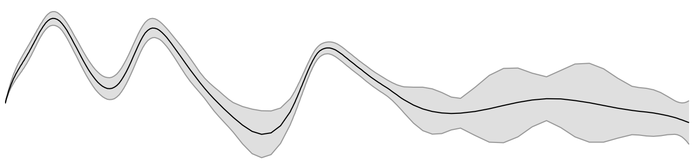
Stochastic PK/PD Modelling

Søren Klim and Stig Mortensen



Kongens Lyngby 2006
IMM-M.Sc.-2006-27

Supervisors: Henrik Madsen and Rune Viig Overgaard
External Supervisor: Niels Rode Kristensen

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

Abstract

This thesis describes the development of a software prototype implemented in Matlab for non-linear mixed effects modelling based on stochastic differential equations (SDEs). The setup aims at modelling measurements originating from more than one individual and it represents a powerful way of modelling systems with complicated and partially unknown dynamics. The incorporation of SDEs enables the setup to separate noise into two fundamentally different parts: uncorrelated measurement noise, arising from data collection etc. and correlated system noise, arising from model deficiencies or true random fluctuation of the system. The mixed-effects model makes it possible to describe variation within a population and to estimate parameters where only few observations are available for each individual.

The setup has been implemented in a prototype, which enables maximum likelihood estimation of model parameters. The likelihood function is created using the First-Order Conditional Estimate (FOCE) used in non-linear mixed effects modelling. This is done in combination with the Extended Kalman Filter used in models with SDEs. The prototype is able to use the estimated model for prediction, filtering, smoothing and simulation for linear as well as non-linear models.

The work using the implemented prototype has focused on pharmacokinetic/ pharmacodynamic (PK/PD) modelling and has been carried out in collaboration with Novo Nordisk A/S. The prototype is compared with existing software for a range of PK models, but also used to perform analysis that is not readily doable in any other software package. Particular attention is devoted towards deconvolution of insulin secretion rate (ISR) and liver extraction of insulin based on a 24h study with three standardized meals. Moreover, an intervention model is proposed which utilizes knowledge of the three meal times and this is used for modelling of the insulin secretion rate.

Overall, the prototype has proven to be a flexible and efficient tool for estimation of non-linear mixed effects models based on SDEs and has been used with success for a range of pharmacokinetic models.

KEY WORDS: stochastic differential equation (SDE), non-linear mixed effects, FOCE approximation, Extended Kalman Filter, maximum likelihood estimation, insulin secretion rate, pharmacokinetic, PK/PD modelling

Resumé

Dette eksamensprojekt omhandler implementeringen af en software prototype i Matlab for en ikke-lineær mixed effekt model baseret på stokatiske differentialligninger. Dette setup tilsigter modellering af målinger fra mere end et individ, og det repræsenterer en effektiv måde at modellere systemer med kompliceret og delvis ukendt dynamik. Brugen af stokatiske differentialligninger gør setup'et i stand til at adskille støj i to fundamentale forskellige dele: ukorreleret målestøj, der typisk stammer fra data opsamling samt korreleret system støj, der stammer fra model mangler eller reelle tilfældige ændringer i systemet. Brugen af mixed effekt modellen gør det muligt at beskrive variationerne indenfor en population og at estimere parametre i situationer, hvor der kun er få observationer tilgængelige fra hvert individ.

Setup'et er blevet implementeret som en prototype og kan blandt andet bruges til parameter estimation baseret på *maksimum likelihood* teorien. Likelihoodfunktionen er dannet ved hjælp af en første ordens betinget estimations metode, der ofte bruges i ikke-lineære mixed effekt modeller. Dette er gjort i kombination med Kalman filteret, som kan bruges i modeller med stokatiske differentialligninger. Prototypen muliggør prædiktion, filtrering, udglatning og simulering for både lineære og ikke-lineære systemer ved hjælp af den estimerede model.

Arbejdet med brugen af den implementerede prototype er fokuseret omkring pharmakokinetiske og pharmakodynamiske modeller og er blevet udført i samarbejde med Novo Nordisk A/S. Prototypen vil blive sammenlignet med eksisterende programmer ved brug af en række PK/PD modeller, og den vil blive brugt til analyser, som på nuværende tidspunkt ikke er mulige at udføre med andre programmer. Analysen vil blive fokuseret på foldning af insulin sekretionsraten samt leverekstraktion af insulin. Data stammer fra et 24 timers forsøg med 3 standardiserede måltider. Endvidere vil en interventions model blive præsenteret til estimation af insulin sekretionsraten ved udnytte viden omkring måltidernes serveringstidspunkter.

Overordnet set har prototypen vist sig at være et fleksibelt og effektivt værktøj til estimering af ikke-lineære mixed effekt modeller baseret på stokastiske differentialligninger. Prototypen har endvidere vist sig at være brugbar til analyse af en række pharmakokinetiske modeller.

Preface

This master thesis was prepared at the institute of Informatics and Mathematical Modelling at the Technical University of Denmark in fulfillment of the requirements for acquiring the Master degree in engineering. The work started September 5, 2005.

The thesis deals with pharmacokinetic and pharmacodynamic modelling based on stochastic differential equations in a mixed-effects setup. A prototype has been implemented in Matlab and different pharmacokinetic models within diabetes has been analysed.

Acknowledgements

We wish to thank our supervisors Henrik Madsen, Rune Viig Overgaard and Niels Rode Kristensen for their support, guidance and participation in discussions during the project. We would also like to thank John Bagterp Jørgensen for his help concerning differential equations and Bernd Dammann for assisting on computational issues.

Lyngby, March 6 - 2006

Søren Klim   Stig Mortensen

Contents

Abstract **i**

Resumé **iii**

Preface **v**

Symbols and Abbreviations **xi**

1 Introduction **1**

1.1 Diabetes 2

1.2 Outline of thesis 4

2 PK/PD Modelling **5**

2.1 Models 6

3 Single Subject Modelling **11**

3.1 State space models 11

3.2 SDEs and stochastic state space models 13

3.3 Extended Kalman filter 14

3.4 Parameter estimation 18

3.5 Smoothing 20

3.6 Simulation 23

4 Population Modelling 25

4.1 Model definition 26

4.2 Parameter estimation 26

4.3 Parameter uncertainty 28

5 Population Stochastic Modelling using PSM 31

5.1 Model specification 32

5.2 Extended Kalman filter implementation 32

5.3 Minimizers 33

5.4 Minimization of APL 35

5.5 Parameter uncertainty 38

5.6 Parallel computing 39

6 Validation of PSM 41

6.1 Kalman filter 42

6.2 Population likelihood function 44

6.3 Simulation 49

7 Insulin Secretion **53**

7.1 Data 53

7.2 Deconvolution of ISR 54

7.3 Intervention model for ISR 67

7.4 Combined models 74

8 Future Work with PSM **91**

9 Discussion **95**

10 Conclusion **99**

A Matlab Implementation **103**

A.1 Considerations 103

A.2 Function specification 107

B ISR **115**

B.1 2 compartment C-peptide model 115

B.2 2 compartment C-peptide log Model 116

B.3 Modelling of ISR 118

B.4 Intervention Model 129

B.5 Model validation for the combined model 134

B.6 Liver Extraction Model 137

B.7 Constrained Liver Extraction Model 140

Symbols and Abbreviations

Symbols

e_k	Residual at time point t_k
ϵ_{prop}	Proportional error term
$\boldsymbol{\eta}$	Random effects
$\boldsymbol{f}(\cdot)$	Non-linear state function - drift
F	Liver extraction rate
$\boldsymbol{h}(\cdot)$	Non-linear output function
L	Population likelihood
l	Observation/Measurement dimension
L_i	Likelihood for individual i.
l_i	Log-likelihood for individual i.
m	Input dimension
N	Number of individuals
n_i	Number of observations for individual i
$\boldsymbol{\Omega}$	Random effects covariance
$\boldsymbol{\phi}_i$	Individual parameters.

$\boldsymbol{S}$	Measurement noise
s	State dimension
σ	Wiener magnitude
$\boldsymbol{\theta}$	Model Parameters
$\boldsymbol{u}_k$	Input at time point t_k
v	Dimension on the random effects
$\boldsymbol{x}_t$	States at time point t
y_{ij}	jth observation for individual i.
$\boldsymbol{y}_k$	Output at time point t_k
$\boldsymbol{Z}_i$	Covariate for the ith individual.

Abbreviations

ACF	Auto Correlation Function
CV	Coefficient of Variation
EKF	Extended Kalman filter
IMM	Informatics and Mathematical Modelling
ISR	Insulin Secretion Rate
KF	Kalman filter
ODE	Ordinary Differential Equations
PD	Pharmacodynamics
PK	Pharmacokinetics
PSM	Population Stochastic Modelling
QQ-plot	Quantile Quantial plot
SDE	Stochastic Differential Equation
SSM	State Space Model
SSSM	Stochastic State Space Model

Chapter 1

Introduction

This thesis deals with the creation of a software prototype, that is able to handle non-linear mixed effects models based on stochastic differential equations (SDEs). This model framework will accommodate a need for better modelling tools to aid in improving existing medication and the development of new drugs in clinical trials. In recent years the focus on pharmacokinetic (PK)/pharmacodynamic (PD) modelling has intensified. This is among other things due to the American Food and Drug Administration who has encouraged a wider use of PK/PD modelling in clinical trials [FDA 1999].

PK/PD modelling is used to describe the properties of a drug starting from when it is introduced into the body, to the efficacy and toxicity of the drug and finally describing how it leaves the body again. In PK/PD various modelling methods are used, each for their specific purpose. The non-linear mixed effects model has the ability to include measurements for several subjects from the same experiment by splitting the variation into intra- and inter individual variation.

Stochastic differential equations in state space models are able to separate measurement and system noise which enables it to account for model deficiencies. Correlated residuals, which are often the result of modelling based on ordinary differential equations, can be handled with the use of SDEs. This property is gaining increased popularity. At present no standard program is able to handle both non-linear mixed effects with SDEs.

The main goal of the thesis is the implementation of a prototype able to handle the combination of the non-linear mixed effects model and the stochastic differential equation used in a state space setup. The prototype will be used to analyse data from a clinical trial within diabetes research. Particular attention is devoted towards deconvolution of insulin secretion rate (ISR) and liver extraction of insulin based on a 24h study with three standardized meals.

The topic of PK/PD modelling has been chosen due to personal interest and an opportunity to write the thesis in cooperation with Novo Nordisk A/S.

A model for ISR and models for the insulin liver extraction will be proposed and analysed. These PK/PD models will be used to illustrate the properties of SDEs in a mixed effects setup. The implementation process should result in a functional prototype and also provide experiences that can be used in a final implementation. The work with the PK/PD models should also yield best practises and desired functionality in the program.

The ambition with the thesis is that it will provide a step towards improved PK/PD modelling and accommodate better insight into the properties of drugs. For Novo Nordisk, this may potentially lead to improved diabetes treatments or new drugs. The following section provides motivation for an increased focus on PK/PD modelling within diabetes.

1.1 Diabetes

In Denmark approximately, 150.000 people are diagnosed with diabetes and another 150.000 are expected to be unaware of their diabetes condition [Sundhedsministeriet 2003]. Worldwide, 171 millions are diagnosed with diabetes and by 2030 [WHO 2006] the number is expected to be 366 millions. The unfortunate tendency is that people getting diagnosed with diabetes already suffer from complications that could have been avoided by earlier treatment.

In Denmark, the cost of diabetes arises to around 2.5 billions Danish kroner a year. The main part of the cost covers the treatment of complications due to late diagnose of illness. The complications and long term effects of diabetes are an increased risk for heart diseases, blindness, nerve damage and kidney damage. The complications can be reduced or postponed significantly by proper treatment improving the life of the patient and reducing the cost to society.

The full name for the disease is diabetes mellitius. Mellitus is a greek word which translates into honey or honey sweet. This word describes the urine from untreated diabetes patients which contains a high concentration of glucose since it is the main source of elimination of unused glucose from the body.

For healthy persons, carbohydrates from food intake are split into glucose molecules which are converted into energy in the mitochondrions in the cells. The insulin hormone is used to unlock the cellular walls and give access for the glucose molecules. Insulin is produced in the β-cells in the pancreas and is distributed through the body by the blood.

Diabetes is split into several different types of which Type 1 and 2 are the most common ones. Type 1 diabetes is caused by an autoimmune destruction of the β-cells resulting in a reduced or even a complete lack of production of insulin. Type 1 diabetes is often diagnosed in childhood and results in a lifetime dependency on insulin injections. Type 2 diabetes is an insulin resistance where cells do not respond to insulin. Type 2 diabetes is a slowly progressing disease and can go unnoticed for many years. Type 2 is the kind of diabetes that is related to the life style and in particular the life style of the western world. It is treated with changes in diet, weight loss and exercise which can often reduce the complications and keep diabetes under control. Earlier, only elderly people where diagnosed with Type 2 diabetes but now patients as young as 15 years old have become more common.

Diabetes is the most common metabolic disease in the world and the number of diagnosed will increase. Even small improvements in the understanding of diabetes could results in better insulin dosage regimes. This could have immense effect on population scale but also improve the quality of life for the individual patient.

1.2 Outline of thesis

Chapter 2 deals with the concepts of Pharmacokinetic and Pharmacodynamic modelling. The pharmacokinetics principles used in this thesis are reviewed and frequently used PK/PD models are introduced.

Chapter 3 explains the theory of single subject modelling. The use of stochastic differential equations in state space models is explained and the Kalman filter is presented as a solution to handle filtering. Finally, maximum likelihood parameter estimation is introduced.

The theory for one individual is extended in **Chapter 4** where the theory for non-linear mixed effects is reviewed. An extended likelihood for the non-linear mixed effect model is established.

Chapter 5 describes the creation of the software prototype and the numerical issues encountered during the implementation process. The implemented prototype is validated against NONMEM and CTSM in **Chapter 6**.

In **Chapter 7**, the implemented prototype is used in a PK/PD setting where several PK/PD models are analysed. The data modelled originates from a clinical trial within diabetes.

The experiences from the implementation and the model building are summarized into a number of recommendations for the next implementation, all found in **Chapter 8**.

Chapter 9 holds the discussion of the results from the PK/PD models, built with the implemented prototype.

Chapter 10 summarizes the discussion and concludes on the goal of the thesis.

Chapter 2

PK/PD Modelling

Pharmacokinetics and Pharmacodynamics modelling covers the area of mathematical models describing the properties of a drug. PK/PD modelling is used within clinical drug development to provide information on the properties of a drug and is especially good at describing the dose-effect relationship [Tornøe 2005]. Moreover, PK/PD modelling is aiding in providing more efficient drug development.

PK describes the movements of a drug and PD describes the pharmacological effect at its destination. The processes described by PK are generally categorized in three different phases, namely the absorbtion after administration, distribution and elimination of a drug. The absorbtion describes the movement of a drug into systemic circulation in the body and it is often modelled with a compartment describing the site of administration. This compartment could be the depot from a subcutaneous injection or the gastrointestinal tract in an oral administration. Distribution describes the movement of the drug into the circulation typically blood and tissue. The distribution can be modelled through flows, mass conservation and other laws from physics and biology. Elimination is the removal of the drug from the body. This could be via the kidneys, excretion or metabolism.

The main focus of this project is within PK and therefore the introduction and models are mainly focused on PK modelling.

In Figure 2.1 the differences between PK and PD and the three phases are illustrated.

