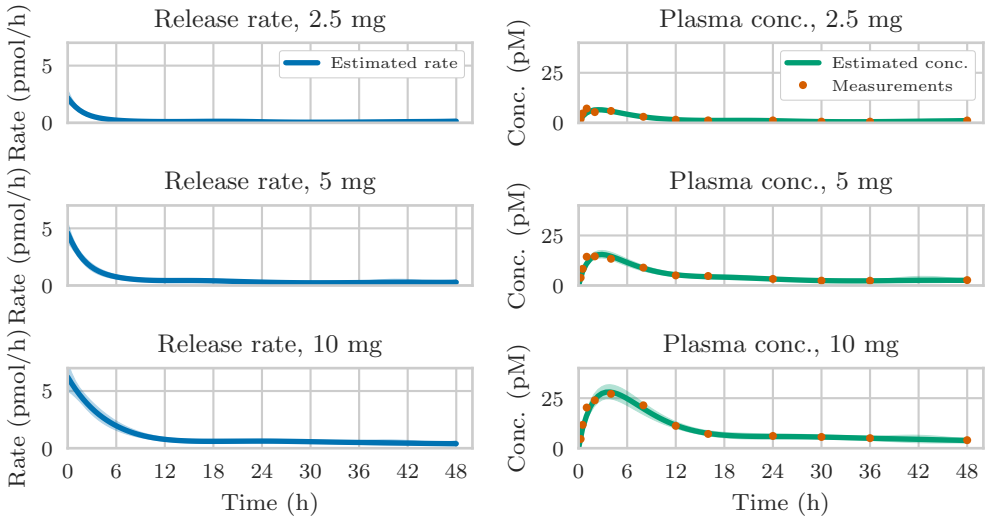

Figure 6.17: MCMC estimation results for the exenatide short-term data using Karhunen-Loève basis functions and a prior based on the second derivative in the log domain. In all cases, SMMALA was used as the sampling method. The 7 mg dose is not shown, as no sampling method was successful for this dose with the given prior and parameterisation.

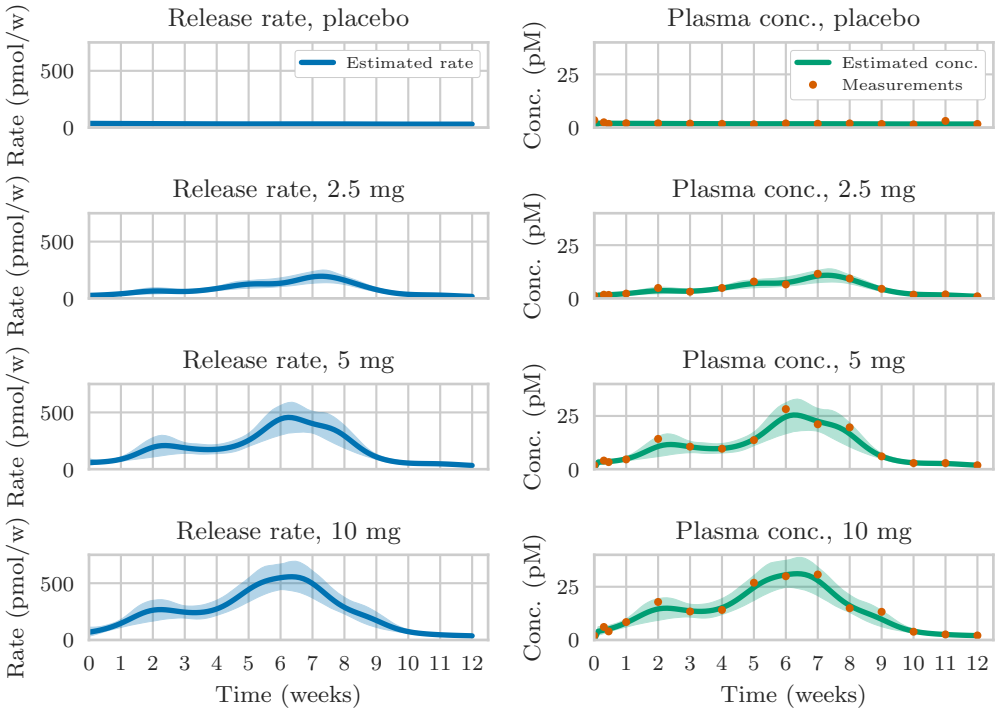

Figure 6.18: MCMC estimation results for the exenatide long-term data using Karhunen-Loève basis functions and a prior based on the second derivative in the log domain. In all cases, SMMALA was used as the sampling method. The 7 mg dose is not shown, as no sampling method was successful for this dose with the given prior and parameterisation.

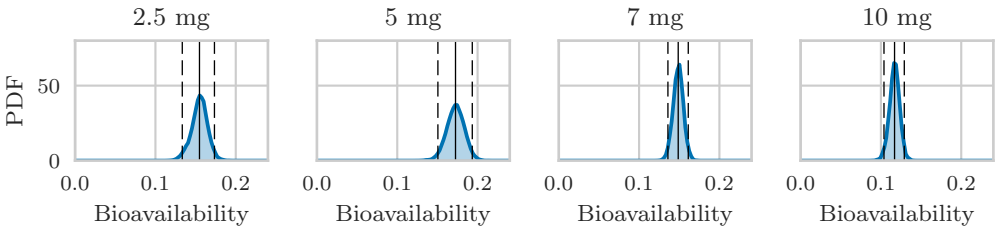

Figure 6.19: Kernel-density estimates for the bioavailability of the long-term release profiles. The calculations assume a body weight of 90 kg, according to the data in Fineman et al. (2011). In the displayed plots, estimation was performed using the same settings as in Fig. 6.18. The solid lines denote means, and the dashed lines show the 2.5th and 97.5th percentiles.

6.6 Summary

In this case study, a nonlinear three-compartment PK model, which is similar to the model in Chapter 4, has been studied. A major difference between these case studies is that here, the input function is more complicated, exhibiting multiple peaks. Additionally, here the study is conducted on various time scales, with the PK model being essentially in steady state on longer time scales. It can be noted that it would be possible to use an algebraic, steady-state approximation in order to make estimation more computationally efficient, although this was not tested in this work.

For the MCMC analysis, the usual Gaussian likelihood function was replaced by a Student's t-distribution with 4 degrees of freedom. This was done to make the method less sensitive to outliers in the data. Such outliers could otherwise drive the regularisation parameter to a very low value and cause problems for the estimation method, as shown in Fig. 6.10. However, it is important to note that such likelihood functions can also cause the method to ignore variations in the data that should not be ignored. Hence, the choice of likelihood function is a modelling decision that has to be made on a case-by-case basis.

When converting between piecewise constant and B-spline basis functions, it was found necessary to add a small regularisation term in order to combat numerical issues. From a mathematical perspective, this is similar to the regularisation that is used in the optimal-control methods, although the purpose is different.

For the optimal-control methods, the choice of prior did not appear to have any major effect on the estimation accuracy, as the RMSE values usually differed by at most a factor of two. The collocation methods were found to be faster than the shooting methods by two orders of magnitude. Additionally, the shooting methods were found to have major robustness issues on the long-term datasets, with many estimation procedures failing to converge. Since no optimisation method was successful in every case, one recommendation is to attempt to use collocation first, and then attempt to use shooting methods if this fails.

For the MCMC methods, the log-domain priors achieved considerably better accuracy than linear-domain priors for the short-term data. This may be because only the log-domain priors provide the nonnegativity constraints that are necessary in order to avoid unphysical solutions. For the long-term data, the choice of prior proved to have less impact on accuracy, possibly because the estimates were mostly positive even in the absence of nonnegativity constraints. For both the short-term and long-term data, Karhunen-Loève basis functions together with SMMALA sampling were superior in terms of computation time and robustness to any other combination of

parameterisation and sampling method. Since the Karhunen-Loève parameterisation also provided better estimation accuracy than the B-splines, a clear recommendation can be made to use this combination for problems of the type considered in this chapter.

Chapter 7

Conclusions and future work

7.1 Summary of the performed work

In this thesis, methods for input estimation (deconvolution) for drug-discovery applications have been investigated. The aim of input estimation is to infer the input function driving a dynamical system, based on measurements of the system. In the applications considered in this thesis, data are often sparse and irregularly sampled in comparison to other fields of engineering. Input estimation is a nonparametric technique which does not constrain the input function to have any particular form. A framework has been presented in which several previously suggested methods can be interpreted as a particular choice of prior, function parameterisation, and estimation algorithm, for point estimates as well as for full Bayesian inference. This framework provides a principled way to reason about and develop new input-estimation methods, and is one of the major contributions of this work.

Based on this framework, two major classes of inference algorithms have been selected: optimal control-based algorithms for point estimates and MCMC algorithms for full inference. These algorithms have been implemented using CasADi, a software for automatic differentiation and optimisation in dynamical systems. While these algorithms are well-known in fields such as optimal control and computational statistics, they have previously not been extensively applied to input-estimation problems in drug-discovery. Introducing these algorithms to drug-discovery applications has therefore been a major aim of this work. All of the selected combinations of prior, parameterisation, and estimation algorithm have been applied to four dynamical models from three case studies, using simulated as well as real data. In total, this amounts to 780 optimal-control and $1,676$ MCMC estimation procedures. To the best of my knowledge, this makes this thesis the most comprehensive evaluation of

input-estimation methods for drug discovery conducted so far. The methods have been evaluated using several criteria, as outlined in Chapter 1. The results from these evaluations provide valuable information about the strengths and weaknesses of the various methods, which will aid future development of new input-estimation methods. The results from the individual case studies are summarised at the end of each chapter. In this chapter, general conclusions, based on all the case studies, are discussed.