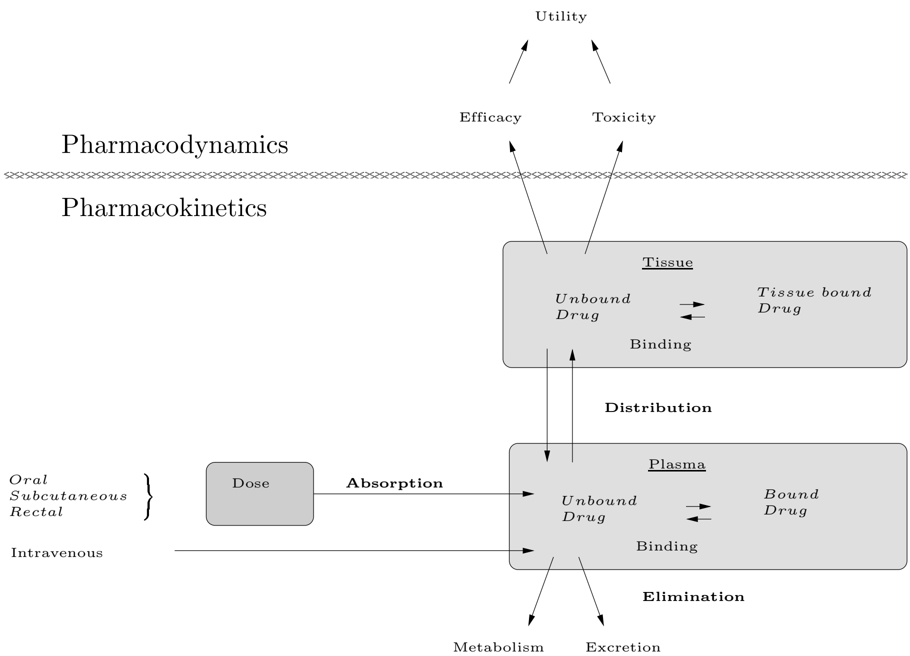


Figure 2.1: Schematic representation of PK/PD. [Gabrielsson & Weiner 1997]

The modelling process uses mathematical and statistical tools combined with biological and pharmacological knowledge resulting in a grey-box model. The word grey-box refers to a mixture of empirical models based solely on data (black-box) combined with theoretical physiological models (white-box).

2.1 Models

The models used in this thesis do not involve actual drugs instead it involves hormones and substances already found in the body. The methods and models from PK modelling can easily be transferred to the actual use. It also means that the use of PK/PD principles in this report lies outside the general definition of PK/PD modelling.

The models are based on mathematical equations describing the dynamics of the drug. The model equations can be stated by focusing on mass conservation and Fick's laws of diffusion between compartments[Warberg 1991, page 161].

The simplest model is the one-compartment model and it can be seen in Figure 2.2. The model equations are often stated as differential equations of order one. This means that the changes in concentration depends on the current concentration.

The analytical solution to a first order ordinary differential equation (ODE) is an exponential decay and the one-compartment model is also referred to as the mono-exponential model.

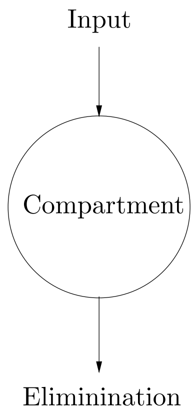


Figure 2.2: One-compartment model. [Gabrielsson & Weiner 1997, p. 60]

The one-compartment model is used when the drug has little distribution to tissue. The drug concentration in the compartment is eliminated over time and is generally modelled as dependent on the current concentration. The input can be an instantaneous intravenous injection or a slower dosage regime.

$$\frac{d\boldsymbol{C}}{dt} = \boldsymbol{I}(t) - k_e\boldsymbol{C} \tag{2.1}$$

where $\boldsymbol{C}$ is the current drug concentration in the main compartment, k_e is the elimination constant and $\boldsymbol{I}(t)$ is the input into the compartment at time t e.g. an injection.

The two-compartment model has an extra peripheral compartment to model the distribution into tissue and back into plasma again.

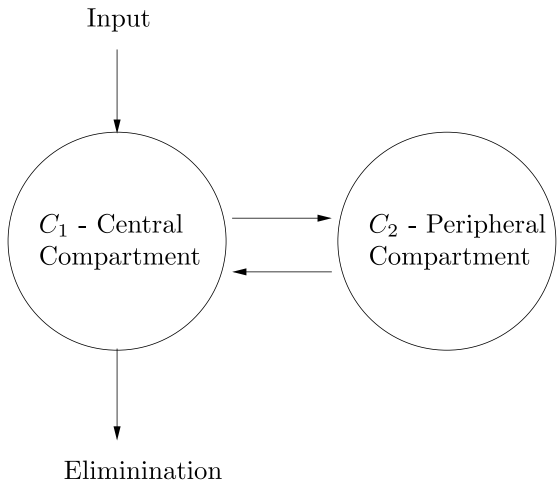


Figure 2.3: Two-compartment model. [Gabrielsson & Weiner 1997, p. 85].

The two-compartment model is able to handle situations where an equilibrium in concentration in tissue and plasma is not reached immediately.

$$\begin{aligned} \frac{d\boldsymbol{C}_1}{dt} &= \boldsymbol{I}(t) - k_1\boldsymbol{C}_1 + k_2\boldsymbol{C}_2 - k_e\boldsymbol{C}_1 \\ &= \boldsymbol{I}(t) - (k_1 + k_e)\boldsymbol{C}_1 + k_2\boldsymbol{C}_2 \qquad (2.2) \\ \frac{d\boldsymbol{C}_2}{dt} &= k_1\boldsymbol{C}_1 - k_2\boldsymbol{C}_2 \qquad (2.3) \end{aligned}$$

where k_1 and k_2 describes the distribution between compartment 1 and 2. k_e is the elimination from compartment 1.

The models in PK/PD Modelling are generally linear but non-linear models describing more complicated processes are used.

2.1.1 Diabetes Modelling

The modelling of data in connection with diabetes is typically based on *glucose*, *C-peptide* and *insulin* measurements.

A deeper understanding of the processes surrounding diabetes will aid in the model formulation. Insulin is used by the cells to produce energy from glucose molecules in the mitochondrion. High levels of glucose will for a healthy person result in a high secretion of insulin. In biological terms the glucose and insulin have a negative feedback meaning that a increase in glucose will result in an increase in insulin which will result in a decrease in glucose [Warberg 1991, page 213].

C-peptide is a by-product from the insulin production in pancreas and importantly it is produced in equimolar amounts. Insulin and C-Peptide is secreted directly into the bloodstream and passes through the liver before entering the systemic circulation. A large portion of the secreted insulin is extracted in the liver where it is used in connection with the stored glucose (glucogen). C-peptide passes directly through the liver and is eliminated from the body via elimination from the kidneys. The half-life insulin is approximately 5 minutes whereas the half-life for C-peptide is 30 minutes. This means that C-peptide is often used as an indicator of the insulin secretion due to this rapid decrease in insulin over time.

The extraction of insulin in the liver is not constant and is quite often modelled using a Michaelis-Menten saturation. The liver has an upper limit for excretion pr. time and in periods with high insulin secretion this limit is reached. This threshold limit in the liver results in a non-linear output of insulin from the liver. This process can be modelled using a Michaelis-Menten saturation.

The measurements of insulin, glucose and C-peptide originate from blood samples from the arms. The observations will due to measurement techniques have an uncertainty that should be incorporated into the model.

The non-linear mixed effects model with SDEs is ideal to model clinical trials as measurements from all subjects on the same process can be included and the variation arising from individual differences and deviations from the specified model can be handled. As an example within diabetes the individual differences could be the steady state level of insulin secretion.

This modelling of individual variation is valuable when analyzing and optimizing the treatment of patients. The aim is a treatment supported by information on individual variation and thereby adapted for the individual patient.

Chapter 3

Single Subject Modelling

This chapter will describe techniques used when modelling a single subject. This is referred to as a 1st stage model. The emphasis is on reviewing the mathematical and statistical methods and their properties used in this project.

Many mathematical models evolve around a physical problem. The modelling of a system begins with the understanding on how the systems internal states interact. e.g. how wind speed affects the wind power production or how temperature affects bacterial growth. When dealing with dynamical systems it is of interest to know the states of the system in order to monitor the performance or condition of the system. These internal states are useful for prediction but also for control purposes. One way of modelling using internal states of the system is called state space modelling. The next section will give an introduction to this topic.

3.1 State space models

This section will introduce State Space Models (SSM) which form the basis for most of this project. The state space approach for modelling a system is based on states which are observed indirectly as a function of the space observations.

The states are modelled by differential equations which describes the dynamics of the unobservable part of the system. Measurements are modelled through a space equation which is a function of the states, where this function is observed with noise.

The set of state/space equations are for most practical purposes either discrete/discrete or continuous/discrete. In many physical systems the state model is continuous and measurements are sampled at discrete time intervals. This is also the case in PK/PD modelling and therefore the continuous/discrete version will be used in this thesis.

State space models hold the property of being a Markov process as all information about the system is embedded in the current states. This means that all available information at present time is found in the current states and prediction is therefore based solely on current model states.

The general mathematical formulation of a state space model is shown in Eq. (3.1) and Eq. (3.2). This formulation can account for non-linearities in the states and time variability in parameters.

$$\frac{d\boldsymbol{x}_t}{dt} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_k, t, \boldsymbol{\theta}) \tag{3.1}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k, \boldsymbol{\theta}) + \boldsymbol{e}_k \tag{3.2}$$

where $k = 1...n$, $t \in \mathbb{R}$ is time, $\boldsymbol{x}_t \in \mathbb{R}^s$ is a vector of state variables, $\boldsymbol{u}_t \in \mathbb{R}^m$ is a vector of input variables, $\boldsymbol{y}_k = \boldsymbol{y}_{t_k} \in \mathbb{R}^l$ is a vector of output variables and $\boldsymbol{\theta}$ is a vector of parameters. $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$ are known non-linear functions and $e_k \in N(\boldsymbol{0}, \boldsymbol{S}(\boldsymbol{u}_k, t_k, \boldsymbol{\theta}))$.

This formulation uses standard ordinary differential equations (ODEs) to model the state system. A consequence of using ODEs for modelling the state is that states are predicted without uncertainty incorporated into the model. This highly deterministic behavior of the states is not well suited for PK/PD modelling as it results in all unmodelled variation being categorized as measurement noise.

The ODE basis for the state space models, limits the number of systems it is able to handle. A much wider class of systems can be modelled by extending the ODE to Stochastic Differential Equations (SDE). An in depth review of the theory of SDEs is a substantial topic and is beyond the scope of this thesis. However, a general introduction to SDEs and their properties are needed to understand the pros and cons.

3.2 SDEs and stochastic state space models

The standard state space model uses ODEs to model the dynamics of the states. By using SDEs noise is allowed into the state equations which is beneficial within many fields including PK/PD modelling [Øksendal 1995], [Kristensen et al. 2005] and [Jazwinski 1970].

The extension to a state space model with SDEs is called a stochastic state space model. By using SDEs the states are assumed to evolve with a stochastic behavior and are thus predicted with uncertainty as opposed to state space models based on ODEs. This is useful in system where fluctuations or disturbances are expected to be present in the states. These disturbances may be due to unmodelled dynamics. A powerful property of the SDE extension is the possibility to split variation into correlated system noise and uncorrelated observation noise.

The extension to the state space models can be seen in equations (3.3) and (3.4).

$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, t, \boldsymbol{\theta})dt + \boldsymbol{\sigma}(\boldsymbol{u}_t, t, \boldsymbol{\theta})d\boldsymbol{\omega}_t \tag{3.3}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k, \boldsymbol{\theta}) + \boldsymbol{e}_k \tag{3.4}$$

where $\boldsymbol{\omega}_t$ is a standard Wiener process also often referred to as a random walk, and it holds the property $\omega_{t_2} - \omega_{t_1} \in N(0, |t_2 - t_1|I)$. $\boldsymbol{f}(\cdot)dt$ is called the drift term and $\boldsymbol{\sigma}(\boldsymbol{u}_t, t, \boldsymbol{\theta})d\boldsymbol{\omega}_t$ is called the diffusion term where $\boldsymbol{\sigma}(\cdot)$ is the magnitude of the Wiener process.

The notation using $d\boldsymbol{x}/dt$ cannot be used since the quotient $d\omega/dt$ is ill defined due to the changing values of $\boldsymbol{\omega}$. Several methods exists for dealing with the integral of the diffusion term where one of the them is the Itô method which has been chosen in this thesis. The diffusion is in rare cases possible to solve analytically but must generally be solved numerically.

The model formulation reduces back into a system of ordinary differential equations when the magnitude $\boldsymbol{\sigma}$ is zero. The chosen scheme for estimating the states of the model is the Kalman filter explained in the following.

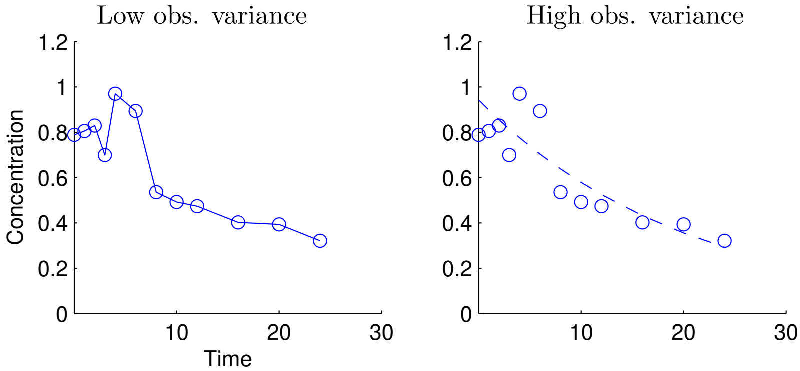


Figure 3.1: EKF filtering for different variances.

3.3 Extended Kalman filter

R.E. Kalman published his article on "A new approach to linear filtering and prediction" in 1960 [Kalman 1960]. The Kalman filter has since proven useful within a range of applications. The Kalman filter is a recursive minimum variance estimator for the states. The Kalman filter minimizes the prediction variance for the model by combining the observation and model variance and propagating it in time.

The Kalman filter is a 2 part algorithm consisting of *prediction* and *updating*. In the prediction part the current states and covariances are used to create a prediction of states, covariance and observation to a time point t_k given the information in time t_{k-1}. In the updating part an observation has just arrived and the states and covariances are updated accordingly.

The updating is based on a compromise between the observation and current model state. In a situation where the model is good but the observations are dominated by measurement error, the Kalman states should rely on the model as opposed to fitting the observations. Vice versa if the model is incomplete the states should rely more on the observations than the model. This trust in model versus observations can be adjusted by the magnitude of system noise $\boldsymbol{\sigma}$ and observation noise $\boldsymbol{S}$.

In Figure 3.1 the two extremes of observation variances can be seen. The example shows an exponential decay with a log-normal observation noise. If S is low the filtered output is almost identical to the observations and for high S the filtered output follows the state model.

In the present thesis the Kalman filter is based on the equations found in [Kristensen & Madsen 2003, p.12] and [Gelb et al. 1982, p.188]. In the following a review of the Extended Kalman Filter (EKF) will be given [Kalman & Bucy 1961] [Tornøe 2005]. Although the linear Kalman filter (KF) is also used in the thesis, it will not be presented here, since EKF reduces into KF for linear systems. The presented EKF is able to handle multi-variate input sampled at measurement times. It has been decided to use zero-order hold for the input meaning that it is assumed constant in between measurements.

As previously mentioned the Kalman filter consists of two parts - an updating and a prediction part. Initial conditions are needed to start the recursive filter as specified below.

$$\hat{\boldsymbol{x}}_{1|0} = \boldsymbol{x}_0 \tag{3.5}$$

$$\boldsymbol{P}_{1|0} = \boldsymbol{P}_0 = \int_{t_1}^{t_2} e^{\boldsymbol{A}_t s} \boldsymbol{\sigma}\boldsymbol{\sigma}^T (e^{\boldsymbol{A}_t s})^T ds \tag{3.6}$$

where $\boldsymbol{P}_{t|t-1}$ is the prediction covariance on the states. The initial prediction covariance is either specified or generally estimated using the time interval between the two first observations, the model dynamics and the magnitude of the system noise as seen in Eq. (3.6). The latter is chosen in this thesis.

$\boldsymbol{A}_t$ is a linearization around the state prediction $\hat{\boldsymbol{x}}_{(t|k)}$ based on the predicted state, defined as:

$$\boldsymbol{A}_t = \left.\frac{d\boldsymbol{f}}{d\boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{t|k}} \tag{3.7}$$

In the linear case $\boldsymbol{A}$ is independent of the state and may also be time invariant. The initial conditions are used to form the prediction for the first observation. The prediction equations are shown in the following.

$$\frac{d\hat{\boldsymbol{x}}_{t|k}}{dt} = \boldsymbol{f}(\hat{\boldsymbol{x}}_{t|k}, \boldsymbol{u}_k, t, \theta) \tag{3.8}$$

$$\frac{d\boldsymbol{P}_{t|k}}{dt} = \boldsymbol{A}_t \boldsymbol{P}_{t|k} + \boldsymbol{P}_{t|k} \boldsymbol{A}_t^T + \boldsymbol{\sigma}\boldsymbol{\sigma}^T \tag{3.9}$$

where $t \in [t_k, t_{k+1}]$. $\boldsymbol{u}_k$ is the input at time t_k

The prediction of the output from the system is based on the predicted states and the predicted covariance.

$$\hat{\boldsymbol{y}}_{k|k-1} = \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{u}_k, t_k, \boldsymbol{\theta}) \tag{3.10}$$

$$\boldsymbol{R}_{k|k-1} = \boldsymbol{C}_k \boldsymbol{P}_{k|k-1} \boldsymbol{C}_k^T + \boldsymbol{S} \tag{3.11}$$

The output prediction is calculated as the output given the predicted state. The observation covariance $\boldsymbol{S}$ forms a lower boundary for the expected 1-step prediction covariance and $\boldsymbol{C}$ propagates the state covariance into the expected prediction covariance.

$$\boldsymbol{C}_k = \left.\frac{d\boldsymbol{h}}{d\boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k|k-1}} \tag{3.12}$$

$\boldsymbol{C}$ is in the linear case state independent and again as with $\boldsymbol{A}$ it may also be time invariant.

The updating part of the Kalman filter occurs when an observation has just arrived and the states and covariances should be updated accordingly to the residual ϵ_k.

$$\boldsymbol{\epsilon}_k = \boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1} \tag{3.13}$$

The states are updated by the Kalman gain $\boldsymbol{K}$ which is a combination of the predicted state covariance and prediction covariance.

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{C}_k^T \boldsymbol{R}_{k|k-1}^{-1} \tag{3.14}$$

The compromise of covariances discussed earlier is connected directly to the value of the Kalman gain. Large observations covariances leads to a small Kalman gains resulting in the updated state being almost identical to the predicted - hence following the model closely.

The updated state and covariance is often referred to as the filtered estimate of the state and covariance.

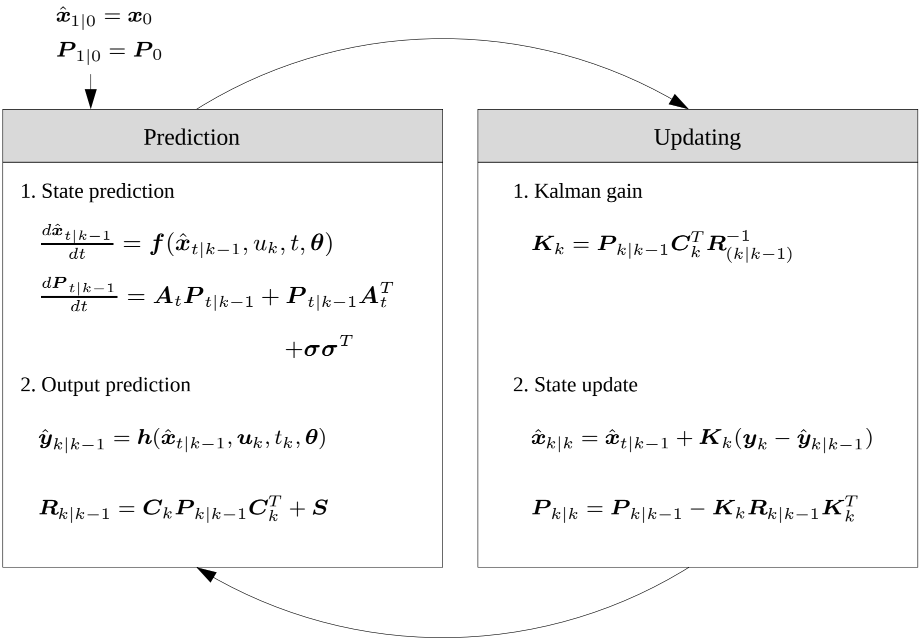


Figure 3.2: Schematic overview of the Kalman filter. [Welch & Bishop 2004]

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \epsilon_k \tag{3.15}$$

$$\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k \boldsymbol{R}_{k|k-1} \boldsymbol{K}_k^T \tag{3.16}$$

The Extended Kalman filter explained in this section is summarized in Figure 3.2. The figure shows the recursive structure repeated for every observation with an updating and a prediction part.

The Extended Kalman filter reduces into the ordinary Kalman filter when the functions $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$ are linear. The approximations $\boldsymbol{A}$ and $\boldsymbol{C}$ are for this case exact and thus in the linear case the Kalman filter is an exact minimal variance estimator for the state filtering problem.

3.4 Parameter estimation

The maximum likelihood principle is used to estimate parameters for the stochastic state space model. This is also denoted the first stage model and this section introduces the first stage likelihood function [Overgaard et al. 2005] [Kristensen & Madsen 2003].

The optimal set of parameters is found by maximizing the likelihood L_i for the model. The notation with subscript i is bound to individual i. It is not important at present but becomes relevant when the likelihood for an entire population is examined. For the same reason the data structure is defined for more than one individual although it is not used in the first stage model.

The data from a study on a number of individuals has the following general structure

$$\boldsymbol{y}_{ij}, \quad i = 1, ..., N, \quad j = 1, ..., n_i \tag{3.17}$$

where N is the number of individuals and n_i is the number of measurements for individual i. Note that the number of measurements for each individual can vary.

The likelihood function is a product of the probabilities for each observation. When working with SDEs, the residuals from the ODE part of the model are correlated and it is thus necessary to use conditional densities to form the likelihood function. By introducing the notation $\mathcal{Y}_{ij} = [y_{i1}, y_{i2}, ..., y_{ij}]$ the first stage likelihood function is defined as

$$L_i(\boldsymbol{\theta}|\mathcal{Y}_{in_i}) = p_1(\mathcal{Y}_{in_i}|\boldsymbol{\theta}) \tag{3.18}$$

and by applying Bayes rule $P(A \cap B) = P(B|A)P(A)$ it follows that

$$\begin{aligned} L_i(\boldsymbol{\theta}|\mathcal{Y}_{in_i}) &= p(y_{in_i}|\mathcal{Y}_{i(n_i-1)}, \boldsymbol{\theta})\ p_1(\mathcal{Y}_{i(n_i-1)}|\boldsymbol{\theta}) && (3.19) \\ &= p(y_{in_i}|\cdot)\ p(y_{i(n_i-1)}|\cdot)\ p_1(\mathcal{Y}_{i(n_i-2)}|\boldsymbol{\theta}) && (3.20) \\ &\vdots \\ &= \left(\prod_{j=2}^{n_i} p(y_{ij}|\mathcal{Y}_{i(j-1)}, \boldsymbol{\theta}) \right) p(y_{i1}|\boldsymbol{\theta}) && (3.21) \end{aligned}$$

By approximation the conditional density of each observation in Eq. (3.21) is assumed to be Gaussian. This is identical to the assumption for the EKF, and therefore the conditional density for each observation is given by the one-step prediction density from EKF

$$\begin{aligned} \hat{\boldsymbol{y}}_{i(j|j-1)} &= E(\boldsymbol{y}_{ij}|\boldsymbol{\theta}, \mathcal{Y}_{i(j|j-1)}) && (3.22) \\ \boldsymbol{R}_{i(j|j-1)} &= V(\boldsymbol{y}_{ij}|\boldsymbol{\theta}, \mathcal{Y}_{i(j|j-1)}) && (3.23) \end{aligned}$$

By using the above, the conditional density of the one-step prediction error is approximated by

$$\boldsymbol{\epsilon}_{ij} = \boldsymbol{y}_{ij} - \hat{\boldsymbol{y}}_{i(j|j-1)} \in N(\mathbf{0}, \boldsymbol{R}_{i(j|j-1)}) \tag{3.24}$$

In order to evaluate the likelihood function L_i the multivariate normal distribution is used for calculating the probability $p(\boldsymbol{y}_{ij}|\boldsymbol{\theta}, \mathcal{Y}_{i(j|j-1)})$

$$\begin{aligned} p(\boldsymbol{y}_{ij}|\boldsymbol{\theta}, \mathcal{Y}_{i(j|j-1)}) &= p(\boldsymbol{\epsilon}_{ij}|\boldsymbol{\theta}, \mathcal{Y}_{i(j|j-1)}) && (3.25) \\ &\approx |2\pi \boldsymbol{R}_{i(j|j-1)}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{\epsilon}_{ij}^T \boldsymbol{R}_{i(j|j-1)} \boldsymbol{\epsilon}_{ij}\right) && (3.26) \end{aligned}$$

The individual likelihood function can now be defined given Eq. (3.21) and the conditional density in (3.26) such that

$$L_i(\boldsymbol{\theta}|\mathcal{Y}_{in_i}) \approx \prod_{j=1}^{n_i} \frac{\exp\left(-\frac{1}{2}\boldsymbol{\epsilon}_{ij}^T \boldsymbol{R}_{i(j|j-1)} \boldsymbol{\epsilon}_{ij}\right)}{\sqrt{|2\pi \boldsymbol{R}_{i(j|j-1)}|}} \tag{3.27}$$

By further taking the logarithm of L_i gives the individual log-likelihood function defined as

$$\log L_i(\boldsymbol{\theta}|\mathcal{Y}_{in_i}) \approx -\frac{1}{2}\sum_{j=1}^{n_i} \left(\boldsymbol{\epsilon}_{ij}^T \boldsymbol{R}_{i(j|j-1)} \boldsymbol{\epsilon}_{ij} + \log|2\pi \boldsymbol{R}_{i(j|j-1)}|\right) \tag{3.28}$$

It should be noted that an initial state must be given in order to evaluate $\log L_i$ with EKF, but this condition can be included into the parameters to be maximum likelihood estimated. For a linear system where EKF reduces to KF the likelihood function in Eq. (3.28) is exact.

The parameters in the stochastic state space model is estimated by maximizing the log-likelihood function. The parameters in the model can be anything from the covariance elements, initial starting conditions or model parameters.

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \ \log L_i(\theta|\mathcal{Y}_{in_i}) \tag{3.29}$$

3.5 Smoothing

The Kalman filter provides the minimal variance estimate for the state prediction and filtering problem. In post hoc analysis of a study it is of interest to use all available data to obtain an optimal estimate at every time point. This is called the state smoothing problem. A smoothed estimator is based on information from all the observations. That is before and after the time of interest whereas the estimates previously discussed have been based on past observations.

The smoothed estimate can be created by combining a forward prediction and a backward filtering. By combining the two filtering sweeps all information from observations is included into the smoothed estimate at each time point. The following two sections contain descriptions of the linear and non-linear smoother.

3.5.1 Linear smoothing

For the linear model the Bryson Frazier algorithm has been used to create a smoother. The Bryson Frazier algorithm can be found in Kailath et al. (2000, pages 373-374).

The forward prediction is performed with the linear Kalman filter. The backward filter and the combined smoothed estimate are described below.

The initial conditions are shown below

$$\boldsymbol{\lambda}_{N+1|N} = 0 \tag{3.30}$$
$$\boldsymbol{\Lambda}_{N+1|N} = 0 \tag{3.31}$$

The filtering equations.

$$\begin{aligned}
\boldsymbol{F}_k &= \boldsymbol{A}_k - \boldsymbol{K}_{p,k}\boldsymbol{C}_k &(3.32)\\
\boldsymbol{\lambda}_{k|N} &= \boldsymbol{F}_k^T\boldsymbol{\lambda}_{k+1|N} + \boldsymbol{C}_k^T\boldsymbol{S}_k^{-1}\boldsymbol{e}_k &(3.33)\\
\boldsymbol{\Lambda}_{k|N} &= \boldsymbol{F}_k^T\boldsymbol{\Lambda}_{k+1|N}\boldsymbol{F}_k + \boldsymbol{C}_k^T\boldsymbol{S}_k^{-1}\boldsymbol{C}_k &(3.34)
\end{aligned}$$

where the prediction Kalman gain is

$$\boldsymbol{K}_{p,k} = \boldsymbol{A}_k\boldsymbol{K}_k \quad (3.35)$$

The smoothing equations are

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k|N} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{P}_{k|k-1}\boldsymbol{\lambda}_{k|N} &(3.36)\\
\boldsymbol{P}_{k|N} &= \boldsymbol{P}_{k|k-1} - \boldsymbol{P}_{k|k-1}\boldsymbol{\Lambda}_{k|N}\boldsymbol{P}_{k|k-1} &(3.37)
\end{aligned}$$

3.5.2 Non-linear smoothing

The smoothing of a non-linear process is in this project a forward filtering combined with a backwards prediction found in [Gelb et al. 1982] and [Kristensen & Madsen 2003]. The EKF is used to create the standard forward filtered estimates and the backward filter is defined in the following.

For use in the backward filtering a new time variable is introduced

$$\tau = t_N - t \quad (3.38)$$

This new time variable runs backward from the last observation time point. This leads to a changed state equation.

$$d\boldsymbol{x}_{t_N-\tau} = -\boldsymbol{f}(\boldsymbol{x}_{t_N-\tau}, \boldsymbol{u}_{t_n-\tau}, t_N-\tau, \boldsymbol{\theta})d\tau - \boldsymbol{\sigma}(\boldsymbol{u}_{t_N-\tau}, t_N-\tau, \boldsymbol{\theta})d\boldsymbol{\omega}_\tau \quad (3.39)$$

where the observation equation remains unchanged.

The actual smoothing is done by combining the results from the forward filtering and the backwards prediction. The equations used to create the smoothed estimates are (3.40) and (3.41). The line over the estimates $\overline{\boldsymbol{P}}$ and $\overline{\boldsymbol{x}}$ characterizes estimates originating from the backwards sweep.

$$\boldsymbol{P}_{k|N} = \left(\boldsymbol{P}_{k|k}^{-1} + \overline{\boldsymbol{P}}_{k|k+1}^{-1}\right)^{-1} \quad (3.40)$$

$$\hat{\boldsymbol{x}}_{k|N} = \boldsymbol{P}_{k|N}\left(\boldsymbol{P}_{k|k}^{-1}\hat{\boldsymbol{x}}_{k|k} + \overline{\boldsymbol{P}}_{k|k+1}^{-1}\hat{\overline{\boldsymbol{x}}}_{k|k+1}\right) \quad (3.41)$$

The forward sweep has the original initial conditions but the backwards sweep does not have a defined starting state or covariance. An intuitive solution would be to create a prediction from the final filtered estimate and use it as initial conditions for the backward sweep. However, this is not feasible as the final observation would be used twice. Instead, the initial conditions for the backward sweep can be found by noting that the forward filtered state covariance $\boldsymbol{P}_{N|N}$ is equal to the smoothed covariance at time $t = t_N$. This implies that the initial inverse covariance from the backwards sweep must be zero according to equation (3.40).

The initial condition for the states in the backward sweep is handled through a variable transformation.

$$\boldsymbol{s}_t = \overline{\boldsymbol{P}}_t^{-1}\hat{\overline{\boldsymbol{x}}}_t \quad (3.42)$$

The new transformed variable $\boldsymbol{s}$ is zero as initial starting point due to the covariance. The prediction and updating equations are rewritten to match the new variable.