7.2 Parametric and nonparametric methods

In drug discovery, data analysis is usually performed using parametric models. Whether parametric or nonparametric techniques are appropriate depends on the aim of the analysis, as well as on the available system knowledge and data. The main strength of nonparametric methods is that they require fewer assumptions about the system than parametric methods. When the objective of the analysis is to model the available data, such as determining the bioavailability based on PK data, nonparametric methods are an interesting alternative to the empirical compartmental models that are otherwise typically used. However, another reason for building models is to obtain a better understanding of the system studied. In this case, detailed mechanistic models may be the more appropriate option, where the model structure and the model parameters can help in drawing conclusions about the system. Input estimation can still be useful as a first step in building such a model: being able to predict the time course of a part of the system that cannot be measured directly can help in formulating a hypothesis for the underlying mechanisms.

It should be noted that although nonparametric methods have the strength of requiring fewer assumptions than parametric methods, they still do require some assumptions to be made. These are reflected in the choice of prior. Although the assumptions may be weaker than those in parametric models, they may also be more difficult to justify. It is not evident whether penalisation of the first or the second derivative is more appropriate. As an example, the estimation results for the mouse body-weight model (Fig. 5.4) are strongly dependent on the choice of prior. In a practical setting, it may be prudent to perform estimation using various priors, in order to determine whether the estimation results are sensitive to the choice of prior. This is more likely to be the case when data are sparse. Using nonparametric methods on very sparse data essentially amounts to trying to draw conclusions based on neither good data nor strong assumptions. This is not likely to be successful.

In summary, input estimation is useful in situations where the focus is on

analysing the data at hand, rather than on obtaining a detailed understanding of the system under study, and when it is not possible or desirable to construct a parametric model. Nonparametric methods aim to let the data speak for themselves, which is only possible when the data are sufficiently densely sampled.

7.3 Comparison of input-estimation methods

Once a decision has been made to use input estimation, the next question that needs to be addressed is what estimation method to use. Below, the investigated methods are compared to each other, based on the evaluation criteria outlined in Chapter 1. The two major classes of algorithms are intended for different types of problems. While optimal-control algorithms are generally able to produce point estimates in a short amount of time, MCMC algorithms provide full Bayesian posteriors, at considerable computational cost.

Accuracy can only be measured on data for which the true input is known, and hence this assessment can only be performed on simulated data. Evaluation has mainly been performed using sparse and noisy simulated data, as these represent the most realistic case. The accuracy depends on the choice of prior in the case of optimal control-based methods, and on the choice of prior as well as on the function parameterisation in the case of MCMC methods. The choice of optimisation or sampling method does not affect the accuracy, as these choices do not affect the underlying probabilistic assumptions. Accuracy was measured using RMSE for the optimal control-methods and mean RMSE for the MCMC methods. The choice of prior clearly affected the resulting estimated functions qualitatively. In terms of RMSE, the maximum differences between priors were in most cases approximately a factor of 1.5–2. In the evaluation of MCMC methods on eflornithine and short-term exenatide data, log-domain priors systematically performed better than linear-domain priors. This may be partly explained by noting that linear-domain priors do not impose nonnegativity constraints, which is an important issue in regions where measurement values are close to 0 and sampling is sparse. Considering the choice of input function parameterisation, Karhunen-Loève basis functions generally achieved a higher accuracy than B-splines. The comparison of parameterisations is relatively incomplete, as in many cases the B-spline estimation methods failed and did not produce any results.

For the cases where only point estimates were desired, very good computational performance could be achieved. Across all case studies, collocation methods had running times of 0.1–0.5 s on a typical desktop computer, while shooting methods

typically required of the order of seconds or tens of seconds. Achieving high speed when computing full posteriors posed a considerably greater challenge. For each model, the highest-performing MCMC method achieved a median required running time of 60–90 s. However, no method was consistently successful across all case studies. While SMMALA with Karhunen-Loève basis functions achieved the highest performance for the eflornithine, human body weight, and exenatide datasets, this method required one order of magnitude more computing time for the mouse body weight dataset, where instead block RWMH with Karhunen-Loève functions proved to be superior. The fact that the sampling performance is problem-dependent to such a large extent has considerable practical implications. If it cannot be known beforehand which method will perform well, the modeller may be forced to test several methods, each of which could potentially be time-consuming. Block RWMH generally requires a large number of samples to be generated, but the required amount of computation per sample is low. In contrast, the MALA and SMMALA approaches generate higher-quality samples, at the expense of a longer computation time per sample. Single-component RWMH combines the worst qualities of the other methods.

These results support the conclusion that point-estimation algorithms have value, even though algorithms that can estimate the uncertainty provide more information. A modeller needs to have a certain amount of expertise in selecting, using, and diagnosing MCMC algorithms in order to apply them effectively. In contrast, optimal-control algorithms are easier to use for a casual user. If the estimation procedure fails, optimal-control algorithms can detect this automatically, and either attempt to use a different algorithm, or notify the user.

In terms of usability, most of the evaluated methods were chosen in order to require as little manual setting of algorithmic parameters as possible. This was largely done for practical reasons, in order to make it possible to run a large number of estimation procedures. The main exception to this principle is the L-curve method which was briefly described in Chapter 4. All methods require the number of basis functions to be manually specified. The collocation methods additionally require a choice of type and order of the interpolating polynomials.

All methods are based on similar statistical principles. The statistical soundness of the methods depends on to what extent the chosen prior and likelihood are appropriate to the problem at hand. Assessing the suitability of a prior is a difficult problem, as discussed in Section 7.2. Discussions on the justification of priors can be found in the case study chapters. In all cases, the underlying assumptions of the methods are clearly and explicitly stated.

For the eflornithine and the body-weight case studies, the optimal-control

methods achieved success rates of around 95%, irrespective of the choice of optimisation method. In the exenatide case study, robustness issues were found. In particular, for long-term data, only collocation methods performed relatively robustly, with shooting methods having failure rates of around 50%. In general, no method was universally successful. The robustness performance of the MCMC methods closely followed the computational speed performance, as a major cause of failed estimation procedures was failure to generate a sufficiently large number of samples in the allocated time. Hence, block RWMH with Karhunen-Loève basis functions achieved the highest success rate for the mouse body-weight model, while SMMALA with Karhunen-Loève basis functions achieved the highest success rate for all other models. For the eflornithine and exenatide datasets, the most successful method achieved success rates of 90–100%. For the body-weight case study, the highest performing methods achieved approximately 80% for the mouse and 60% for the human model.

In terms of usefulness, it is evident from the case studies that the performance of the methods can be highly problem-dependent. This demands a certain amount of experimentation on the part of the modeller when a new dataset is being analysed.

In summary, when point estimates are required, collocation methods were considerably faster than the alternatives, and in certain cases have also been shown to be more robust. When full posteriors are required, no single MCMC sampling method performed adequately in every investigated case. However, SMMALA achieved higher computational speed and robustness than the other methods in the majority of cases. Karhunen-Loève basis functions have been shown to perform consistently better than B-splines, both in terms of estimation accuracy, computational speed, and robustness. The choice of prior did not, in general, have a major impact on the measured accuracy. However, log-domain priors are preferable in cases where it is important to enforce nonnegativity constraints.

7.4 Summary of results

In the previous section, the investigated methods were compared to each other. Here, the highest-performing methods will be discussed in relation to the criteria that were presented in Chapter 1.

Ability to handle nonlinear models. In this work, only methods which are able to handle nonlinear models have been investigated.

Ability to provide uncertainty estimates. The MCMC methods do provide such estimates. However, a decision was made to also investigate optimal-control

methods, which lack this capability, as these methods could be shown to have better performance in other regards, such as computational speed and ease of use.

Good performance. This is discussed in further detail below.

The performance criteria for the input-estimation methods were satisfied to the following extent:

Accuracy. The accuracy of the methods is highly dependent on the sparsity of the measurements, as well as on the characteristics of the input function. For smooth functions which are consistent with the chosen prior, such as the short-term exenatide dataset in Fig. 6.15, a very high accuracy can be achieved. On the other hand, input functions that make sharp transitions immediately following a period of sparse data, as shown in Fig. 5.9, cannot be expected to be accurately recovered.

Computational speed. The achieved computational speed for the optimal-control methods is clearly good enough for practical use. For the MCMC methods, whether this is the case depends on the intended use. The speed is clearly adequate if estimation only needs to be performed once, but is lower than ideal for exploratory data analysis, where the modeller is working with the data interactively. For interactive use, it would be preferable to see the results in near-real time. From a user's perspective, having to wait a full minute in order to fit a single function may seem excessive.

Usability. Apart from the L-curve methods, all of the investigated methods require a relatively small amount of manual settings. However, all methods require the user to select the number of basis functions. It has not been investigated to what extent the methods are sensitive to this number.

Statistical soundness. In all of the investigated methods, the assumptions about the input function are encoded in the choice of prior and likelihood. Hence, the assumptions are clearly stated. The main difficulty in terms of statistical soundness is in justifying the choice of prior, as discussed in Section 7.2.

Robustness. The success rate of the best-performing methods reached approximately 90–95% in several cases. However, robustness was not universally satisfactory. The most successful MCMC method for the human body-weight dataset (SMMALA with Karhunen-Loève basis functions) achieved a success rate of approximately 60%.

Usefulness. The optimal-control methods appear to be well suited for all of the investigated datasets. For the MCMC methods, it proved to be challenging to find methods which are universally applicable. The most successful method overall, SMMALA with Karhunen-Loève basis functions, performed satisfactorily on 3 out of 4 datasets.

7.5 Suggestions for future work

Future work could address some of the shortcomings of the methods presented here. In order to improve accuracy and statistical soundness, it could be worthwhile to investigate whether other priors would be more suitable for these applications. In particular, the theory of Gaussian processes provides a rich selection of priors which could be evaluated. It may also be possible to incorporate several priors into the model, and let the estimation algorithm automatically select between them. As an example, the prior could involve one term penalising the first derivative and one term penalising the second derivative, each with its own regularisation parameter. If the data force the input function to attain a form which is improbable under one prior, its regularisation parameter will be forced to be small, as discussed in Section 6.4.2. Hence, the regularisation term which fits the data better would in practice dominate. Another possible method is to let the choice of regularisation term be a parameter which is sampled together with the other parameters in the Markov chain. This would be a form of Bayesian model averaging.