The updating equations become

$$\boldsymbol{s}_{k|k} = \boldsymbol{s}_{k|k+1} + \boldsymbol{C}_k^T\boldsymbol{S}_k^{-1}\left(\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{u}_k, t_k, \boldsymbol{\theta}) + \boldsymbol{C}_k\hat{\boldsymbol{x}}_{k|k-1}\right) \quad (3.43)$$

$$\overline{\boldsymbol{P}}_{k|k}^{-1} = \overline{\boldsymbol{P}}_{k|k+1}^{-1} + \boldsymbol{C}_k^T\boldsymbol{S}_k^{-1}\boldsymbol{C}_k \quad (3.44)$$

The prediction equations become

$$\begin{aligned}\frac{d\boldsymbol{s}_{t_N-\tau|k}}{d\tau} &= \boldsymbol{A}_\tau^T\boldsymbol{s}_{t_N-\tau|k} - \overline{\boldsymbol{P}}_{t_N-\tau|k}^{-1}\boldsymbol{\sigma}_\tau\boldsymbol{\sigma}_\tau^T\boldsymbol{s}_{t_N-\tau|k} \quad (3.45)\\ &\quad -\overline{\boldsymbol{P}}_{t_N-\tau|k}^{-1}\left(\boldsymbol{f}(\hat{\boldsymbol{x}}_{t_N-\tau|k}, \boldsymbol{u}_{t_N-\tau}, t_N-\tau, \boldsymbol{\theta}) - \boldsymbol{A}_\tau\hat{\boldsymbol{x}}_{t_N-\tau|k}\right)\end{aligned}$$

$$\frac{d\overline{\boldsymbol{P}}_{t_N-\tau|k}^{-1}}{d\tau} = \overline{\boldsymbol{P}}_{t_N-\tau|k}^{-1}\boldsymbol{A}_\tau + \boldsymbol{A}_\tau^T\overline{\boldsymbol{P}}_{t_N-\tau|k}^{-1} - \overline{\boldsymbol{P}}_{t_N-\tau|k}^{-1}\boldsymbol{\sigma}_\tau\boldsymbol{\sigma}_\tau^T\overline{\boldsymbol{P}}_{t_N-\tau|k}^{-1} \quad (3.46)$$

Using the initial condition of $\boldsymbol{s}_{N|N+1} = \mathbf{0}$ and $\overline{\boldsymbol{P}}^{-1}_{N|N+1} = \mathbf{0}$, the backward update and prediction equations can be used to create an smoothed estimate by use of equation (3.47) and (3.48).

$$\boldsymbol{P}_{k|N} = \left(\boldsymbol{P}^{-1}_{k|k} + \overline{\boldsymbol{P}}^{-1}_{k|k+1}\right)^{-1} \tag{3.47}$$

$$\hat{\boldsymbol{x}}_{k|N} = \boldsymbol{P}_{k|N}\left(\boldsymbol{P}^{-1}_{k|k}\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{s}_{k|k+1}\right) \tag{3.48}$$

3.6 Simulation

So far methods for filtering, prediction and smoothing have been presented, which are all based on a set of observations $\mathcal{Y}$ being available. Simulation can be useful in assessing model properties without any observations available. It can be used for analysing identifiability in model components e.g. testing for the ability to separate system and measurement noise or testing for parameter identifiability.

By using simulation a new set of observations can be produced given the model. A simulated observations set is a combination of the model, given parameters, the realization of the Wiener process and the realization of the observation noise.

Simulation on SDEs are most easily performed with an Euler method. It is important to have a small step length such that the approximation to the stochastic behavior of the states is reevaluated often and the Wiener process in allowed to enter into the states more continuously. The Wiener process will hence have the possibility to disturb the approximation in every small sub-step. More advanced methods for simulation of SDEs are described in Kloeden & Platen (1992) but have not been considered.

If the model considered does not include a diffusion term a deterministic and more robust method for estimating the ODEs may be used to allow for a faster and more accurate simulation using a larger step length. To account for this situation a Runge Kutta method has been chosen. However, it is important to note that for realistic simulation of SDEs the step length must still be chosen sufficiently small for proper simulation of the Wiener process.

The sample points are named $T = \{t_1, t_2, ..., t_{n_i}\}$, and the subsampled points are named $t_1^*, t_2^*, ..., t_s^*$, where $dt = t_{i+1}^* - t_i^*$. It follows that $s = (t_{n_i} - t_1)/dt + 1$.

A 4th order Runge-Kutta method is used to find the state at each subsample point. Zero-order hold are used for the input between subsampled points. The Runge-Kutta method is shown below:

$$\boldsymbol{k}_1 = \boldsymbol{f}(\boldsymbol{x}_{t_i}, \boldsymbol{u}_{t_i}, t_i, \boldsymbol{\theta}) \tag{3.49}$$
$$\boldsymbol{k}_2 = \boldsymbol{f}(\boldsymbol{x}_{t_i} + 1/2\boldsymbol{k}_1, \boldsymbol{u}_{t_i}, t_i + 1/2\boldsymbol{d}t, \boldsymbol{\theta}) \tag{3.50}$$
$$\boldsymbol{k}_3 = \boldsymbol{f}(\boldsymbol{x}_{t_i} + 1/2\boldsymbol{k}_2, \boldsymbol{u}_{t_i}, t_i + 1/2\boldsymbol{d}t, \boldsymbol{\theta}) \tag{3.51}$$
$$\boldsymbol{k}_4 = \boldsymbol{f}(\boldsymbol{x}_{t_i} + \boldsymbol{k}_3, \boldsymbol{u}_{t_i}, t_i + \boldsymbol{d}t, \boldsymbol{\theta}) \tag{3.52}$$
$$\boldsymbol{x}_{t_{i+1}} = \boldsymbol{x}_{t_i} + \frac{1}{6}(\boldsymbol{k}_1 + \boldsymbol{k}_4) + \frac{1}{3}(\boldsymbol{k}_2 + \boldsymbol{k}_3) \tag{3.53}$$

The next step is to add the Wiener noise. The Wiener Process is a continuous-time stochastic process ω_t characterized by $[\omega_{t_2} - \omega_{t_1}] \sim N(\boldsymbol{0}, |t_2 - t_1|\boldsymbol{I})$, and can thus be approximated by a sum of gaussian increments. By setting $t_2 = t_1 + dt$ it follows that the increments at each subsample point should be sampled from $N(0, t + dt - t) = N(0, dt)$.

The updating formulas at each subsample point is thereby

$$\boldsymbol{y}_{t_i} = \boldsymbol{h}(\boldsymbol{x}_{t_i}, \boldsymbol{u}_{t_i}, t_i, \boldsymbol{\theta}) + \sqrt{\boldsymbol{S}} \cdot \boldsymbol{e}_1 \tag{3.54}$$
$$\boldsymbol{x}_{t_{i+1}} = \boldsymbol{x}_{t_i} + \frac{1}{6}(\boldsymbol{k}_1 + \boldsymbol{k}_4) + \frac{1}{3}(\boldsymbol{k}_2 + \boldsymbol{k}_3) + \boldsymbol{\sigma}(\boldsymbol{u}_{t_i}, t_i, \boldsymbol{\theta})\sqrt{dt} \cdot \boldsymbol{e}_2 \tag{3.55}$$

where $\boldsymbol{e}_1, \boldsymbol{e}_2 \sim N(\boldsymbol{0}, \boldsymbol{I})$. The initial state is given from the non-linear model setup to start the updating formulas.

Single subject summary

This chapter has shown how single subject modelling through state space models extended with SDEs can be treated with the Kalman filter. It has been presented as the basic method for dealing with single subject filtering, prediction and smoothing. Furthermore, it has been presented how parameter estimation can be performed via maximum likelihood principles.

CHAPTER 4

Population Modelling

The first stage stochastic state space model is only able to model a single individual. Most often measurements are available for several individuals in the same experiment and it would be beneficial to be able to use the measurements as one whole instead of just as a set of individual data series.

The solution to this problem is to introduce a second stage model to describe the observed variations between individuals in a given population. This type of model is called a non-linear mixed effects model and has a hierarchical model structure which splits the variation in intra- and inter-individual variation [Racine-Poon & Wakefield 1998], [Overgaard et al. 2005]. For the first stage the stochastic state space model has been chosen for modelling the individual data whereas the second stage extension includes the ability to model relationship between individuals. The intra-individual variation accounts for the random variation in individual data which is not explained by the first stage model and the inter-individual variation accounts for the differences between individual parameters for the first stage model. The hierarchical model allows for an estimation of these variance components in order to describe a population statistically.

4.1 Model definition

The second stage model for the individual parameters can be defined in a number of ways, each with different properties. In the present work it is chosen to use

$$\boldsymbol{\phi}_i = g(\boldsymbol{\theta}, \boldsymbol{Z}_i) \cdot exp(\boldsymbol{\eta}_i) \tag{4.1}$$

This formulation restricts variations in $\boldsymbol{\eta}_i$ from changing the sign of $g(\boldsymbol{\theta}, \boldsymbol{Z}_i)$ which is typically an advantages as $\boldsymbol{\phi}_i$ may be used as parameter for a variance or other sign sensitive parameters. $\boldsymbol{\theta}$ is the fixed effect parameter and $\boldsymbol{Z}_i$ is an optional covariate such as height, weight or other individually measurable covariates. $\boldsymbol{\eta}_i$ is the multivariate random effect parameter for the ith individual, which are assumed normally distributed with mean zero and covariance $\boldsymbol{\Omega}$:

$$\boldsymbol{\eta}_i \in N(\mathbf{0}, \boldsymbol{\Omega}) \tag{4.2}$$

where the dimension of $\boldsymbol{\eta}_i$ is v. By looking at Equation (4.1) it is seen that the individual fixed effect parameters $\boldsymbol{\phi}_i$ has a log-normal distribution.

4.2 Parameter estimation

In order to estimate parameters in the population model it is necessary to define a likelihood function L. The second stage distribution $p_2(\boldsymbol{\eta}_i|\boldsymbol{\Omega})$ is a multivariate Gaussian density and by combining this with the first stage distribution $p_1(\mathcal{Y}_{in_i}|\boldsymbol{\phi}_i)$ using Bayes theorem it results in the population likelihood function

$$\begin{aligned} L(\boldsymbol{\theta}|\mathcal{Y}_{Nn_i}) &\propto \prod_{i=1}^{N} \int p_1(\mathcal{Y}_{in_i}|\boldsymbol{\phi}_i) p_2(\boldsymbol{\eta}_i|\boldsymbol{\Omega}) d\boldsymbol{\eta}_i && (4.3) \\ &= \prod_{i=1}^{N} \int \exp(l_i) d\boldsymbol{\eta}_i && (4.4) \end{aligned}$$

where l_i is the a posteriori log-likelihood function for the random effects of the ith individual. As mentioned $p_2(\cdot)$ is simply a multivariate Gaussian distribution, and using this in connection with the first stage log-likelihood function in Eq. (3.28) it follows that l_i is given as

$$\boldsymbol{l}_i = -\frac{1}{2}\sum_{j=1}^{n_i}\left(\boldsymbol{\epsilon}_{ij}^T \boldsymbol{R}_{i(j|j-1)}^{-1}\boldsymbol{\epsilon}_{ij} + \log|2\pi\boldsymbol{R}_{i(j|j-1)}|\right) - \frac{1}{2}\boldsymbol{\eta}_i^T\boldsymbol{\Omega}^{-1}\boldsymbol{\eta}_i - \frac{1}{2}\log|2\pi\boldsymbol{\Omega}| \tag{4.5}$$

The population likelihood function in Eq. (4.4) can not be evaluated analytically, and therefore l_i is approximated by a second-order Taylor expansion in Eq. (4.6). The is done using the First-Order Conditional Estimation (FOCE) where the expansion is made around the value $\hat{\boldsymbol{\eta}}_i$ that minimizes l_i given $\boldsymbol{\theta}$. In this minimum $\nabla\boldsymbol{l}_{i|\hat{\eta}_i} = 0$ and it thus reduces to Eq. (4.7).

$$\begin{aligned} L(\boldsymbol{\theta}|\mathcal{Y}_{Nn_i}) &\approx \prod_{i=1}^{N}|\Delta\boldsymbol{l}_i|^{-1/2}\exp\left[\boldsymbol{l}_i - \frac{1}{2}\nabla\boldsymbol{l}_i^T\Delta\boldsymbol{l}_i^{-1}\nabla\boldsymbol{l}_i\right] && (4.6)\\ &\approx \prod_{i=1}^{N}|\Delta\boldsymbol{l}_i|^{-1/2}\exp(\boldsymbol{l}_i)\Big|_{\hat{\boldsymbol{\eta}}_i} && (4.7)\end{aligned}$$

The Hessian of the individual log-likelihood function which is needed in the evaluation of the population log-likelihood function is approximated by

$$\Delta\boldsymbol{l}_i \approx -\sum_{j=1}^{n_i}\left(\nabla\boldsymbol{\epsilon}_{ij}^T\boldsymbol{R}_{i(j|j-1)}^{-1}\nabla\boldsymbol{\epsilon}_{ij}\right) - \boldsymbol{\Omega}^{-1} \tag{4.8}$$

To avoid numerical problems with large numbers, the population log-likelihood function is derived by taking the logarithm of Eq. (4.7). This yields the main objective function which will for the remainder of this thesis be called the *Approximate Population Likelihood* (APL).

$$\log L(\boldsymbol{\theta}|\mathcal{Y}_{Nn_i}) \approx \sum_{i=1}^{N}\left[\boldsymbol{l}_i - \frac{1}{2}\log|\Delta\boldsymbol{l}_i|\right] \tag{4.9}$$

The maximum likelihood estimate of the parameters are found by

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} \quad \log L(\boldsymbol{\theta}|\mathcal{Y}_{Nn_i}) \tag{4.10}$$

4.3 Parameter uncertainty

It is as important to asses the uncertainty of the parameter estimates as it is to estimate the parameters themselves. The assessment of uncertainty reveals knowledge about which parameters estimates that should be trusted and perhaps also about which parameters that can be omitted from the model.

The maximum likelihood theory states that the covariance matrix for the estimated parameters are given by the inverse of the observed information [Thyregod & Madsen 2004]. If the parameters to be estimated are $\boldsymbol{\theta}$ the observed information is defined as

$$\boldsymbol{j}(\boldsymbol{\theta}) = -\frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \log L(\boldsymbol{\theta}) = -\boldsymbol{\nabla}^2 \log L(\boldsymbol{\theta}) \tag{4.11}$$

which is equal to the Hessian matrix of the negative log-likelihood function. If the parameters maximizing the likelihood function are called $\hat{\boldsymbol{\theta}}$ they will asymptotically have the distribution

$$\hat{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta}, \boldsymbol{j}(\hat{\boldsymbol{\theta}})^{-1}) \tag{4.12}$$

By using Eq. (4.12) it is possible to test for significance of specific parameters. A test of the assumption $H_0 : \theta_i = \theta_{i,0}$ is given by the Walds test statistics

$$z_i = \frac{\hat{\theta}_i - \theta_{i,0}}{\hat{\sigma}_{\hat{\theta}_i}} \tag{4.13}$$

where $\hat{\sigma}_{\hat{\theta}_i} = \sqrt{diag_i(\boldsymbol{j}(\hat{\boldsymbol{\theta}})^{-1})}$ and $diag_i$ means the ith diagonal element. Under H_0 the Wald test statistics is approximately $N(0, 1)$-distributed.

4.3.1 Numerical computation of Hessian

The Hessian is evaluated using a approximation, calculated as a slight extension of the formulas found in [Dennis & Schnabel 1983]. Let $f : \mathbb{R}^p \to \mathbb{R}$, an example of f being a likelihood function, then the Hessian of f, denoted $\boldsymbol{\nabla}^2 f$ can be approximated by $\boldsymbol{H}$. The ijth element of $\boldsymbol{H}$ is calculated as

$$
\begin{aligned}
k^+ &= f(\boldsymbol{x}+h\boldsymbol{e}_i+h\boldsymbol{e}_j) - f(\boldsymbol{x}+h\boldsymbol{e}_i) - f(\boldsymbol{x}+h\boldsymbol{e}_j) + f(\boldsymbol{x}) && (4.14)\\
k^- &= f(\boldsymbol{x}-h\boldsymbol{e}_i-h\boldsymbol{e}_j) - f(\boldsymbol{x}-h\boldsymbol{e}_i) - f(\boldsymbol{x}-h\boldsymbol{e}_j) + f(\boldsymbol{x}) && (4.15)\\
\boldsymbol{H}_{ij} &= \frac{k^+ + k^-}{2h^2} && (4.16)
\end{aligned}
$$

where $\boldsymbol{e}_x \in \mathbb{R}^p$ is a unit vector along the x axis. In [Dennis & Schnabel 1983] only the foreward permutation of $\boldsymbol{x}$ is used, that is $\boldsymbol{H}_{ij} = k^+/h^2$ where as this version also includes the backward permution and then takes the mean of the two. The benefit of this extra step is a more robust Hessian estimate at the cost of a longer computation time, but a good estimate of Hessian is vital for a robust uncertainty which is important in the evaluation of the model being considered.

The brute force way of calculating the Hessian is simply by doing a double loop over i and j. With p parameters in the model this would require $1 + 2 \cdot 3p^2$ function evaluations. By taking a closer look at the formulas (4.14) and (4.15) it is seen that all the function evaluations where just one parameter is changed is calculated $2p^2$ times instead of just $2p$ times (2 is for the forward and backward calculation). By taking the function evaluations where one parameter is changed out of the double loop, the required evaluation are reduced to $1 + 2p + 2p^2$. By further analysis it is seen that the function evaluations where two parameters are changed are identical across the diagonal, then the upper triangular elements can be omitted leading to just $1+2p+2\frac{1}{2}p(p+1)$ required function evaluations.

In Table 4.1 the required number of function evaluations is displayed to get a feel of the computational intensity of the Hessian evaluation. The brute force method is also shown as reference.

p	2	3	4	5	6
$1 + 2 \cdot 3p^2$	25	55	97	151	217
$1 + 2p + 2\frac{1}{2}p(p+1)$	11	19	29	41	55

Table 4.1: Number of function evaluations required.

The step length h in the Hessian approximation should be chosen sufficiently small to ensure a robust estimate.

Population modelling summary

This chapter has presented the non-linear mixed effects model with SDEs and showed the strong property of being able to model situations with a correlated residual structure in a setup with fixed and random effect population parameters. Furthermore, the approximate population likelihood is defined to enable maximum likelihood parameter estimation. The approximate population likelihood is also used to estimate the parameter uncertainties through the evaluation of the Hessian.

Chapter 5

Population Stochastic Modelling using PSM

The previous chapters have reviewed the theoretical background. This chapter describes the algorithms and the numerical implementation of the developed prototype based on the presented theory. The prototype was developed in Matlab that is a mathematical flexible language but slow compared to other more low-level languages. The goal of the project was to build a prototype that could easily be modified to test variations of the implemented modules.

Parameter estimation is a time consuming task as it requires numerous calculations of the APL. For simpler purposes as plotting the prediction for a single subject given the parameters the Kalman filter or the Extended Kalman filter is all that is needed.

The prototype was named Population Stochastic Modelling - *PSM*. The following sections will describe the different parts of the implementation.

5.1 Model specification

PSM is able to handle all models of the type non-linear mixed effects models with SDEs described in the previous chapters. The model specification is achieved through a set of m-files.[1] and it has been chosen that the complete model should be supplied. This means that no derivatives of states or observations equations are found automatically. A complete model specification consists of:

- State dynamics $\boldsymbol{f}(\cdot)$ and diffusion term $\boldsymbol{\sigma}$
- Output function $\boldsymbol{h}(\cdot)$ and uncertainty $\boldsymbol{S}$
- Derivatives of state $d\boldsymbol{f}/dt$ and output $d\boldsymbol{h}/dt$
- Initial state $\boldsymbol{x}_0$
- Second stage parameter model $\boldsymbol{g}(\cdot)$ and variance $\boldsymbol{\Omega}$

Additionally, for the parameter estimation an initial guess for the parameters must be provided along with boundaries $\boldsymbol{\theta}_{min} < \boldsymbol{\theta} < \boldsymbol{\theta}_{max}$. Finally, a list of files names for the Matlab function defining the complete model is required. In the linear case some of the functions are simply matrices which simplifies the model specification.

5.2 Extended Kalman filter implementation

The Kalman filter is reviewed in Chapter 3.3. It is an exact minimal variance filter for the linear case whereas the Extended Kalman filter is an approximate filter for the non-linear case.

The linear Kalman filter has to be implemented with several special cases to overcome e.g. singular state matrix $\boldsymbol{A}$ and input $\boldsymbol{u}$. PSM has been implemented to account for both but not at the same time. The EKF should be used in cases where this need arises. This can be done without any loss in accuracy unfortunately at the price of longer computation times.

In the non-linear case the main part of the filter can be easily implemented using standard linear algebra calculations. More effort has to be put into the prediction part. In the non-linear case the derivative of the states can be state

[1] Matlab functions and script files.

dependent. For an accurate state prediction it is important that the linearization of $\boldsymbol{f}(\cdot)$ is reevaluated during the prediction. To obtain the desired level of accuracy in the predictions a frequent subsampling combined with the Euler method could be a solution.

An alternative and the chosen solution is to use an existing ODE-solver. The advantage of using an existing ODE-solver is a higher accuracy in the predictions. This is achieved by using the work put into creating robust algorithms for solving ODEs. *ode15s*[2] is a solver for stiff systems but can also handle non-stiff differential equations. It has variable step length which gives desired accuracy in fewer steps. By choosing a standard solver as a module in the implementation, it enables easy interchange of solvers without the need for a complete code update. An interchange of solver could be interesting in later analysis.

The prediction differential equations (3.8) and (3.9) can be seen to be coupled through $\boldsymbol{A}$. To account for this coupling, the two prediction equations have been collected into one system with a combined input vector Z which stores both the states and the covariance matrix. The symmetry in the covariance matrix is exploited so only the upper part is transferred.

$$Z = \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} = \begin{pmatrix} \hat{\boldsymbol{x}}_{t|k} \\ U(\boldsymbol{P}_{t|k}) \end{pmatrix} \tag{5.1}$$

where $U()$ is the upper matrix.

The conversion to Z is then used in conjunction with ode15s and the output is converted back into states and covariance.

The implemented Kalman filter and Extended Kalman filter returns the negative log-likelihood Eq. (3.28). This returned neg. log-likelihood value is used in parameter estimation where it should be minimized using a minimizer. The next section describes the minimizers and their work.

5.3 Minimizers

The main computational effort in this project is spent on minimizing functions. The problem of minimization is encountered in many areas and a wide range of algorithms exists to determine the optimal parameters. The goal is to find

[2] *odes15s* is a built-in Matlab ODE-solver for stiff ODE systems.

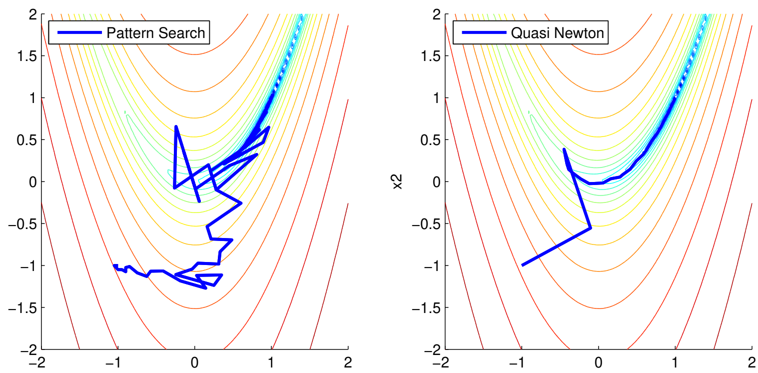


Figure 5.1: Minimization paths for the 2 different minimizers.

the set of parameters that corresponds to the global minima of the objective function.

In our project two methods have been used. These are a Quasi-Newton method based on a BFGS scheme and a Nelder-Mead pattern search.

The pattern search method[3] explorers the local neighbourhood of the parameter space by creating small changes in the parameters. The method does not require the calculation of a gradient and is ideal to search in a small area around the initial parameters.

The BFGS method utilizes the gradient in the current parameters to move forward to a new set of parameters. *fminunc*[4] was used in the beginning of the project but at a later stage *ucminf*[5] was used. The shift was done due to lack of transparency of the built-in function. *ucminf* is an implementation of the BFGS algorithm [Frandsen et al. 2004]. The gradients of the objective function with relation to the parameters are calculated numerically by either a forward or central gradient. A property of the Quasi-Newton algorithm is that it ensures quadratic convergence near optimum [Frandsen et al. 2004, p.42].

In Figure 5.1 the two minimizers have been used on a modified Rosenbrock[6] function. It is clear that the pattern search method searches through the lo-

[3] *fminsearch* is the built-in pattern search method in Matlab
[4] *fminunc* Matlabs built-in gradient minimizer
[5] *ucminf* is a minimizer developed at IMM - www.imm.dtu.dk/ hbn/Software/
[6] Rosenbrock is a standard example used for testing minimizations algorithms.

cal neighborhood while Quasi-Newton utilizes the gradient as the direction to move. It should be noticed that the Quasi-Newton method uses extra function evaluations to calculate the gradient in each step. The choice of minimizer is a compromise between the overhead in calculating gradient versus the same number of function evaluations in the local neighborhood.

Both methods requires a set of options for the criteria for termination. The used options are tolerance on parameter changes, tolerance on objective function changes, tolerances on gradient or maximum number of function evaluations. These options are set accordingly to the specific minimization.

The minimizers are used both for parameter estimation for a single subject and population case. The population likelihood minimization is described in the following section.

5.4 Minimization of APL

Population modelling is described in Chapter 4. The modelling now deals with two different kind of parameters - population and individual parameters. Population parameters $\boldsymbol{\theta}$ are identical for all subjects whereas the individual parameters $\boldsymbol{\phi}$ are unique for each subject.

The algorithm to calculate the approximate population likelihood involves the a posteriori individual log-likelihood and its gradient. The FOCE approximation requires that the individual log-likelihoods are minimized for a given set of population parameters. Hence for each set of population parameters a minimization must be performed on each subject.

In Figure 5.2 the overview of the minimization of the APL is shown. In the overview the Kalman filter (KF) is used as an example but the scheme is identical for the EKF. The goal of the population minimization is to find the optimal population parameters. The workload of the population minimization is highly dependent on the number of parameters to be estimated but also on the number of individual minimizations N.

The workload for the individual minimization is heavily dependent on the number of random effects $\boldsymbol{\eta}$. The individual minimization is performed with the Quasi-Newton method.

The extent of a minimization of the APL is also seen in Figure 5.2. With increasing model complexity and larger parameters spaces to search, this minimization

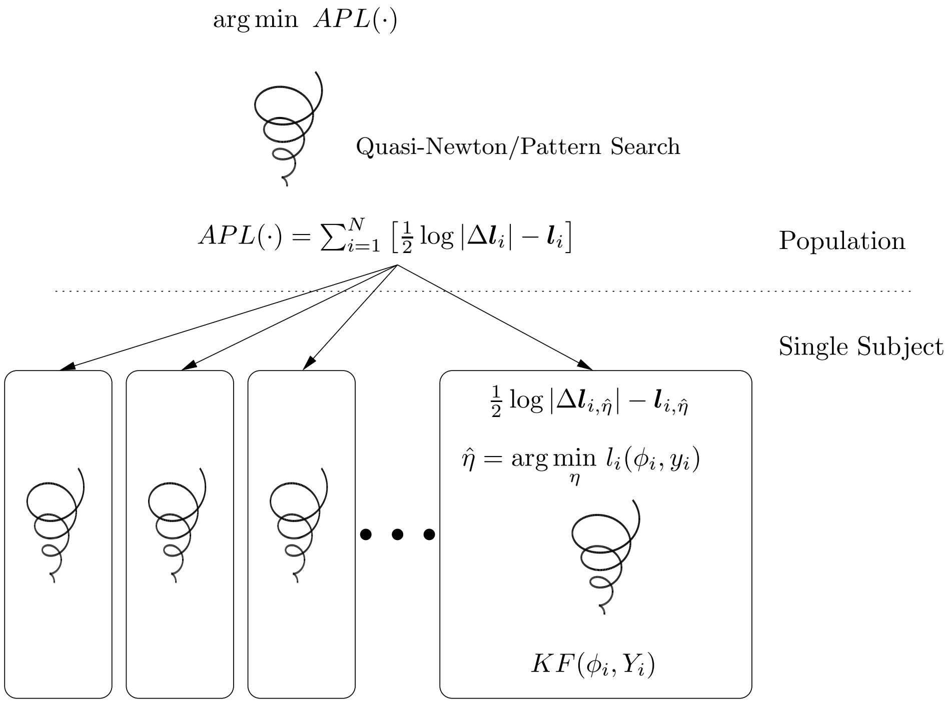


Figure 5.2: Schematic overview of PSM.

can become a massive task. The minimization task is further reviewed in the following sections.

5.4.1 Population Minimization

The minimization of the APL is the main minimization in the parameter estimation for a population. The minimization can be performed with either a Quasi-Newton or a pattern search method.

The gradient used in the Quasi-Newton method is numerically computed with a relative step size of 10^{-2} of the parameter. The gradient is a forward gradient as the objective function can be a substantial computational task. The parameter step size in the gradient is chosen relatively large to capture the global trend as small step sizes can result in unstable gradients due to noise. The noise levels will be further discussed in Section 7.2.1.

To overcome the noise disturbing the gradient near minimum, the pattern search method can be used for a final minimization. In general it is advisable to generate profiles of the objective function (APL) for each parameter and thereby confirming if a minimum was found.

The APL calculation consists of a sum of contributions from each individual. These contributions are the result of an individual minimization where the optimal $\hat{\boldsymbol{\eta}}$ is found. These individual minimizations are the subject of the following section.

5.4.2 Single Subject Minimization

The single subject minimization finds the optimal $\hat{\boldsymbol{\eta}}$s for each subject given a set of population parameters. The dimensionality of the parameter space for the minimization is given by the number of individual parameters.

The minimization is performed using Quasi-Newton which has been found to perform well on minimizing the individual log-likelihood in examples tested in this thesis. The initial guess is always $\boldsymbol{\eta} = \mathbf{0}$ which is expected to be in the proximity of the global minimum when the population parameters are close to optimum. On the other hand, when the population parameters are far from optimum the individual minimizations will be more demanding.

The gradient of the individual log-likelihood used in the Quasi-Newton minimizations is a central gradient based on a additive change in $\boldsymbol{\eta}$ by 10^{-4}. An additive change is chosen since $\boldsymbol{\eta}$s are assumed to be normally distributed with mean zero so a relative change would be less robust.

The approximated Hessian for the individual log-likelihood function is calculated based on the gradient of the prediction residuals with respect to $\boldsymbol{\eta}$. The gradient is a central gradient with an additive step length of 10^{-5}.

The minimization termination criteria for the individual minimization are defined by the infinity norm of the gradient and the relative change in step length given as

$$\begin{aligned} \|d(l_i)/d\boldsymbol{\eta}\|_\infty &< c_1 && (5.2)\\ \|\nabla\boldsymbol{\eta}\|_2 &< c_2(c_2 + \|\boldsymbol{\eta}\|_2) && (5.3) \end{aligned}$$

where both c_1 and c_2 are set to 10^{-4} to ensure an accurate estimate of $\hat{\boldsymbol{\eta}}$ is obtained. These termination criteria are an important factor in the compromise between speed and accuracy for the inner optimization.