Improving computational speed for fully Bayesian methods would be useful. Since MCMC methods are not known to be fast, it could be of interest to investigate techniques based on analytical approximations, such as variational methods. When only a point estimate is desired, the collocation methods investigated here are probably already fast enough for any practical purposes.

Usability could be improved by finding methods to automatically set algorithm parameters such as the collocation method settings and the number of basis functions. In the latter case, it may be possible to allow the number of basis functions to be estimated as part of the input-estimation procedure. This could be achieved by treating this number as a parameter in the probabilistic model, similar to how the regularisation parameter is treated. At each stage of the Markov chain, the number of basis functions would be sampled. Since the number of parameters to be estimated is variable, it would be necessary to use transdimensional sampling methods, such as the methods that were briefly mentioned in Section 2.4.2.

In order to improve robustness, it is necessary to understand why methods fail.

Convergence problems in optimal-control methods may be the result of numerical issues. Finding automated ways to reparameterise the models could help here. For the MCMC methods, ensuring robustness would involve developing sampling methods which can consistently generate sets of samples that characterise the true posterior well in typical drug-discovery applications. Finding such methods would also improve usability, in that the user would not be required to manually select sampling methods. Another way to increase robustness is to develop methods for structural identifiability in input-estimation problems, which could help ensure that the problem has a well-defined solution.

It is also desirable to develop more user-friendly software for input estimation. Currently, general-purpose software for MCMC sampling does not include support for manifold sampling algorithms, which where shown in this work to have major advantages over more traditional sampling algorithms. This lack of support makes these methods inaccessible to non-experts. Productivity could also be increased by developing software which allows the user to specify the input-estimation problem at a high level of abstraction, reducing the need to manually specify details about how estimation is performed.

This thesis presented an exploratory study, which involved multiple comparisons between models, datasets, and estimation methods. In order to ensure that the results here hold in general, it would be necessary to design studies which aim to answer more specific questions about method performance, and apply formal statistical testing to the results.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu and Xiaoqiang Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2016. arXiv: `1603.04467 [cs.DC]`.

[2] Jan Albersmeyer and Moritz Diehl. 'The lifted Newton method and its application in optimization'. In: *SIAM Journal on Optimization* 20.3 (2010), pp. 1655–1684.

[3] David B Allison, Kishore M Gadde, William Timothy Garvey, Craig A Peterson, Michael L Schwiers, Thomas Najarian, Peter Y Tam, Barbara Troupin and Wesley W Day. 'Controlled-release phentermine/topiramate in severely obese adults: a randomized controlled trial (EQUIP)'. In: *Obesity* 20.2 (2012), pp. 330–342.

[4] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of information geometry*. Vol. 191. Translations of Mathematical Monographs. American Mathematical Society, 2000.

[5] Joel Andersson. 'A General-Purpose Software Framework for Dynamic Optimization'. PhD thesis. Arenberg Doctoral School, KU Leuven, 2013.

[6] Joel Andersson, Joris Gillis and Moritz Diehl. *User Documentation for CasADi v3.0.0*. 2016.

[7] Christophe Andrieu, Arnaud Doucet and Roman Holenstein. 'Particle markov chain monte carlo methods'. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72.3 (2010), pp. 269–342.

[8] Mosab Arafat. 'Approaches to achieve an oral controlled release drug delivery system using polymers: a recent review'. In: *International Journal of Pharmacy and Pharmaceutical Sciences* 7.7 (2015), pp. 16–21.

[9] Davis Yohanes Arifin, Lai Yeng Lee and Chi-Hwa Wang. 'Mathematical modeling and simulation of drug release from microspheres: Implications to drug delivery systems'. In: *Advanced drug delivery reviews* 58.12 (2006), pp. 1274–1325.

[10] L J Aronne, S Tonstad, M Moreno, I Gantz, N Erondu, S Suryawanshi, C Molony, S Sieberts, J Nayee, A G Meehan, D Shapiro, SB Heymsfield, KD Kaufman and JM Amatruda. 'A clinical trial assessing the safety and efficacy of taranabant, a CB1R inverse agonist, in obese and overweight patients: a high-dose study'. In: *International Journal of Obesity* 34.5 (2010), pp. 919–935.

[11] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon and Tim Clapp. 'A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking'. In: *Signal Processing, IEEE Transactions on* 50.2 (2002), pp. 174–188.

[12] Gregory L Austin, Lorraine G Ogden and James O Hill. 'Trends in carbohydrate, fat, and protein intakes and association with energy intake in normal-weight, overweight, and obese individuals: 1971–2006'. In: *The American journal of clinical nutrition* 93.4 (2011), pp. 836–843.

[13] Manica Balasegaram, Steve Harris, Francesco Checchi, Sara Ghorashian, Catherine Hamel and Unni Karunakara. 'Melarsoprol versus eflornithine for treating late-stage Gambian trypanosomiasis in the Republic of the Congo'. In: *Bulletin of the World Health Organization* 84.10 (2006), pp. 783–791.

[14] Kesara Na-Bangchang, F Doua, Julraht Konsil, Warunee Hanpitakpong, Benjamas Kamanikom and F Kuzoe. 'The pharmacokinetics of eflornithine (α-difluoromethylornithine) in patients with late-stage T.b. gambiense sleeping sickness'. In: *European journal of clinical pharmacology* 60.4 (2004), pp. 269–278.

[15] David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[16] Roscoe Bartlett. *A derivation of forward and adjoint sensitivities for ODEs and DAEs*. Tech. rep. SAND2007-6699. Sandia National Laboratories, 2008.

[17] M Jésus Bayarri and James O Berger. 'The interplay of Bayesian and frequentist analysis'. In: *Statistical Science* 19.1 (2004), pp. 58–80.

[18] Matthew James Beal. 'Variational algorithms for approximate Bayesian inference'. PhD thesis. University of London, 2003.

[19] Bradley M Bell and Gianluigi Pillonetto. 'Estimating parameters and stochastic functions of one variable using nonlinear measurement models'. In: *Inverse Problems* 20.3 (2004), pp. 627–646.

[20] John T Betts. *Practical methods for optimal control and estimation using nonlinear programming*. 2nd ed. Advances in Design and Control. Siam, 2010.

[21] Lorenz T Biegler. *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. MOS-SIAM Series on Optimization. SIAM, 2010.

[22] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[23] Carl de Boor. *B (asic)-Spline Basics*. Tech. rep. DTIC Document, 1986.

[24] William Munger Boothby. *An introduction to differentiable manifolds and Riemannian geometry*. 2nd ed. Elsevier, 2003.

[25] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[26] Steve Brooks, Andrew Gelman, Galin Jones and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. CRC Press, 2011.

[27] Reto Brun, Johannes Blum, Francois Chappuis and Christian Burri. 'Human african trypanosomiasis'. In: *The Lancet* 375.9709 (2010), pp. 148–159.

[28] Arthur Earl Bryson and Yu-Chi Ho. *Applied optimal control: optimization, estimation and control*. New York: Taylor & Francis Group, 1975.

[29] John B Buse, Daniel J Drucker, Kristin L Taylor, Terri Kim, Brandon Walsh, Hao Hu, Ken Wilhelm, Michael Trautmann, Larry Z Shen and Lisa E Porter. 'DURATION-1: exenatide once weekly produces sustained glycemic control and weight loss over 52 weeks'. In: *Diabetes Care* 33.6 (2010), pp. 1255–1261.

[30] John B Buse, Robert R Henry, Jenny Han, Dennis D Kim, Mark S Fineman and Alain D Baron. 'Effects of exenatide (exendin-4) on glycemic control over 30 weeks in sulfonylurea-treated patients with type 2 diabetes'. In: *Diabetes care* 27.11 (2004), pp. 2628–2635.

[31] John B Buse, Michael Nauck, Thomas Forst, Wayne H H Sheu, Sylvia K Shenouda, Cory R Heilmann, Byron J Hoogwerf, Aijun Gao, Marilyn K Boardman, Mark Fineman, Lisa Porter and Guntram Schernthaner. 'Exenatide once weekly versus liraglutide once daily in patients with type 2 diabetes (DURATION-6): a randomised, open-label study'. In: *The Lancet* 381.9861 (2013), pp. 117–124.

[32] John C Butcher. *Numerical methods for ordinary differential equations*. 2nd ed. John Wiley & Sons, 2008.

[33] Ben Calderhead. 'Differential geometric MCMC methods and applications'. PhD thesis. University of Glasgow, 2011.

[34] Yang Cao, Shengtai Li, Linda Petzold and Radu Serban. 'Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution'. In: *SIAM Journal on Scientific Computing* 24.3 (2003), pp. 1076–1089.

[35] Olivier Cappé, Simon J Godsill and Eric Moulines. 'An overview of existing methods and recent advances in sequential Monte Carlo'. In: *Proceedings of the IEEE* 95.5 (2007), pp. 899–924.

[36] J-M Cardot and B M Davit. 'In vitro–in vivo correlations: tricks and traps'. In: *The AAPS journal* 14.3 (2012), pp. 491–499.

[37] Bradley P Carlin and Siddhartha Chib. 'Bayesian model choice via Markov chain Monte Carlo methods'. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 57.3 (1995), pp. 473–484.

[38] Manfredo Perdigão do Carmo. *Riemannian geometry*. Birkhäuser, 1992.

[39] Bob Carpenter, Andrew Gelman, Matt Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Michael A Brubaker, Jiqiang Guo, Peter Li and Allen Riddell. 'Stan: A probabilistic programming language'. In: *Journal of Statistical Software* 76.1 (2017), pp. 1–32.

[40] George Casella and Roger L Berger. *Statistical inference*. 2nd ed. Duxbury, 2002.

[41] François Chappuis, Nitya Udayraj, Kai Stietenroth, Ann Meussen and Patrick A Bovier. 'Eflornithine is safer than melarsoprol for the treatment of second-stage Trypanosoma brucei gambiense human African trypanosomiasis'. In: *Clinical Infectious Diseases* 41.5 (2005), pp. 748–751.

[42] M K Charter and S F Gull. 'Maximum entropy and its application to the calculation of drug absorption rates.' In: *Journal of pharmacokinetics and biopharmaceutics* 15.6 (1987), pp. 645–55.

[43] Ting Chen, Donald E Mager and Leonid Kagan. 'Interspecies modeling and prediction of human exenatide pharmacokinetics'. In: *Pharmaceutical research* 30.3 (2013), pp. 751–760.

[44] Zhe Chen. 'Bayesian filtering: From Kalman filters to particle filters, and beyond'. In: *Statistics* 182.1 (2003), pp. 1–69.

[45] Siddhartha Chib and Edward Greenberg. 'Understanding the metropolis-hastings algorithm'. In: *The american statistician* 49.4 (1995), pp. 327–335.

[46] Chantal Csajka, David Drover and Davide Verotta. 'The use of a sum of inverse Gaussian functions to describe the absorption profile of drugs exhibiting complex absorption.' In: *Pharmaceutical research* 22.8 (2005), pp. 1227–1235.

[47] Jean Daunizeau, Karl J Friston and Stefan J Kiebel. 'Variational Bayesian identification and prediction of stochastic nonlinear dynamic causal models'. In: *Physica D: nonlinear phenomena* 238.21 (2009), pp. 2089–2118.

[48] Giuseppe De Nicolao, Giovanni Sparacino and Claudio Cobelli. 'Nonparametric input estimation in physiological systems: Problems, methods, and case studies'. In: *Automatica* 33.5 (1997), pp. 851–870.

[49] Ralph A DeFronzo, Robert E Ratner, Jenny Han, Dennis D Kim, Mark S Fineman and Alain D Baron. 'Effects of exenatide (exendin-4) on glycemic control and weight over 30 weeks in metformin-treated patients with type 2 diabetes'. In: *Diabetes care* 28.5 (2005), pp. 1092–1100.

[50] Pierre Del Moral, Arnaud Doucet and Ajay Jasra. 'Sequential monte carlo samplers'. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.3 (2006), pp. 411–436.

[51] Pierre Del Moral and Lawrence M Murray. 'Sequential Monte Carlo with highly informative observations'. In: *SIAM/ASA Journal on Uncertainty Quantification* 3.1 (2015), pp. 969–997.

[52] Jean-Pierre Després, Alain Golay and Lars Sjöström. 'Effects of rimonabant on metabolic risk factors in overweight patients with dyslipidemia'. In: *New England Journal of Medicine* 353.20 (2005), pp. 2121–2134.

[53] Mary Beth DeYoung, Leigh MacConell, Viren Sarin, Michael Trautmann and Paul Herbert. 'Encapsulation of exenatide in poly-(D, L-lactide-co-glycolide) microspheres produced an investigational long-acting once-weekly formulation for type 2 diabetes'. In: *Diabetes technology & therapeutics* 13.11 (2011), pp. 1145–1154.

[54] Arnaud Doucet, Mark Briers and Stéphane Sénécal. 'Efficient block sampling strategies for sequential Monte Carlo methods'. In: *Journal of Computational and Graphical Statistics* 15.3 (2006), pp. 693–711.

[55] Arnaud Doucet and Adam M Johansen. 'A tutorial on particle filtering and smoothing: Fifteen years later'. In: *The Oxford Handbook of Nonlinear Filtering*. Ed. by Dan Crisan and Boris Rozovskii. Oxford University Press, 2009, pp. 656–704.

[56] Simon Duane, Anthony D Kennedy, Brian J Pendleton and Duncan Roweth. 'Hybrid monte carlo'. In: *Physics letters B* 195.2 (1987), pp. 216–222.

[57] Gilles Eperon, Manica Balasegaram, Julien Potet, Charles Mowbray, Olaf Valverde and François Chappuis. 'Treatment options for second-stage gambiense human African trypanosomiasis'. In: *Expert review of anti-infective therapy* 12.11 (2014), pp. 1407–1417.

[58] European Medicines Agency. *Assessment Report for Bydureon*. report EMEA/H/C/002020. 2011.

[59] Alan H Fairlamb. 'Chemotherapy of human African trypanosomiasis: current and future prospects'. In: *Trends in parasitology* 19.11 (2003), pp. 488–494.

[60] Mark Fineman, Shawn Flanagan, Kristin Taylor, Maria Aisporna, Larry Z Shen, Kenneth F Mace, Brandon Walsh, Michaela Diamant, Brenda Cirincione, Prajakti Kothare, Wen-I Li and Leigh MacConell. 'Pharmacokinetics and pharmacodynamics of exenatide extended-release after single and multiple dosing'. In: *Clinical pharmacokinetics* 50.1 (2011), pp. 65–74.

[61] Brian P Flannery, William H Press, Saul A Teukolsky and William Vetterling. *Numerical recipes in C*. Press Syndicate of the University of Cambridge, 1992.

[62] Catherine Forbes, Merran Evans, Nicholas Hastings and Brian Peacock. *Statistical Distributions*. 4th ed. Wiley, 2010.

[63] Daniel Foreman-Mackey, David W Hogg, Dustin Lang and Jonathan Goodman. 'emcee: The MCMC hammer'. In: *Publications of the Astronomical Society of the Pacific* 125.925 (2013), pp. 306–312.

[64] Jerome Friedman, Trevor Hastie and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. Springer series in statistics Springer, Berlin, 2001.

[65] Nial Friel and Anthony N Pettitt. 'Marginal likelihood estimation via power posteriors'. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70.3 (2008), pp. 589–607.

[66] Johan Gabrielsson and Daniel Weiner. *Pharmacokinetic and Pharmacodynamic Data Analysis: Concepts and Applications*. 4th ed. Swedish Pharmaceutical Press, 2007.

[67] Wei Gao and William J Jusko. 'Target-mediated pharmacokinetic and pharmacodynamic model of exendin-4 in rats, monkeys, and humans'. In: *Drug Metabolism and Disposition* 40.5 (2012), pp. 990–997.

[68] Andrew Gelman. 'Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper)'. In: *Bayesian analysis* 1.3 (2006), pp. 515–534.

[69] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari and Donald B. Rubin. *Bayesian Data Analysis*. 3rd ed. CRC Press, 2013.

[70] Stuart Geman and Donald Geman. 'Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* PAMI-6.6 (1984), pp. 721–741.

[71] Peter Gennemark, Rasmus Jansson-Löfmark, Gina Hyberg, Maria Wigstrand, Dorota Kakol-Palm, Pernilla Håkansson, Daniel Hovdal, Peter Brodin, Maria Fritsch-Fredin, Madeleine Antonsson, Karolina Ploj and Johan Gabrielsson. 'A modeling approach for compounds affecting body composition'. In: *Journal of pharmacokinetics and pharmacodynamics* 40.6 (2013), pp. 651–667.

[72] Peter Gennemark, Magnus Trägårdh, Daniel Lindén, Karolina Ploj, Anders Johansson, Andrew Turnbull, Björn Carlsson and Madeleine Antonsson. 'Translational Modeling to Guide Study Design and Dose Choice in Obesity Exemplified by AZD1979, a Melanin-concentrating Hormone Receptor 1 Antagonist'. In: *CPT: Pharmacometrics & Systems Pharmacology* 6.7 (2017), pp. 458–468.

[73] Charles Geyer. 'Introduction to Markov Chain Monte Carlo'. In: *Handbook of Markov Chain Monte Carlo*. Ed. by Steve Brooks, Andrew Gelman, Galin L Jones and Meng Xiao-Li. Handbooks of Modern Statistical Methods. Chapman & Hall/CRC, 2011, pp. 3–48.

[74] Charles Geyer. 'Practical markov chain monte carlo'. In: *Statistical Science* 7.4 (1992), pp. 473–483.

[75] Walter R Gilks, Sylvia Richardson and David J Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, 1996.

[76] Mark Girolami and Ben Calderhead. 'Riemann manifold langevin and hamiltonian monte carlo methods'. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.2 (2011), pp. 123–214.

[77] Britta Göbel, Arjun Sanghvi and Kevin D Hall. 'Quantifying energy intake changes during obesity pharmacotherapy'. In: *Obesity* 22.10 (2014), pp. 2105–2108.

[78] Simon Godsill, Arnaud Doucet and Mike West. 'Monte Carlo smoothing for nonlinear time series'. In: *Journal of the american statistical association* 99.465 (2004), pp. 156–168.

[79] Gene H Golub, Michael Heath and Grace Wahba. 'Generalized cross-validation as a method for choosing a good ridge parameter'. In: *Technometrics* 21.2 (1979), pp. 215–223.

[80] Jonathan Goodman and Jonathan Weare. 'Ensemble samplers with affine invariance'. In: *Communications in applied mathematics and computational science* 5.1 (2010), pp. 65–80.

[81] Neil J Gordon, David J Salmond and Adrian F M Smith. 'Novel approach to nonlinear/non-Gaussian Bayesian state estimation'. In: *IEE Proceedings F (Radar and Signal Processing)* 140.2 (1993), pp. 107–113.

[82] Peter J Green. 'Reversible jump Markov chain Monte Carlo computation and Bayesian model determination'. In: *Biometrika* 82.4 (1995), pp. 711–732.

[83] Peter J Green. 'Trans-dimensional markov chain monte carlo'. In: *Highly Structured Stochastic Systems*. Ed. by Peter J Green, Nils Lid Hjort and Sylvia Richardson. Oxford Statistical Science Series. Oxford University Press, 2003, pp. 179–198.

[84] Andreas Griewank. 'A mathematical view of automatic differentiation'. In: *Acta Numerica* 12 (2003), pp. 321–398.

[85] Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. Siam, 2008.

[86] Neha Gulati and Himanshu Gupta. 'Parenteral drug delivery: a review'. In: *Recent patents on drug delivery & formulation* 5.2 (2011), pp. 133–145.

[87] Juen Guo and Kevin D Hall. 'Estimating the continuous-time dynamics of energy and fat metabolism in mice'. In: *PLoS computational biology* 5.9 (2009), e1000511.

[88] Juen Guo and Kevin D Hall. 'Predicting changes of body weight, body fat, energy expenditure and metabolic fuel selection in C57BL/6 mice'. In: *PLoS One* 6.1 (2011), e15961.