5.5 Parameter uncertainty

The assessment of the parameter uncertainty reviewed in Section 4.3 is evaluated in the minimum found by the optimal parameters. A description of the equations used in the numerical approximations for the Hessian is found in Section 4.3.1.

The step length h used in the numerical Hessian calculation is a relative to the parameter and unless other is specified it is set to 10^{-3}. The step length were tested for a small range of values from 10^{-2} to 10^{-4} and were found to provide robust estimates.

The implemented function calculating the parameter covariance matrix also returns the correlation matrix for the parameters. The covariance matrix shows the uncertainty of the specific parameter whereas the correlation matrix shows the interaction among parameters. A high correlation indicates that the model specification makes it hard to separate the effect of each parameter in the APL function.

5.6 Parallel computing

The scale and magnitude of the computational effort in estimating parameters are extensive. The combination of an optimization of several optimizations generates a large need for computational power. Adding to this workload the program language Matlab results in longer execution times compared to other languages. An implementation in another language would increase the speed significantly but it would still be a large computational task.

The preferred solution and what seems to be the trend is to use parallel computing. There are several areas in the PSM prototype where parallel computing would be beneficial. It has been chosen to implement parallelization in the prototype in the split from overall minimization to minimization pr. individual. The parallelization results in N separate calculations which is seen clearly by all the arrows in Figure 5.2. Furthermore, but not implemented it would be advantageous to introduce parallel execution of the numerically calculated gradients.

Matlab does not have the option of parallel computing by default. MatMPI[7] enables parallel computing in Matlab by creating a set of scripts that is executed in separate processes. MatMPI uses message parsing but it was found faster to parse all data and parameters through files. The individual calculation extracts its unique part of data by using its identifier number. The individual log-likelihood result is parsed back into the leader thread by proper message parsing to avoid deadlocks or race conditions. A shared memory environment is required as the message parsing is implemented through shared files.

MatMPI works by issuing commands in blocks. This means that all working processors are started at the same time and next block of data is not started until all have finished. Some consideration should be put into matching the number of CPUs with the data or simply just choosing the maximum number of CPUs available.

In Figure 5.3 a simulated data set consisting of 20 series of objects in free fall was generated. A single APL was calculated for a different number of working processors. The server handled a large number of jobs simultaneously for other users which have influenced the computation times. It can be seen that the computation time is reduced to around a fifth of the original. In this example it appears that the overhead is relatively large, however for more computational intensive models the benefit of adding more CPUs may be substantial.

[7]MatMPI is a package to Matlab - developed at MIT Lincoln Laboratory by Dr. Jeremy Kepner. - http://www.ll.mit.edu/MatlabMPI/

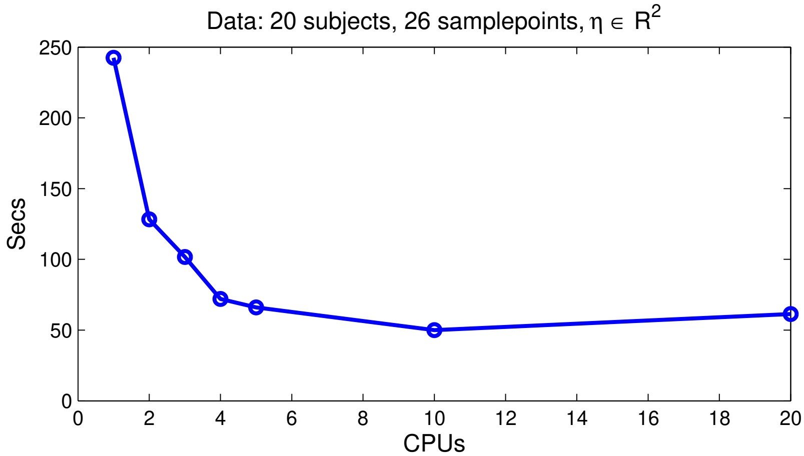


Figure 5.3: Computation Times using Parallel Computing

PSM summary

A prototype has been developed in Matlab implementing a non-linear mixed-effect model with SDEs. The numerical implementation is based on an ODE-solver, minimizers and own code.

Functions for estimating the parameter uncertainty have been implemented. The Hessian calculation has been analysed and optimized to avoid unnecessary calculations of the costly APL function.

The computational work in solving parameter estimation in the prototype is substantial but parallelization has been shown to improve execution time. A more thorough validation of the implemented prototype is performed in the next chapter.

Chapter 6

Validation of PSM

The main purpose of the validation of PSM is to ensure that the results produced can be trusted. This will be done by comparing with existing PK/PD software programs. However, no single program can handle non-linear mixed effects models with SDEs and for this reason the prototype is validated against two different programs. The first is CTSM (Continuous Time Stochastic Modelling) is able to handle stochastic state space models for a single subject. The second program NONMEM is able to handle non-linear mixed effects model but only in ODE context.

The implemented prototype consists of several functions which together are able to handle both the stochastic state space model and the non-linear mixed effect part. Many of the functions are interlinked making it possible to test several functions at the same time as it is unlikely that erroneous subfunctions will produce a correct result in the overall function.

The validation will be done in two steps. Firstly, the Kalman filter will be verified by comparing results with CTSM using a model based on SDEs on a single individual. Secondly, the APL function will be compared with NONMEM. This can only be done with ODEs in NONMEM, and thus the validation is based upon that the SDE extension is validated against CTSM.

CTSM	259.028450
PSM EKF	259.027492
PSM KF	259.027543

Table 6.1: Comparison of CTSM and PSM log-likelihood functions.

It is also needed to validate that the parameter estimation yields correct population parameters as well as individual parameters. Finally, the Kalman smoother will be validated.

All of the above must further more be carried out for both the linear and non-linear versions of the implementation.

6.1 Kalman filter

The Kalman filter is the basis for the implemented population model since the likelihood function is based on the one-step predictions and variances. The Kalman filter in PSM is validated against CTSM by comparing the values of the likelihood functions for one individual, see Eq. (3.28). This is done by comparing with a two-compartment C-Peptide model which will be described in further detail in Section 7.2. The data are taken from subject 1.

The model parameters to be estimated are the initial states, C_1^0, C_2^0, ISR^0, measurement noise S, and Wiener magnitude for the secretion rate σ_{ISR}. Normally, the initial state is constrained to steady state giving just one parameter to estimate, but it is not possible to set this constraint in CTSM. The model is estimated using CTSM, and the resulting estimates are $\hat{\boldsymbol{\theta}} = (C_1, C_2, ISR, S, \sigma_{ISR})$ $= (906.62, -8.7908, 104.58, 13763, 7.1796)$. It is noted that C_2 is estimated to a negative initial concentration, which is of course not physically possible. However, the estimated standard deviation in CTSM is 29.8 and it can therefore not be assumed significantly different from 0.

The estimated parameter values from CTSM are used with both the non-linear and linear Kalman filters in PSM and the resulting likelihood values are shown in Table 6.1. It is seen that the values deviate but they are still acceptably close to being equal. The deviation can be caused by numerical differences in implementation or that the parameters from CTSM were only available with 5 significant digits.

As mentioned the likelihood value is only based on the predictions from the Kalman filter. To verify the filtered values from the Kalman filter, it is compared to CTSM by plotting the secretion rate (3rd state) and by visual inspection it is seen that both the non-linear and linear Kalman filters in PSM yields filtered estimates approximately equal to CTSM.

6.1.1 Kalman smoother

The Kalman smoother is validated against CTSM using the same example as in the previous section. Smoothed estimates are generated in CTSM and in Figure 6.1 the smoothed ISR-state is compared to the non-linear PSM smoother estimates. By looking at the figure it is seen that they are practically identical in estimate as well as standard deviation. CTSM only returns values for observation time points whereas PSM has inserted subpoints between observations. The comparison should only be based on the value at observation time points.

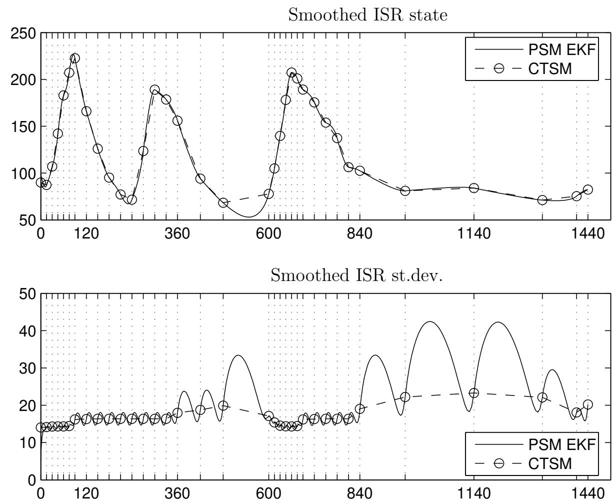


Figure 6.1: Comparison of EKF smoothed secretion rate values.

The linear version of the Kalman Smoother based on the Bryson-Frazier (B-F) formulas is also tested against CTSM and the output is shown in Figure 6.2. At a first glance this also corresponds with the CTSM smoother. However, there seems to be some small numerical problems with the method just before $T = 600s$ for this model, see Figure 6.3. It is expected that the smoothed estimate of the state and standard deviation are continuous, and not exhibit the discontinuous behavior seen in the figure. However, right after $T = 600$ the

B-F method returns to the CTSM estimates. A thorough investigation of the problem should be carried out, but it has not been done as it is considered of minor interest.

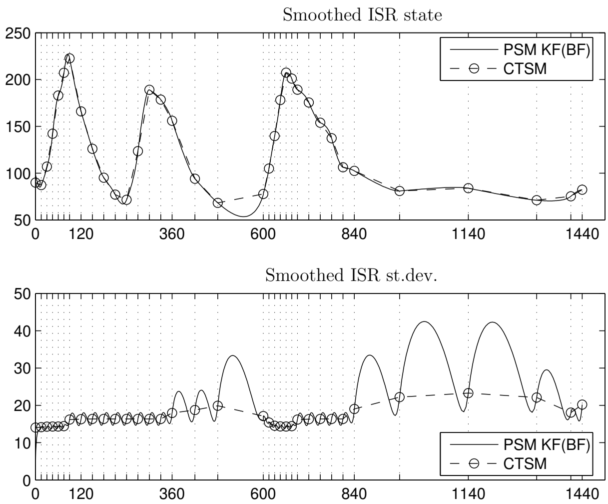


Figure 6.2: Comparison of KF smoothed secretion rate values.

The Kalman filter and smoother have been successfully validated with CTSM as reference. This is considered sufficient for the present need and in line with focus of the thesis.

6.2 Population likelihood function

To validate the approximate population likelihood function, a one-compartment ODE model for insulin data is set up and used for simulation to create validation data. The model only includes elimination, which is modelled as a constant $K = CL_i/V_i$ multiplied by the concentration of insulin in the compartment. The model is initiated at $t = 0$ with a bolus injection of 10pmol insulin, giving an initial concentration of $10/V_i$ in the compartment.

If the measurement equation for the concentration is modelled with additive white noise, then the model can be written as

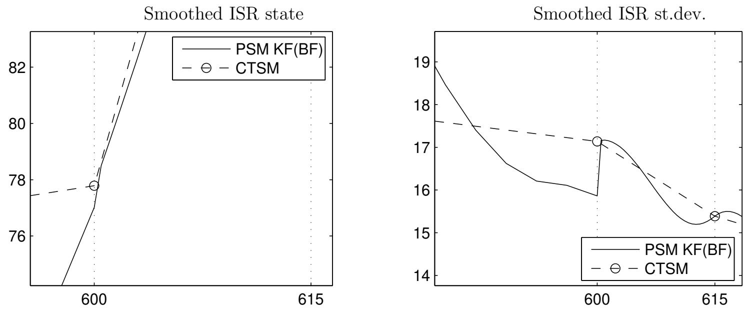


Figure 6.3: Close up of KF smoothed secretion rate values.

$$
\begin{aligned}
dC &= -KCdt && (6.1)\\
Y_{ij} &= C + e_{ij} && (6.2)
\end{aligned}
$$

where C is the concentration of insulin in the compartment. Given the state equation it is seen that the initial concentration will decay toward zero exponentially. Since the noise is additive this will result in negative measured concentrations in a simulation, which is obviously meaningless. One way to avoid this, is to stop the simulation before the concentration becomes close too zero, but the preferred way is to introduce a log-transformation of the concentration in the compartment

$$B = \log C \tag{6.3}$$

which results in the following ODE state space model

$$
\begin{aligned}
dB &= \frac{1}{C}dC = -Kdt && (6.4)\\
\log Y &= B + e_{ij} && (6.5)
\end{aligned}
$$

This formulation of the model results in multiplicative measurement noise since $Y = \exp(B + e_{ij}) = C \cdot \exp(e_{ij})$ but is otherwise the same. The advantage is that it will not lead to negative measured concentrations and the noise is proportional with the concentration which is not unlikely. Figure 6.4 shows an example of one simulation with the log-transformed model.

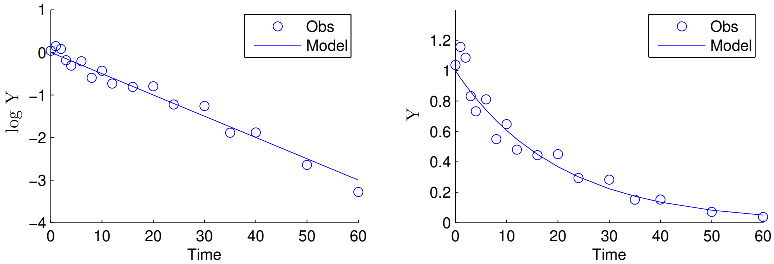


Figure 6.4: Example of a simulation with the log-transformed model. The two figures illustrate the same data on a linear and logarithmic scale.

NONMEM objective function

Individuals	2	4	10	20
1 ind. par.	-21.581	-28.298	-55.527	-331.700
2 ind. par.	-52.494	-104.863	-239.831	-519.315

PSM negative log-likelihood function

Individuals	2	4	10	20
1 ind. par.	11.264	29.959	82.510	54.713
2 ind. par.	-2.355	-4.647	-0.454	-20.733

Table 6.2: Comparison of NONMEM and PSM log-likelihood functions.

The data used for the comparison with NONMEM is simulated using two individual parameters for CL and V. A total of four validation data sets are created with 2, 4, 10 and 20 subjects. The two individual parameters are chosen in the range from approx. $[-1.5\ 1.5]$. The data is estimated using the true model as well as a reduced model where only CL is modelled with an individual parameter.

The estimation is done using NONMEM, and estimated parameters are then evaluated in PSM in order to compare the APL value. The population parameters for the true model are CL, V, Ω_{CL}, Ω_V and S. In the reduced model the individual parameter on V and its corresponding variance Ω_V is removed. Table 6.2 shows the resulting likelihood values.

The values from Table 6.2 are also plotted in Figure 6.5. First of all it is quite obvious that values are not the same. For 2, 4 and 10 individuals it looks like there is a linear dependence indicating a constant difference in the likelihood function per individual. The extension to 20 individuals yields that the linear relationship does not hold.

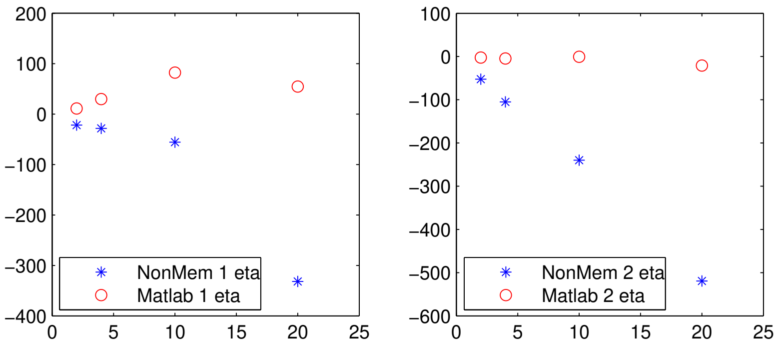


Figure 6.5: Plot of NONMEM and PSM log-likelihood functions.