[89] H Haario, E Saksman and J Tamminen. 'An adaptive Metropolis algorithm'. In: *Bernoulli* 7.2 (2001), pp. 223–242.

[90] Ernst Hairer and Gerhard Wanner. 'Stiff differential equations solved by Radau methods'. In: *Journal of Computational and Applied Mathematics* 111.1 (1999), pp. 93–111.

[91] Kevin D Hall, Gary Sacks, Dhruva Chandramohan, Carson C Chow, Y Claire Wang, Steven L Gortmaker and Boyd A Swinburn. 'Quantification of the effect of energy imbalance on bodyweight'. In: *The Lancet* 378.9793 (2011), pp. 826–837.

[92] Per Christian Hansen and Dianne Prost O'Leary. 'The use of the L-curve in the regularization of discrete ill-posed problems'. In: *SIAM Journal on Scientific Computing* 14.6 (1993), pp. 1487–1503.

[93] W Keith Hastings. 'Monte Carlo sampling methods using Markov chains and their applications'. In: *Biometrika* 57.1 (1970), pp. 97–109.

[94] John G Hattersley, Neil D Evans, C Hutchison, P Cockwell, G Mead, A R Bradwell and Michael J Chappell. 'Nonparametric prediction of free-light chain generation in multiple myeloma patients'. In: *17th International federation of automatic control world congress (IFAC), Seoul, Korea*. Ed. by Myung Jin Chung and Pradeep Misra. 2008, pp. 8091–8096.

[95] H Hauner, M Meier, G Wendland, T Kurscheid and K Lauterbach. 'Weight reduction by sibutramine in obese subjects in primary care medicine: the SAT Study'. In: *Experimental and clinical endocrinology & diabetes* 112.4 (2004), pp. 201–207.

[96] O Heby, S C Roberts and Buddy Ullman. 'Polyamine biosynthetic enzymes as drug targets in parasitic protozoa'. In: *Biochemical Society Transactions* 31.2 (2003), pp. 415–419.

[97] Alan C Hindmarsh, Peter N Brown, Keith E Grant, Steven L Lee, Radu Serban, Dan E Shumaker and Carol S Woodward. 'SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers'. In: *ACM Transactions on Mathematical Software (TOMS)* 31.3 (2005), pp. 363–396.

[98] Matthew D Hoffman and Andrew Gelman. 'The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo'. In: *Journal of Machine Learning Research* 15 (2014), pp. 1593–1623.

[99] Roman Hovorka, Michael J Chappell, Keith R Godfrey, Francis N Madden, Martin K Rouse and Paul A Soons. 'CODE: a deconvolution program implementing a regularization method of deconvolution constrained to non-negative values. Description and pilot evaluation.' In: *Biopharmaceutics & drug disposition* 19.1 (1998), pp. 39–53.

[100] A S Jackson, P R Stanforth, J Gagnon, T Rankinen, A S Leon, D C Rao, J S Skinner, C Bouchard and J H Wilmore. 'The effect of sex, age and race on estimating percentage body fat from body mass index: The Heritage Family Study.' In: *International journal of obesity* 26.6 (2002), pp. 789–796.

[101] Robert T Jacobs, Bakela Nare and Margaret A Phillips. 'State of the art in African trypanosome drug discovery'. In: *Current topics in medicinal chemistry* 11.10 (2011), pp. 1255–1274.

[102] John A Jacquez. *Compartmental Analysis in Biology and Medicine*. 2nd ed. University of Michigan Press, 1985.

[103] William Philip T James. 'The epidemiology of obesity: the size of the problem'. In: *Journal of internal medicine* 263.4 (2008), pp. 336–352.

[104] Edwin T Jaynes. *Probability theory: the logic of science*. Cambridge University Press, 2003.

[105] Andrew H Jazwinski. *Stochastic processes and filtering theory*. Dover Publications, 2007.

[106] Carl-Christer Johansson, Peter Gennemark, Per Artursson, Angela Äbelö, Michael Ashton and Rasmus Jansson-Löfmark. 'Population pharmacokinetic modeling and deconvolution of enantioselective absorption of eflornithine in the rat.' In: *Journal of pharmacokinetics and pharmacodynamics* 40.1 (2013), pp. 117–128.

[107] Kenneth A Johnson and Roger S Goody. 'The original Michaelis constant: translation of the 1913 Michaelis–Menten paper'. In: *Biochemistry* 50.39 (2011), pp. 8264–8269.

[108] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola and Lawrence K Saul. 'An introduction to variational methods for graphical models'. In: *Machine learning* 37.2 (1999), pp. 183–233.

[109] John Bagterp Jørgensen. 'Adjoint sensitivity results for predictive control, state-and parameter-estimation with nonlinear models'. In: *Control Conference (ECC), 2007 European*. IEEE. 2007, pp. 3649–3656.

[110] SJ Julier and JK Uhlmann. 'Unscented filtering and nonlinear estimation'. In: *Proceedings of the IEEE* 92.3 (2004), pp. 401–422.

[111] Rudolf E Kalman. 'A new approach to linear filtering and prediction problems'. In: *Transactions of the ASME–Journal of Basic Engineering* 82.Series D (1960), pp. 35–45.

[112] Jun Goo Kang and Cheol-Young Park. 'Anti-obesity drugs: a review about their effects and safety'. In: *Diabetes & metabolism journal* 36.1 (2012), pp. 13–25.

[113] Robert E Kass, Bradley P Carlin, Andrew Gelman and Radford M Neal. 'Markov chain Monte Carlo in practice: a roundtable discussion'. In: *The American Statistician* 52.2 (1998), pp. 93–100.

[114] Yoshiaki Kawajir, Carl Laird, Stefan Vigerske and Andreas Wächter. *Introduction to IPOPT: a tutorial for downloading, installing, and using IPOPT*. 2016.

[115] Peter G E Kennedy. 'Clinical features, diagnosis, and treatment of human African trypanosomiasis (sleeping sickness)'. In: *The Lancet Neurology* 12.2 (2013), pp. 186–194.

[116] Lilesh Khalane, Atul Alkunte and Arunadevi Birajdar. *Sustained Release Drug Delivery System: A Concise Review*. 2016. URL: `http://www.pharmatutor.org/articles/sustained-release-drug-delivery-system-concise-review`.

[117] George S Kimeldorf and Grace Wahba. 'A correspondence between Bayesian estimation on stochastic processes and smoothing by splines'. In: *The Annals of Mathematical Statistics* 41.2 (1970), pp. 495–502.

[118] Fima C Klebaner. *Introduction to stochastic calculus with applications*. 3rd ed. Imperial College Press, 2012.

[119] Peter E Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, 1999.

[120] S. Kolås, B.a. Foss and T.S. Schei. 'Constrained nonlinear state estimation based on the UKF approach'. In: *Computers & Chemical Engineering* 33.8 (2009), pp. 1386–1401.

[121] S Kumanyika, R W Jeffery, A Morabia, C Ritenbaugh and V Antipatis. 'Public Health Approaches to the Prevention of Obesity (PHAPO) Working Group of the International Obesity Task Force (IOTF)'. In: *International Journal of Obesity* 26.3 (2002), pp. 425–436.

[122] Olivier Le Maître and Omar M Knio. *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics*. Springer, 2010.

[123] Christopher J Lelliott, Andrea Ahnmark, Therese Admyre, Ingela Ahlstedt, Lorraine Irving, Feenagh Keyes, Laurel Patterson, Michael B Mumphrey, Mikael Bjursell, Tracy Gorman, Mohammad Bohlooly-Y, Andrew Buchanan, Paula Harrison, Tristan Vaughan, Hans-Rudolf Berthoud and Daniel Lindén. 'Monoclonal Antibody Targeting of Fibroblast Growth Factor Receptor 1c Ameliorates Obesity and Glucose Intolerance via Central Mechanisms'. In: *PLoS ONE* 9.11 (2014). Ed. by Marc Claret, e112109.

[124] Bernard C Levy. *Principles of signal detection and parameter estimation*. Springer, 2008.

[125] Hanqing Li, Jiayin Xu and Xiaohong Fan. 'Target-mediated pharmacokinetic/pharmacodynamic model based meta-analysis and dosing regimen optimization of a long-acting release formulation of exenatide in patients with type 2 diabetes mellitus'. In: *Journal of pharmacological sciences* 127.2 (2015), pp. 170–180.

[126] Daniel Liberzon. *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press, 2012.

[127] Fredrik Lindsten. 'Particle filters and Markov chains for learning of dynamical systems'. PhD thesis. Linköping University, 2013.

[128] Fredrik Lindsten and Thomas B Schön. 'Backward Simulation Methods for Monte Carlo Statistical Inference.' In: *Foundations and Trends in Machine Learning* 6.1 (2013), pp. 1–143.

[129] J C K Loo and S Riegelman. 'New method for calculating the intrinsic absorption rate of drugs'. In: *Journal of pharmaceutical sciences* 57.6 (1968), pp. 918–928.

[130] Ying Lu, Sungwon Kim and Kinam Park. 'In vitro–in vivo correlation: Perspectives on model development'. In: *International journal of pharmaceutics* 418.1 (2011), pp. 142–148.

[131] David Lunn, David Spiegelhalter, Andrew Thomas and Nicky Best. 'The BUGS project: Evolution, critique and future directions'. In: *Statistics in medicine* 28.25 (2009), pp. 3049–3067.

[132] David J C MacKay. 'Bayesian interpolation'. In: *Neural Computation* 4.3 (1992), pp. 415–447.

[133] David J C MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

[134] Francis N Madden, Keith R Godfrey, Michael J Chappell, Roman Hovorka and Ronald A Bates. 'A comparison of six deconvolution techniques'. In: *Journal of pharmacokinetics and biopharmaceutics* 24.3 (1996), pp. 283–299.

[135] Paolo Magni, Riccardo Bellazzi and Giuseppe De Nicolao. 'Bayesian function learning using MCMC methods'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 20.12 (1998), pp. 1319–1331.

[136] Timothy Maly and Linda R Petzold. 'Numerical methods and software for sensitivity analysis of differential-algebraic systems'. In: *Applied Numerical Mathematics* 20.1 (1996), pp. 57–79.

[137] MATLAB. *R2016a*. Natick, Massachusetts: The MathWorks Inc., 2016.

[138] Makoto Matsumoto and Takuji Nishimura. 'Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator'. In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 8.1 (1998), pp. 3–30.

[139] James Mercer. 'Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations'. In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical character* 209 (1909), pp. 415–446.

[140] Rudolph Van Der Merwe. 'Sigma-point Kalman filters for probabilistic inference in dynamic state-space models'. PhD thesis. Oregon Health & Science University, 2004.

[141] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller and Edward Teller. 'Equation of state calculations by fast computing machines'. In: *The journal of chemical physics* 21.6 (1953), pp. 1087–1092.

[142] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. 2nd ed. Cambridge University Press, 2009.

[143] Mark D Mifflin, Sachiko T St Jeor, Lisa A Hill, Barbara J Scott, Sandra A Daugherty and Y O Koh. 'A new predictive equation for resting energy expenditure in healthy individuals.' In: *The American journal of clinical nutrition* 51.2 (1990), pp. 241–247.

[144] Raymond Miller, Wayne Ewy, Brian W Corrigan, Daniele Ouellet, David Hermann, Kenneth G Kowalski, Peter Lockwood, Jeffrey R Koup, Sean Donevan, Ayman El-Kattan, Cheryl SW Li, John L Werth, Douglas E Feltner and Richard L Lalonde. 'How modeling and simulation have enhanced decision making in new drug development'. In: *Journal of pharmacokinetics and pharmacodynamics* 32.2 (2005), pp. 185–197.

[145] Thomas P Minka. 'A family of algorithms for approximate Bayesian inference'. PhD thesis. Massachusetts Institute of Technology, 2001.

[146] Thomas P Minka. 'Expectation propagation for approximate Bayesian inference'. In: *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Ed. by John Breese and Daphne Koller. Morgan Kaufmann Publishers Inc., 2001, pp. 362–369.

[147] Samir Mitragotri, Paul A Burke and Robert Langer. 'Overcoming the challenges in administering biopharmaceuticals: formulation and delivery strategies'. In: *Nature reviews Drug discovery* 13.9 (2014), pp. 655–672.

[148] Radford M Neal. 'MCMC using Hamiltonian dynamics'. In: *Handbook of Markov Chain Monte Carlo*. Ed. by Steve Brooks, Andrew Gelman, Galin L Jones and Meng Xiao-Li. Vol. 2. Handbooks of Modern Statistical Methods. Chapman & Hall/CRC, 2011, pp. 113–162.

[149] Radford M Neal. 'Slice sampling'. In: *Annals of statistics* 31.3 (2003), pp. 705–767.

[150] Q U Ning, Natalia A Ignatenko, Phillip Yamauchi, David E Stringer, Corey Levenson, Patrick Shannon, Scott Perrin and Eugene W Gerner. 'Inhibition of human ornithine decarboxylase activity by enantiomers of difluoromethylornithine'. In: *Biochemical journal* 375.2 (2003), pp. 465–470.

[151] Jorge Nocedal and Stephen Wright. *Numerical optimization*. 2nd ed. Springer, 2006.

[152] Bernt Øksendal. *Stochastic differential equations: an introduction with applications*. 6th ed. Springer, 2003.

[153] Raj S Padwal and Sumit R Majumdar. 'Drug treatments for obesity: orlistat, sibutramine, and rimonabant'. In: *The Lancet* 369.9555 (2007), pp. 71–77.

[154] Anand Patil, David Huard and Christopher J Fonnesbeck. 'PyMC: Bayesian stochastic modelling in Python'. In: *Journal of statistical software* 35.4 (2010), pp. 1–81.

[155] Roger Penrose. 'A generalized inverse for matrices'. In: *Mathematical proceedings of the Cambridge philosophical society* 51.3 (1955), pp. 406–413.

[156] Gianluigi Pillonetto and Bradley M. Bell. 'Bayes and empirical Bayes semi-blind deconvolution using eigenfunctions of a prior covariance'. In: *Automatica* 43.10 (2007), pp. 1698–1712.

[157] Gianluigi Pillonetto, Giovanni Sparacino and Claudio Cobelli. 'Handling non-negativity in deconvolution of physiological signals: a nonlinear stochastic approach'. In: *Annals of biomedical engineering* 30.8 (2002), pp. 1077–1087.

[158] Michael K Pitt and Neil Shephard. 'Filtering via simulation: Auxiliary particle filters'. In: *Journal of the American statistical association* 94.446 (1999), pp. 590–599.

[159] Martyn Plummer. 'JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling'. In: *Proceedings of the 3rd international workshop on distributed statistical computing*. Ed. by Kurt Hornik, Friedrich Leisch and Achim Zeileis. 2003.

[160] Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. CRC Press, 1987.

[161] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. 2016.

[162] Adrian E Raftery and Steven Lewis. 'How many iterations in the Gibbs sampler'. In: *Bayesian statistics* 4.2 (1992), pp. 763–773.

[163] Adrian E. Raftery and Steven M. Lewis. 'The Number of Iterations, Convergence Diagnostics and Generic Metropolis Algorithms'. In: *Practical Markov Chain Monte Carlo*. Ed. by Walter R Gilks, David J Spiegelhalter and Sylvia Richardson. Chapman and Hall, 1995.

[164] Louis B Rall and George F Corliss. 'An introduction to automatic differentiation'. In: *Computational Differentiation: Techniques, Applications, and Tools* (1996), pp. 1–17.

[165] Anil V Rao. 'A survey of numerical methods for optimal control'. In: *Advances in the Astronautical Sciences*. Ed. by Anil V Rao, T Alan Lovell, F Kenneth Chan and L Alberto Cangahuala. Vol. 135. 1. 2009, pp. 497–528.

[166] Carl Edward Rasmussen and Christopher K I Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[167] M P Ratnaparkhi and P Gupta Jyoti. 'Sustained release oral drug delivery system-an overview'. In: *International Journal of Pharma Research & Review* 2.3 (2013), pp. 11–21.

[168] H E Rauch, C T Striebel and F Tung. 'Maximum likelihood estimates of linear dynamic systems'. In: *AIAA journal* 3.8 (1965), pp. 1445–1450.

[169] Yun-Seok Rhee, Chun-Woong Park, Patrick P DeLuca and Heidi M Mansour. 'Sustained-Release Injectable Drug Delivery. A review on the current status of long-acting injectables, including commercially marketed products'. In: *Pharmaceutical Technology* 2010 Supplement.6 (2010).

[170] Christian Robert and George Casella. *Monte Carlo statistical methods*. 2nd ed. Springer, 2004.

[171] Gareth O Roberts. 'Markov chain concepts related to sampling algorithms'. In: *Markov chain Monte Carlo in practice*. Ed. by Walter R Gilks, Sylvia Richardson and David J Spiegelhalter. Chapman & Hall, 1996, pp. 45–54.

[172] Gareth O Roberts and Jeffrey S Rosenthal. 'General state space Markov chains and MCMC algorithms'. In: *Probability Surveys* 1 (2004), pp. 20–71.

[173] Gareth O Roberts and Jeffrey S Rosenthal. 'Harris recurrence of Metropolis-within-Gibbs and trans-dimensional Markov chains'. In: *The Annals of Applied Probability* 16.4 (2006), pp. 2123–2139.

[174] Gareth O Roberts and Jeffrey S Rosenthal. 'Optimal scaling for various Metropolis-Hastings algorithms'. In: *Statistical science* 16.4 (2001), pp. 351–367.

[175] Gareth O Roberts and Osnat Stramer. 'Langevin diffusions and Metropolis-Hastings algorithms'. In: *Methodology and computing in applied probability* 4.4 (2002), pp. 337–357.

[176] Gareth O Roberts and Richard L Tweedie. 'Exponential convergence of Langevin distributions and their discrete approximations'. In: *Bernoulli* 2.4 (1996), pp. 341–363.

[177] Jeffrey Seth Rosenthal. *A first look at rigorous probability theory*. 2nd ed. World Scientific, 2006.

[178] Sheldon M Ross. *Introduction to probability and statistics for engineers and scientists*. 5th ed. Academic Press, 2014.

[179] John Salvatier, Thomas V Wiecki and Christopher Fonnesbeck. 'Probabilistic programming in Python using PyMC3'. In: *PeerJ Computer Science* 2 (2016), e55.

[180] Simo Särkkä. *Bayesian filtering and smoothing*. Cambridge University Press, 2013.

[181] Simo Särkkä. 'Unscented Rauch-Tung-Striebel Smoother'. In: *IEEE Transactions on Automatic Control* 53.3 (2008), pp. 845–849.

[182] Larry Schumaker. *Spline functions: basic theory*. 3rd ed. Cambridge University Press, 2007.

[183] Steven P Schwendeman, Ronak B Shah, Brittany A Bailey and Anna S Schwendeman. 'Injectable controlled release depots for large molecules'. In: *Journal of Controlled Release* 190 (2014), pp. 240–253.

[184] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. Vol. 551. Wiley Series in Probability and Statistics. John Wiley & Sons, 2015.

[185] Radu Serban and Alan C Hindmarsh. 'CVODES: the sensitivity-enabled ODE solver in SUNDIALS'. In: *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers. 2005, pp. 257–269.

[186] Jie Shen and Diane J Burgess. 'In vitro–in vivo correlation for complex non-oral drug products: Where do we stand?' In: *Journal of Controlled Release* 219 (2015), pp. 644–651.

[187] J Siepmann and A Göpferich. 'Mathematical modeling of bioerodible, polymeric drug delivery systems'. In: *Advanced drug delivery reviews* 48.2 (2001), pp. 229–247.

[188] J Siepmann and N A Peppas. 'Modeling of drug release from delivery systems based on hydroxypropyl methylcellulose (HPMC)'. In: *Advanced drug delivery reviews* 48.2 (2001), pp. 139–157.