In terms of parameter estimation, it is not a problem that the likelihood functions differ in values as long as they have optimum for the same parameters. To test this, the APL profiles are plotted around the optimal NONMEM parameters which are seen in Figure 6.6 as the center point.

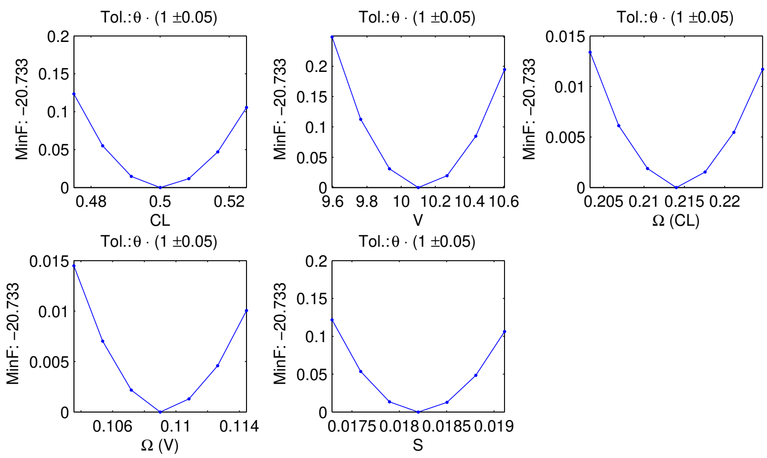


Figure 6.6: PSM likelihood function around NONMEM optimum.

Figure 6.6 shows the PSM APL plotted around the NONMEM optimum, which is the center point in all plots. The width of the interval for each parameter is the optimum value $\pm 5\%$. The data contains 20 individuals and the model has two individual parameters. By plotting the APL profiles it is seen that the NONMEM optimal parameters also minimizes the PSM likelihood function. The same is done for the remaining data sets, and in all cases it is found that the NONMEM optimum also minimizes the PSM likelihood function. The conclusion is that the implemented likelihood function yields the same parameter estimates as NONMEM.

As a further confirmation of the equivalence of the two population likelihood functions it should hold that also the individual likelihood function has the same optima. The individual likelihood function is a function of $\boldsymbol{\eta}_i$ as stated in Eq. (4.1). In Figure 6.7 the optimal $\boldsymbol{\eta}$ values from PSM and NONMEM are plotted against each other. The data with 20 individuals has been used with the model with two individual parameters. By a visual inspection of the plot it is found that the estimated parameters are very similar.

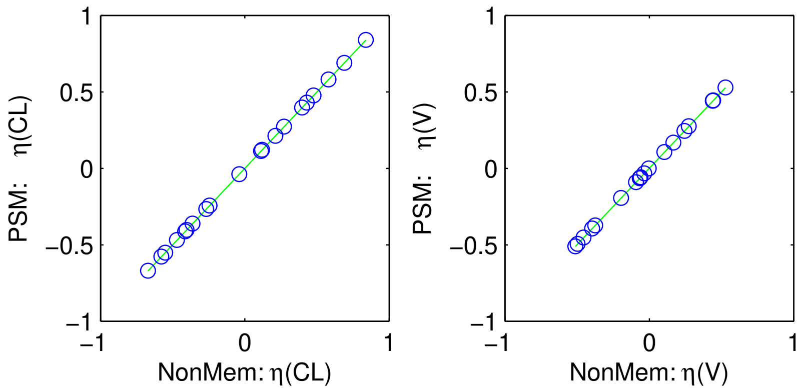


Figure 6.7: NONMEM vs. PSM individual η-values.

6.2.1 Likelihood ratio test

Even though the two log-likelihood functions differ in absolute values, they should still yield the same difference to a model of lower dimension. This is used in the likelihood ratio test to test for significance of specific parameters in a given model [Bickel & Doksum 1976].

If two models with m and n parameters are considered where $m > n$, it holds that the difference of $-2\log L$ is $\chi^2(m-n)$ distributed if the model based on n parameters is sufficient.

NONMEM objective function difference

Individuals	2	4	10	20
	30.9130	76.5650	184.3040	187.6150

2× PSM negative log-likelihood function difference

Individuals	2	4	10	20
	27.2380	69.2120	165.9280	150.8920

Table 6.3: Comparison of NONMEM and PSM log-likelihood difference.

In the example used previously in this section the large model is based on 5 population parameters $(CL, V, \Omega_{CL}, \Omega_V, S)$ and the reduced is based on 4 (CL, V, Ω_{CL}, S). Thus the difference in $-2\log L$ can be tested against $\chi^2(1)$. NONMEM's objective function is based on $-2\log L$ where as PSM's is based on $-\log L$ and this difference should thus be multiplied by 2. The comparison based on the results in Table 6.2 is shown in Table 6.3.

The log-likelihood differences are similar but in no way equal. The reason is most likely implementation issues concerning the approximation of the likelihood function. It is observed that all the NONMEM differences are larger resulting in the reduced model being rejected more often.

The 99.9% percentile of $\chi^2(1)$ is 10.8276, and the reduced model can thus be strongly rejected in all cases. This is also expected since the data is simulated using the large model.

6.3 Simulation

The simulation part of PSM is validated using the log-transformed one-compartment insulin model from Section 6.2 only this time there is also insulin added to the system. The insulin secretion is modelled as a Wiener process. The system is defined as

$$dB = -\frac{CL}{V}dt + \sigma d\omega \tag{6.6}$$

$$\log Y = B + e_{ij} \tag{6.7}$$

Figure 6.8 shows the result of a simulation with the model for $t = 0, 2, 4, ..., 300$ giving 151 measurements. The parameters used are $(CL, V, \sigma, S) = (0.5, 10, 0.2, 0.3)$.

The simulation program outputs the simulated states and measurements. Since the ODE solution is known (the left most figure) this can be subtracted from the states to reveal the Wiener process. The measurement noise process is found as the states subtracted from the measurements, see Eq. 6.7.

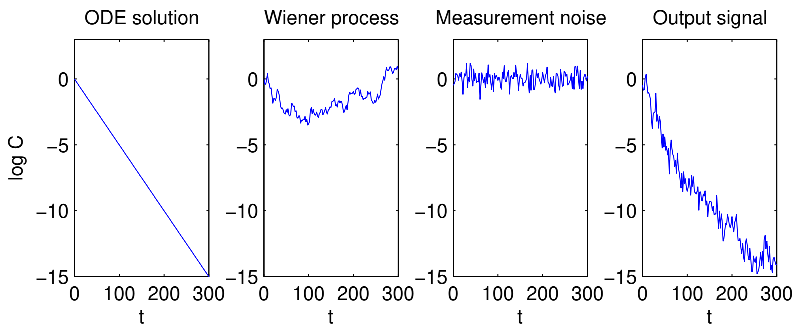


Figure 6.8: Illustration of a simulation.

To validate the implementation it is necessary to verify that the measurement noise e_{ij} comes from an $N(0, S)$ distribution and that the Wiener noise has the property $[\omega(u) - \omega(t)] \sim N(0, u - t)$. Since all increments are multiplied by σ the resulting distribution is $N(0, \sigma^2[u - t])$ in this 1-dimensional problem.

The moments of the measurement noise is estimated as $\hat{\mu}_e = 0.0064$ and $\hat{s}_e^2 = 0.2924$. A t-test for $\mu_e = 0$ yields $t = (\hat{\mu}_e - 0)/\hat{s}_e = 0.0119$. This gives $p = 0.5047$ and can thus not be rejected. A χ^2-test for $\hat{s}_e^2 = .3$ yields $\chi^2 = (151 - 1)\hat{s}_e^2/.3 = 146.1946$. This gives $p = 0.4274$ and can thus neither be rejected.

The Wiener noise is tested by looking at $W = \omega_t - \omega_{t+10}$ which should have a $N(0, 0.2^2 \cdot 10)$ distribution. A t-test for the mean yields $p = 0.4810$ and a χ^2-test for the variance yields $p = 0.5514$.

In short it has been shown that it can not be rejected that the measurement and Wiener noise is correct.

6.3.1 Validation summary

The validation chapter has shown that the Kalman filter and smoother have output similar to corresponding functions found in CTSM. Small deviations are seen but probably caused by difference in numerical implementation, ODE-solver or specified parameters.

The APL function is compared to NONMEM in an ODE case. They differ but the found optimum for parameter estimation is shown to be identical for the two programs. The difference is also examined for effects in a likelihood ratio test. The conclusion is that a difference exists and in short that NONMEM rejects parameters less frequent. It has been not been possible to find documentation for the APL function for NONMEM and various attempts to explain the difference in values have not been successful.

Finally a validation showed that properties specified in simulation could also be found in simulated data.

Chapter 7

Insulin Secretion

This chapter will focus on estimating properties of the insulin process using the PSM setup based on data from type 2 diabetic patients. The methods however are generally applicable and can also be used on healthy individuals.

The first part focuses on estimating insulin secretion rate (ISR) using deconvolution, whereas the second part takes it a step further and suggests a model for ISR. In the last part of the chapter it is shown how the PSM setup can be used with multi-variate data to estimate ISR and the insulin extraction by the liver.

7.1 Data

The data originates from a double-blind[1], placebo-controlled, randomized crossover[2] study with a duration of 24 hours starting at 8 a.m. in the morning. Thirteen patients (5 women and 8 men) with type 2 diabetes were examined. Their age given as mean ± 1 standard deviation was 56.4 ± 9.2 years, BMI was 31.2 ±

[1] A study comparing two or more treatments where neither the subject nor the person giving the treatment knows who has received which treatment.

[2] A study in which subjects are randomly assigned to different treatments and then switched at the halfway point in the treatment.

3.6 kg/m^2 and the duration of diabetes was 3.0 $\pm$ 2.6 years (range 5 months to 8 years).

For each patient a number of variables are measured. C-peptide and insulin measurements will be used for analysis in this thesis. Also for this thesis only the placebo data is used since the effect of the drug given in the study is not of interest. One of the patients was discarded since the measurement times were delayed compared to the rest. The data used in this thesis thus consists of 24-hour C-peptide and insulin profiles for 12 individuals.

The profiles were sampled 35 times during the 24 hours at varying time interval mainly concentrated after meal times. A total of 3 standard meals were given at 8 a.m., noon and 6 p.m., each to be finished within 20 minutes. These times corresponds to 0, 240 and 600 minutes after the study was initiated, see Figure 7.1. The meal times are naturally important, since food intake is a main factor of controlling insulin secretion.

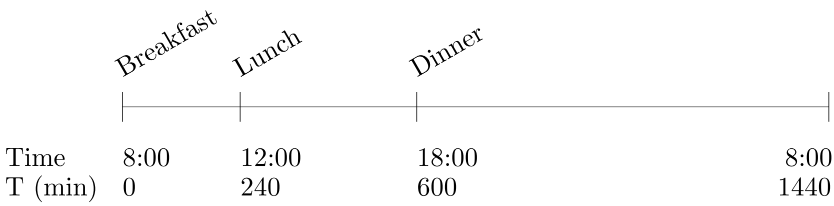


Figure 7.1: Meal times during 24H study period.

In Figure 7.2 the individual profiles can be seen for the two measured variables. It is clearly visible that insulin production increases rapidly right after meal intake. Overall there is generally a high resemblance between the profiles but the scale and level of the *3 hill* profile is different.

7.2 Deconvolution of ISR

The first model approach to estimate the insulin secretion rate (ISR) by deconvolution is performed with a standard two-compartment model for C-peptide measurements, see Figure 7.3. The model is created using the stochastic differential equations for the C-peptide kinetics with the ISR modelled as a Wiener process (also loosely denoted a random walk) with magnitude σ_{ISR}. The magnitude parameter influences the Kalman gain for the Wiener process, and a larger magnitude will thus lead to a more fluctuating Wiener process with larger increments and vice versa.

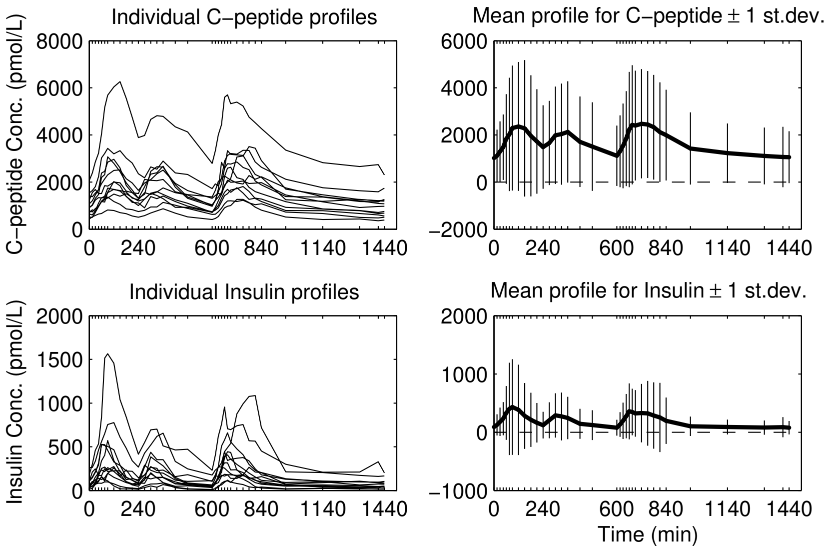


Figure 7.2: C-peptide and insulin data for 12 individuals.

The setup requires three states, namely a central compartment state C_1 containing the measured C-peptide concentration, a peripheral compartment state C_2 and a state ISR for the estimation of ISR. The C-peptide kinetic parameters k_1, k_2, k_e are set equal to the Van Cauter estimates [Cauter et al. 1992]. In the model there will only be a magnitude parameter for the Wiener process on the ISR but not on C_1 and C_2. This would otherwise lead to a model where C-peptide could randomly appear and disappear and this is a violation of the law of mass conservation.

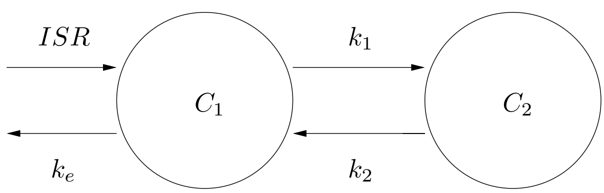


Figure 7.3: Two-compartment model used for estimation of ISR.

The C-peptide measurement error is assumed to be additive gaussian white noise with variance S. The model states are constrained to steady state at $t = 0$ given an initial estimated concentration in C_1 as shown in Eq. (7.1).

$$\boldsymbol{x}_0 = \begin{bmatrix} C_1^0 \\ C_2^0 \\ ISR^0 \end{bmatrix} = \begin{bmatrix} C_1^0 \\ \frac{k1}{k2} C_1^0 \\ k_e C_1^0 \end{bmatrix} \tag{7.1}$$

C_1^0 is estimated individually (denoted C_i) resulting in one random effect η and a corresponding population parameter for the variance Ω_{C_1}. The population parameters to be estimated are C_1, S, σ_{ISR} and Ω_{C_1}, which is illustrated in the model layout shown in the box below.

Standard two-compartment C-peptide Model

Description: A two-compartment model with individualized starting point in the initial conditions. ISR is modelled as a random walk.

State variables

$$\boldsymbol{x} = [\, C_1 \quad C_2 \quad ISR \,]^T \tag{7.2}$$

Initial Conditions:

$$C_i = C_1^0 \exp \eta \tag{7.3}$$

$$\boldsymbol{x}_0 = \begin{bmatrix} C_i & \frac{k1}{k2} C_i & k_e C_i \end{bmatrix} \tag{7.4}$$

Model :

$$d\boldsymbol{x} = \begin{bmatrix} -(k_1 + k_e) & k_2 & 1 \\ k_1 & -k_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}\, dt + diag \begin{bmatrix} 0 \\ 0 \\ \sigma_{ISR} \end{bmatrix} d\boldsymbol{\omega} \tag{7.5}$$

Output :

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \boldsymbol{x} + \epsilon, \qquad \text{where} \quad \epsilon \in N(0, S) \tag{7.6}$$

Parameters to estimate

$$\boldsymbol{\theta} = [\, C_1^0 \quad \sigma_{ISR} \quad S] \tag{7.7}$$

The model is only to be used for estimating ISR after a clinical trial has been performed. It cannot be used for simulation studies since it follows from the model setup that ISR will be estimated as strictly positive random walk. Any simulation will yield both positive and negative values for the ISR random walk leading to a physically invalid realization of the process.