[189] Analiza M Silva, Jack Wang, Richard N Pierson Jr, ZiMian Wang, John Spivack, David B Allison, Steven B Heymsfield, Luis B Sardinha and Stanley Heshka. 'Extracellular water across the adult lifespan: reference values for adults'. In: *Physiological measurement* 28.5 (2007), pp. 489–502.

[190] D. Simon. 'Kalman filtering with state constraints: a survey of linear and nonlinear algorithms'. In: *IET Control Theory & Applications* 4.8 (2010), pp. 1303–1318.

[191] John Skilling. 'The axioms of maximum entropy'. In: *Maximum-Entropy and Bayesian Methods in Science and Engineering*. Ed. by Gary J Erickson and C Ray Smith. Vol. 1. 1988, pp. 173–187.

[192] John Skilling and RK Bryan. 'Maximum entropy image reconstruction-general algorithm'. In: *Monthly Notices of the Royal Astronomical Society* 211.1 (1984), pp. 111–124.

[193] Ian G Smith and Michael A Goulder. 'Randomized placebo-controlled trial of long-term treatment with sibutramine in mild to moderate obesity'. In: *Journal of Family Practice* 50.6 (2001), pp. 505–512.

[194] S R Smith, N J Weissman, C M Anderson, M Sanchez, E Chuang, S Stubbe, H Bays and W R Shanahan. 'Behavioral Modification and Lorcaserin for Overweight and Obesity Management (BLOOM) Study Group. Multicenter, placebo-controlled trial of lorcaserin for weight management'. In: *New England Journal of Medicine* 363.3 (2010), pp. 245–256.

[195] Giovanni Sparacino and Claudio Cobelli. 'A stochastic deconvolution method to reconstruct insulin secretion rate after a glucose stimulus'. In: *Biomedical Engineering, IEEE Transactions on* 43.5 (1996), pp. 512–529.

[196] Giovanni Sparacino, Gianluigi Pillonetto, Giuseppe De Nicolao and Claudio Cobelli. 'Deconvolution for Physiological Signal Analysis'. In: *Advanced Methods of Biomedical Signal Processing*. Ed. by Sergio Cerutti and Carlo Marchesi. Wiley-IEEE Press, 2011, pp. 169–198.

[197] David Spiegelhalter, Andrew Thomas, Nicky Best and David Lunn. *WinBUGS user manual*. Version 1.4. MRC Biostatistics Unit. 1999.

[198] Haijun D Sun, Maria Malabunga, James R Tonra, Roberto DiRenzo, Francine E Carrick, Huiyuan Zheng, Hans-Rudolf Berthoud, Owen P McGuinness, Juqun Shen, Peter Bohlen, Rudolph L Leibel and Paul Kussie. 'Monoclonal antibody antagonists of hypothalamic FGFR1 cause potent but reversible hypophagia and weight loss in rodents and monkeys'. In: *American Journal of Physiology-Endocrinology and Metabolism* 292.3 (2007), E964–E976.

[199] F Xavier Pi-Sunyer, Louis J Aronne, Hassan M Heshmati, Jeanne Devin and Julio Rosenstock. 'Effect of rimonabant, a cannabinoid-1 receptor blocker, on weight and cardiometabolic risk factors in overweight or obese patients: RIO-North America: a randomized controlled trial'. In: *Jama* 295.7 (2006), pp. 761–775.

[200] Theano Development Team. *Theano: A Python framework for fast computation of mathematical expressions*. 2016. arXiv: 1605.0 [cs.SC].

[201] Luke Tierney. 'Introduction to general state-space Markov chain theory'. In: *Markov chain Monte Carlo in practice*. Ed. by Walter R Gilks, Sylvia Richardson and David J Spiegelhalter. Chapman & Hall, 1996, pp. 59–74.

[202] Luke Tierney. 'Markov chains for exploring posterior distributions'. In: *the Annals of Statistics* 22.4 (1994), pp. 1701–1762.

[203] Jarl S Torgerson, Jonathan Hauptman, Mark N Boldrin and Lars Sjöström. 'XENical in the Prevention of Diabetes in Obese Subjects (XENDOS) Study A randomized study of orlistat as an adjunct to lifestyle changes for the prevention of type 2 diabetes in obese patients'. In: *Diabetes care* 27.1 (2004), pp. 155–161.

[204] Magnus Trägårdh, Michael J Chappell, Andrea Ahnmark, Daniel Lindén, Neil D Evans and Peter Gennemark. 'Input estimation for drug discovery using optimal control and Markov chain Monte Carlo approaches'. In: *Journal of pharmacokinetics and pharmacodynamics* 43.2 (2016), pp. 207–221.

[205] S Twomey. 'The application of numerical filtering to the solution of integral equations encountered in indirect sensing measurements'. In: *Journal of the Franklin Institute* 279.2 (1965), pp. 95–109.

[206] Sandor Vajda, Keith R Godfrey and Peter Valko. 'Numerical deconvolution using system identification methods'. In: *Journal of pharmacokinetics and biopharmaceutics* 16.1 (1988), pp. 85–107.

[207] Stefan Van Der Walt, S Chris Colbert and Gael Varoquaux. 'The NumPy array: a structure for efficient numerical computation'. In: *Computing in Science & Engineering* 13.2 (2011), pp. 22–30.

[208] Luc F Van Gaal, Aila M Rissanen, André J Scheen, Olivier Ziegler and Stephan Rössner. 'Effects of the cannabinoid-1 receptor blocker rimonabant on weight reduction and cardiovascular risk factors in overweight patients: 1-year experience from the RIO-Europe study'. In: *The Lancet* 365.9468 (2005), pp. 1389–1397.

[209] Luc Van Gaal, Xavier Pi-Sunyer, Jean-Pierre Després, Christine McCarthy and André Scheen. 'Efficacy and Safety of Rimonabant for Improvement of Multiple Cardiometabolic Risk Factors in Overweight/Obese patients Pooled 1-year data from the Rimonabant in Obesity (RIO) program'. In: *Diabetes care* 31.Supplement 2 (2008), S229–S240.

[210] Peter Veng-Pedersen and Nishit B Modi. 'An algorithm for constrained deconvolution based on reparameterization'. In: *Journal of pharmaceutical sciences* 81.2 (1992), pp. 175–180.

[211] Davide Verotta. 'Concepts, properties, and applications of linear systems to describe distribution, identify input, and control endogenous substances and drugs in biological systems'. In: *Critical Reviews^{TM} in Biomedical Engineering* 24.2-3 (1996), pp. 73–139.

[212] Ashlee N Ford Versypt, Daniel W Pack and Richard D Braatz. 'Mathematical modeling of drug delivery from autocatalytically degradable PLGA microspheres—a review'. In: *Journal of controlled release* 165.1 (2013), pp. 29–37.

[213] Sandra A G Visser, Malin Aurell, Rhys D O Jones, Virna J A Schuck, Ann-Charlotte Egnell, Sheila A Peters, Lena Brynne, James W T Yates, Rasmus Jansson-Löfmark, Beesan Tan, Marie Cooke, Simon T Barry, Andrew Hughes and Ulf Bredberg. 'Model-based drug discovery: implementation and impact'. In: *Drug discovery today* 18.15 (2013), pp. 764–775.

[214] Andreas Wächter. 'An interior point algorithm for large-scale nonlinear optimization with applications in process engineering'. PhD thesis. Carnegie Mellon University, 2002.

[215] Andreas Wächter. 'Short Tutorial: Getting Started With Ipopt in 90 Minutes'. In: *Combinatorial Scientific Computing*. Ed. by Uwe Naumann, Olaf Schenk, Horst D. Simon and Sivan Toledo. Dagstuhl Seminar Proceedings 09061. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009.

[216] Andreas Wächter and Lorenz T Biegler. 'On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming'. In: *Mathematical programming* 106.1 (2006), pp. 25–57.

[217] John G Wagner and Eino Nelson. 'Kinetic analysis of blood levels and urinary excretion in the absorptive phase after single doses of drug'. In: *Journal of pharmaceutical sciences* 53.11 (1964), pp. 1392–1403.

[218] Grace Wahba. *Spline models for observational data*. Siam, 1990.

[219] Martin J Wainwright and Michael I Jordan. 'Graphical models, exponential families, and variational inference'. In: *Foundations and Trends® in Machine Learning* 1.1-2 (2008), pp. 1–305.

[220] Limin Wang. 'Karhunen-Loeve expansions and their applications.' PhD thesis. London School of Economics and Political Science (United Kingdom), 2008.

[221] J Wilding, L Van Gaal, Aila Rissanen, F Vercruysse and M Fitchet. 'A randomized double-blind placebo-controlled study of the long-term efficacy and safety of topiramate in the treatment of obese subjects'. In: *International journal of obesity* 28.11 (2004), pp. 1399–1410.

[222] John Michael Winn. 'Variational message passing and its applications'. PhD thesis. University of Cambridge, 2004.

[223] World Health Organization. *Investing to overcome the global impact of neglected tropical diseases: third WHO report on neglected diseases 2015. Third WHO report on neglected tropical diseases*. World Health Organization, 2015.

[224] World Health Organization. *Obesity and overweight fact sheet*. 2016. URL: `http://www.who.int/mediacentre/factsheets/fs311/en/` (visited on 02/12/2016).

[225] Ai-Luen Wu, Ganesh Kolumam, Scott Stawicki, Yongmei Chen, Jun Li, Jose Zavala-Solorio, Khanhky Phamluong, Bo Feng, Li Li, Scot Marsters, Lance Kates, Nicholas van Bruggen, Maya Leabman, Anne Wong, David West, Howard Stern, Elizabeth Luis, Hok Seon Kim, Daniel Yansura, Andrew S Peterson, Filvaroff Ellen, Yan Wu and Junichiro Sonoda. 'Amelioration of type 2 diabetes by antibody-mediated activation of fibroblast growth factor receptor 1'. In: *Science translational medicine* 3.113 (2011), 113ra126.

[226] Wan-Wan Yang and Erik Pierstorff. 'Reservoir-based polymer drug delivery systems'. In: *Journal of laboratory automation* 17.1 (2012), pp. 50–58.

Appendix A

Data

A.1 Eflornithine data

Dose 20 mg/kg		**Dose 1500 mg/kg**	
Time (h)	Plasma conc. (μmol/mL)	Time (h)	Plasma conc. (μmol/mL)
0.52	0.0124	1.45	0.252
0.87	0.0176	2.73	0.314
1.13	0.0192	4.15	0.508
1.67	0.0157	5.57	0.774
2.18	0.00871	6.95	0.646
2.70	0.00614	8.52	0.348
3.15	0.00506	9.95	0.106
3.83	0.00374	11.85	0.0643
4.78	0.00150	14.08	0.0285
7.53	0.00182	20.8	0.0225
		24.02	0.0290

Table A.1: Eflornithine plasma-concentration data (Johansson et al. 2013).

A.2 Mouse body-weight data

Time (days)	Vehicle	R1c mAb opt1 (0.3 mg/kg)	R1c mAb opt1 (3 mg/kg)	R1c mAb opt1 (10 mg/kg)
-9	37.1	37.0	36.9	37.0
-1	37.7	38.1	37.4	38.0
0	38.3	38.6	37.8	38.4
1	38.3	37.9	36.9	37.5
2	37.9	37.2	35.9	36.2
3	37.7	36.6	35.0	35.3
4	37.5	36.4	34.2	34.2
5	37.7	35.9	33.4	33.6
6	37.8	35.9	32.5	32.5
7	37.7	36.3	32.0	31.9
8	37.9	36.3	31.5	31.1
9	37.9	36.8	31.5	30.6
10	37.9	36.6	31.4	30.1
12	38.0	36.8	32.3	29.2
14	38.2	36.4	32.8	28.5
16	39.0	37.5	33.3	28.5
19	39.6	37.9	34.7	29.5
21	39.6	38.6	35.2	30.5
23	40.5	39.2	36.5	31.7
26	41.1	39.9	36.8	33.1
28	41.5	39.9	37.3	33.7
30	42.5	40.9	39.0	35.6

Table A.2: Body weight in g for the mice treated with R1c mAb opt1 (Trägårdh et al. 2016).

Time (days)	R1c mAb opt2 (0.3 mg/kg)	R1c mAb opt2 (3 mg/kg)	R1c mAb opt2 (10 mg/kg)
-9	36.8	36.9	37.0
-1	37.4	36.8	36.8
0	37.6	36.9	37.7
1	37.5	36.3	36.9
2	36.9	35.8	35.8
3	36.7	35.1	34.8
4	36.8	34.7	34.0
5	36.6	34.4	33.4
6	36.4	34.4	32.4
7	36.5	34.5	31.9
8	36.7	34.5	31.2
9	37.3	34.9	30.8
10	37.3	34.9	30.5
12	37.6	35.4	30.3
14	38.0	36.0	30.5
16	38.8	36.7	31.3
19	38.7	37.5	32.3
21	39.2	37.9	32.7
23	39.8	38.8	34.1
26	39.9	38.8	35.3
28	39.8	38.9	35.5
30	40.6	40.2	37.2

Table A.3: Body weight in g for the mice treated with R1c mAb opt2 (Trägårdh et al. 2016).

A.3 Human body-weight data

Rimonabant, 20 mg Van Gaal et al. 2008		**Orlistat, 120 mg bds Torgerson et al. 2004**		**Rimonabant, 20 mg Pi-Sunyer et al. 2006**	
Time (days)	BW (kg)	Time (days)	BW (kg)	Time (days)	BW (kg)
0	101.0	0	110.4	0	103.0
28	99.3	84	103.7	14	102.3
56	97.9	168	100.8	28	101.4
112	95.7	364	99.8	84	98.7
140	94.9	448	100.6	112	97.9
168	94.1	532	100.6	140	97.1
196	93.7	616	101.1	168	96.2
224	93.2	728	101.6	196	95.7
252	92.8	812	102.5	224	95.3
280	92.8			252	94.8
308	92.7			280	94.8
336	92.6			308	94.6
364	92.4			336	94.6
				364	94.4
Sibutramine, 15 mg Smith et al. 2001		**Sibutramine, 15 mg Hauner et al. 2004**		**Taranabant, 4 mg Aronne et al. 2010**	
Time (days)	BW (kg)	Time (days)	BW (kg)	Time (days)	BW (kg)
0	86.8	0	99.5	0	99.2
30	83.8	28	95.7	14	97.7
60	82.3	56	93.0	28	96.5
90	81.3	84	91.2	56	94.9
120	80.8	126	89.9	84	93.7
150	80.5	168	89.1	112	92.8
180	80.2	210	88.8	140	92.1
210	80.2	252	88.9	168	91.6
240	80.3	294	88.8	196	91.2
270	80.2	336	89.2	224	91.0
300	80.2	378	89.8	252	90.6
330	80.5			280	90.6
360	80.5			308	90.6
				336	90.5
				364	90.6

Table A.4: Human body-weight data, part 1 (Van Gaal et al. 2008; Torgerson et al. 2004; Pi-Sunyer et al. 2006; Smith and Goulder 2001; Hauner et al. 2004; Aronne et al. 2010).

Topiramate, 96 mg Wilding et al. 2004		**Topiramate, 192 mg Wilding et al. 2004**		**Topiramate, 256 mg Wilding et al. 2004**	
Time (days)	BW (kg)	Time (days)	BW (kg)	Time (days)	BW (kg)
0	105.3	0	103.3	0	106.3
14	104.6	14	102.6	14	105.6
28	103.5	28	101.5	28	104.4
42	102.2	42	100.3	42	103.2
56	101.6	56	99.2	56	102.1
84	100.9	84	97.8	84	100.6
112	99.9	112	96.5	112	99.3
140	99.1	140	95.3	140	98.1
168	98.7	168	94.7	168	97.4
196	98.1	196	93.8	196	96.6
224	97.9	224	93.2	224	95.6
252	97.2	252	92.7	252	94.8
280	96.9	280	91.9	280	94.4
308	96.7	308	91.2	308	93.8
336	96.4	336	91.1	336	93.4
364	96.3	364	90.7	364	93.4
392	96.3	392	91.0	392	93.3
420	95.7	420	91.0	420	92.7

Rimonabant, 20 mg Van Gaal et al. 2005		**Rimonabant, 20 mg Despres et al. 2005**	
Time (days)	BW (kg)	Time (days)	BW (kg)
0	101.7	0	93.3
14	100.9	28	91.3
28	99.8	56	89.8
56	98.4	84	88.2
84	97.0	112	87.5
112	96.5	140	86.6
140	95.6	168	85.7
168	94.7	196	85.6
196	94.3	224	85.2
224	94.0	252	84.7
252	93.6	280	84.8
280	93.4	308	84.6
308	93.4	336	84.6
336	93.2	364	84.7
364	93.1		

Table A.5: Human body-weight data, part 2 (Wilding et al. 2004; Van Gaal et al. 2005; Després et al. 2005).

Lorcaserin, 10 mg bid Smith et al. 2010		**Phenterm.+Topiram. 3.75+23 mg Allison et al. 2012**		**Phenterm.+Topiram. 15+92 mg Allison et al. 2012**	
Time (days)	BW (kg)	Time (days)	BW (kg)	Time (days)	BW (kg)
0	100.4	0	118.4	0	115.1
14	98.7	28	115.5	28	110.8
28	97.4	56	114.3	56	108.3
56	96.1	84	113.5	84	106.5
84	94.9	112	112.4	112	104.9
112	94.0	140	111.8	140	103.6
140	93.5	168	111.1	168	102.4
168	92.9	196	110.6	196	101.5
196	92.6	224	110.4	224	100.9
224	92.4	252	110.2	252	100.4
252	92.2	280	109.9	280	99.9
280	92.1	308	110.0	308	99.9
308	92.2	336	110.2	336	99.9
336	92.4	364	110.3	364	99.7
364	92.3	392	110.4	392	100.0
392	92.8				
420	93.0				
448	92.9				
476	93.2				
504	93.3				
532	93.5				
560	93.8				
588	93.9				
616	94.0				
644	94.2				
672	94.4				
700	94.6				
728	94.9				

Table A.6: Human body-weight data, part 3 (Smith et al. 2010; Allison et al. 2012).

A.4 Exenatide data

Time (h)	2.5 mg	5 mg	7 mg	10 mg
0.25	1.91	3.91	13.78	4.63
0.5	4.68	8.34	19.21	11.67
1	7.14	14.37	21.82	20.32
2	5.37	14.57	23.83	23.94
4	5.93	13.36	32.28	27.15
8	3.02	8.95	39.93	21.42
12	1.57	5.07	22.02	11.16
16	1.22	4.76	20.41	7.14
24	1.21	3.31	16.09	6.14
30	0.60	2.44	6.94	5.63
36	0.60	2.40	4.53	5.03
48	1.18	2.71	4.93	4.02

Table A.7: Plasma concentration in pM for the short-term exenatide data (Fineman et al. 2011).

Time (weeks)	Vehicle	2.5 mg	5 mg	7 mg	10 mg
0	3.53	1.32	1.89	1.48	2.04
0.3	2.49	1.72	4.05	16.08	6.11
0.45	1.74	1.63	3.33	5.05	4.05
1	2.05	2.25	4.61	4.37	8.42
2	2.00	4.90	14.23	14.71	17.86
3	1.89	3.13	10.58	13.65	13.39
4	1.76	4.90	9.62	13.39	14.11
5	1.60	7.92	13.65	18.52	26.89
6	1.96	6.55	28.22	21.93	29.94
7	1.80	11.55	21.09	23.01	30.75
8	1.96	9.33	19.64	20.57	14.93
9	1.70	4.50	6.01	13.15	13.23
10	1.50	1.75	2.89	3.81	3.89
11	3.13	1.88	2.89	5.61	2.61
12	1.70	0.97	1.84	1.44	2.12

Table A.8: Plasma concentration in pM for the long-term exenatide data (Fineman et al. 2011).