It should be noted that although this way of estimating ISR using stochastic differential equations is loosely denoted deconvolution, it is in fact not strictly speaking a deconvolution. A true deconvolution using the model shown in Figure 7.3 will estimate ISR at each measurement to be equal to the rate giving the 'missing' amount in C_1. With the SSSM approach the measurement noise on C_1 is taken into account by the Kalman gain in a minimal variance way and this leads to a more smooth estimate of ISR where the effect of noise is reduced.

The problem of fitting to noise in data when performing deconvolution is well known and has been addressed by existing software. An example is WinNonlin [Pharsight 2004], which is a standard software solution used for deconvolution of PK/PD data. The program addresses the problem by introducing a smoothness factor and as a consequence it is simply left up to personal choice and preference of the user to specify the level of smoothing. Another better solution can be found using WinStoDec presented by Sparacino et al. (2002) which is based on stochastic deconvolution and can be used for linear time-invariant systems. It has been shown by Kristensen et al. (2004) that the stochastic deconvolution approach is equivalent to the SDE approach presented here, which is further more by nature also able handle to non-linear time-variant systems.

7.2.1 Optimization of likelihood function and noise

The model is estimated by an optimization of the population likelihood function. This has proven to be a non-trivial task because the likelihood function is not entirely continuous but is distorted by a level of noise. This gives rise to significant problems for the optimization. A gradient based method such as Quasi-Newton is the most sensitive, since it is likely to fail on the 'line search' part of the algorithm. This happens when the chosen search direction does not yield a lower likelihood function value and this is usually the case when the found solution is close to the optimum. This is unfortunate since this problem constrains the Quasi-Newton method from achieving quadratic convergence near the optimum and it is instead necessary to use the pattern search method to achieve an optimal solution.

It is of interest to estimate the extent of this noise. Figure 7.4 shows likelihood profiles for a very narrow relative range (±0.0005%) around the optimal parameters. For this range the likelihood function is almost completely flat and it is thus possible to see any significant noise present. The dots in the figure represents evaluated values and the smooth curve is a fitted 2nd degree polynomial.

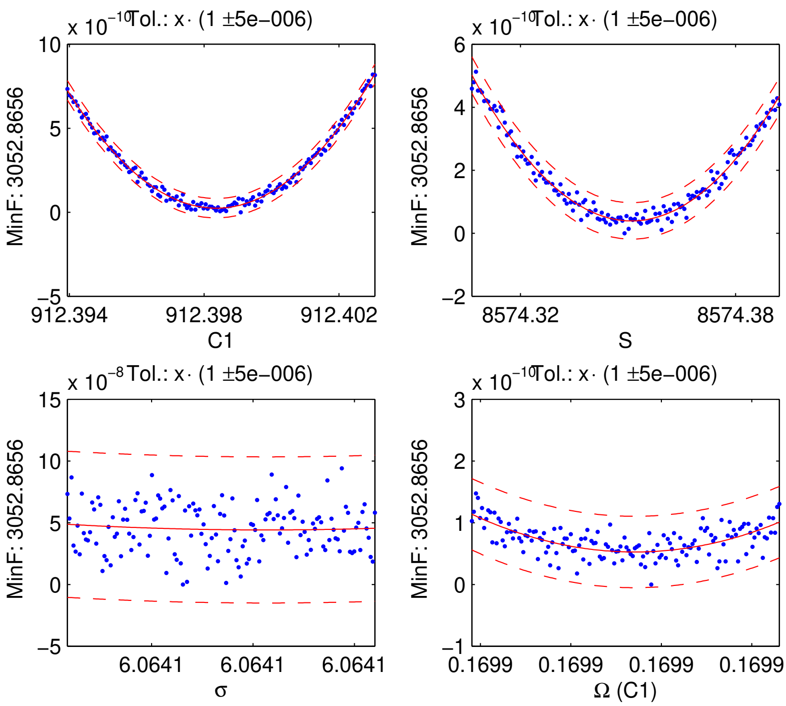


Figure 7.4: Log-likelihood profiles around optimum.

The residuals from the fitted curve are uncorrelated and the 2nd degree polynomial shape is thus a good approximation without bias. The standard deviation of the residuals from the fitted curve gives an indication of the noise. In the plots in Figure 7.4 the smooth curve has been outlined with a ±3 standard deviations confidence band. The width of the bands is shown in Table 7.1. For this example the σ_{ISR} direction appears to have the most significant noise with a standard deviation around 10^{-7}.

Parameter	C_1	S	σ_{ISR}	Ω_{C_1}
Width of band	$1.1586 \cdot 10^{-10}$	$1.1577 \cdot 10^{-10}$	$1.1832 \cdot 10^{-7}$	$1.1562 \cdot 10^{-10}$

Table 7.1: Confidence bands for the population likelihood function.

The reason for this noise have not been studied more thoroughly and therefore just a few observations regarding the problem will be presented here. The individual likelihood functions are for the example shown above entirely smooth, that is the function values have increments of machine uncertainty size. The population likelihood function is a sum of the individual optimizations and the Hessian approximations for the individual likelihood functions. The noise must thus arise from a combination of these two. An attempt was made to set a very low termination criteria for the individual optimizations, but the resulting noise were very similar for all parameters to those seen in Table 7.1.

7.2.2 Results of deconvolution

The found optimal parameters are shown in Table 7.2 together with the standard deviation and coefficient of variation. Parameter correlations are shown in Table 7.3. To verify that an optimum of the likelihood function has been found a likelihood profile plot is made (appendix Figure B.1) which confirms the optimum.

Parameter	Estimate	Std. deviation	CV
C_1	912.5	119.1	0.13
S	8574.6	1533.9	0.18
σ_{ISR}	6.064	0.277	0.05
Ω_{C_1}	0.170	0.083	0.49

Table 7.2: Parameter estimates for two-compartment C-peptide model.

	C_1	S	σ_{ISR}	Ω_{C_1}
C_1	1			
S	0.0124	1		
σ_{ISR}	0.0026	-0.2944	1	
Ω_{C_1}	-0.1426	-0.0054	-0.0196	1

Table 7.3: Correlation estimates.

Most noticeable is the negative correlation between S and σ_{ISR}. This is as expected since a smaller S leads to a less smooth estimate of ISR with large increments in the Wiener process and thus a larger σ_{ISR}.

An analysis of the 1-step prediction rediduals for the model is found in Figure 7.5. The residuals are studentized by the prediction variance from the Kalman filter to remove the effect of the varying time intervals. The residuals for all individuals have been appended to each other into one vector containing $35 \cdot 12 = 420$ residuals. Looking at the figure, the overall the assumption of normality for the residuals seems to be reasonable, although there seems to be too heavy tails. This can also be seen from the 'S' shape in the QQ-plot.

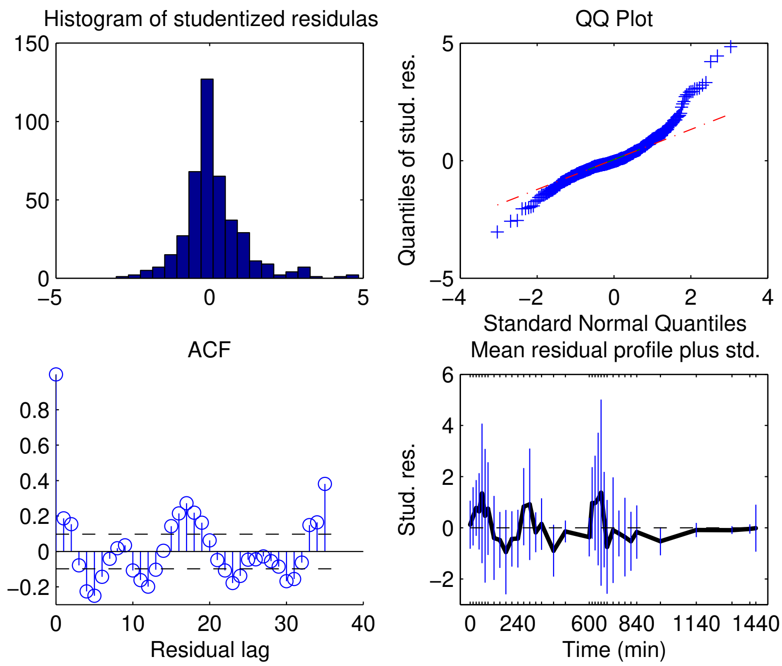


Figure 7.5: APL for $\theta_{opt} \pm 5\%$.

The autocorrelation function shows a large correlation for lag 35 equal to the number of samples per individual. This indicates that there is a large bias in the model i.e. the model is consistently over and under estimating at the same time points for all individuals. This is expected since the model will use the last update of ISR for prediction. Thus when ISR has peaked then the following prediction of C_1 will be to high since the model does not know that ISR has started to decrease.

Figure 7.6 shows the predictions of C-peptide for the 1st and 2nd individual. The figure visualizes the bias indicated by the ACF - it is underestimating before peaks and over estimating after peaks. The bias is more clearly seen in the mean profile of the residuals shown as the lower right plot in Figure 7.5.

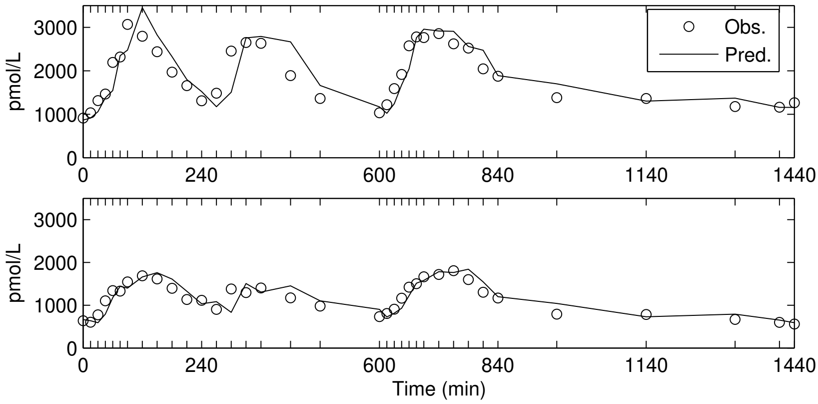


Figure 7.6: C-Peptide prediction for individual 1 and 2.

The problem with bias in the predictions using the model is not of great concern since it is not the intended use. The aim of the model is to estimate ISR, and here the Kalman smoothing technique eliminates the bias by using all observations at all times. This yields an optimal estimate of ISR based on the model. More importantly it is also seen from Figure 7.6 that the individualized starting concentration seems appropriate for the first two individuals and this also holds for the remaining 10.

Figure 7.7 shows the smoothed estimate of ISR for the first two individuals together with the $\pm$ 1 standard deviation band equal to a 67% confidence interval. The assumption of steady state in the beginning defines the initial level of ISR based on C_1 and this appears valid.

The width of the confidence bands is varying but does not depend on the data since both the drift and diffusion term of model are state independent. Thus the uncertainty of ISR only increases with distance a to sample point and is therefore mainly dependent on the sampling rate. This is not a desirable property. An example is the time after T=900min where ISR has stabilized and is thus fairly certain. Due to the low sampling rate the model estimates large confidence band between the observations which seems large for a stabilized system and may not be realistic.

Taking a closer look at the confidence interval for ISR (Figure 7.8) reveals that it is in fact at its narrowest just prior to a measurement. The confidence

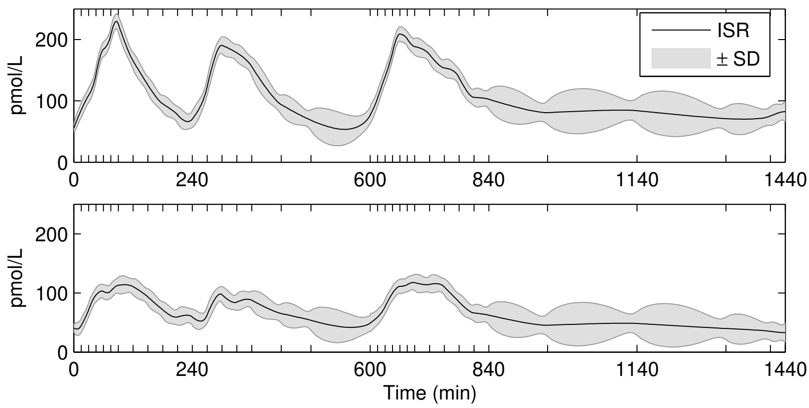


Figure 7.7: Smoothed estimate of ISR for individual 1 and 2.

interval for C_1 is at its narrowest at a measurement, since it is directly observed. The reason for the difference is that C_1 depends on ISR, and the most certain estimate of ISR is thus found just prior to the most certain estimate of C_1. Consequently also for C_2 the narrowest band is found after a measurement since it depends on C_1.

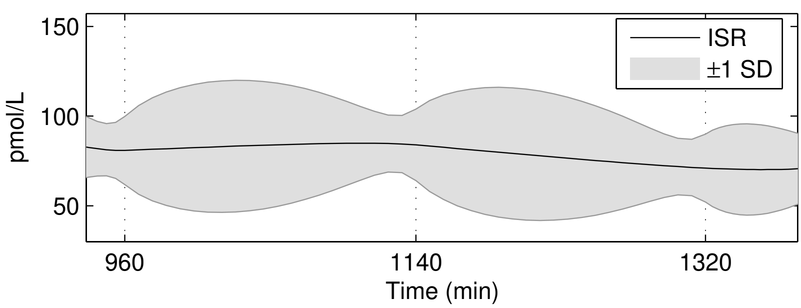


Figure 7.8: Behavior of ISR uncertainty around measurements.

To conclude on the deconvolution model it has been shown that the ISR and uncertainty can be found by deconvolution. The analysis showed signs of bias for the prediction, but this can be eliminated by using the Kalman smoothing technique to obtain the best possible estimate of ISR based on the model.

7.2.3 Log Model

The previous analysis with the two-compartment model for C-peptide was based on the assumption of *additive* gaussian noise for the C-peptide concentration measurements. However, when dealing with measurements of concentrations it is often found that it is more appropriate to use a *proportional* noise model. This will be tried in the following to investigate whether it might improve the model performance.

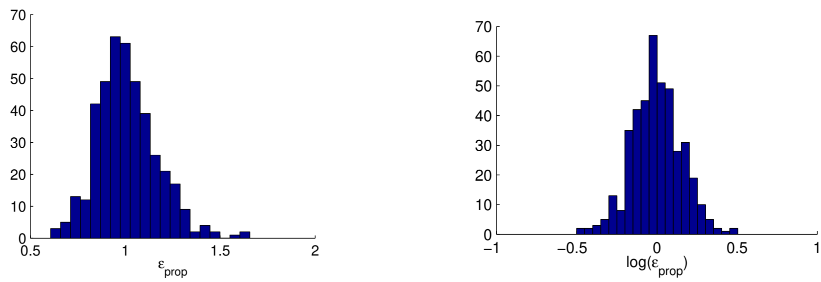
The PSM prototype works with the assumption that the error term in the observation equation is additive. In order to work with proportional error terms a log transformation can be used. The proportional error can be stated and reformulated for PSM as follows.

$$\boldsymbol{y} = \hat{\boldsymbol{y}}\epsilon_{prop} \tag{7.8}$$

$$\log(\boldsymbol{y}) = \log(\hat{\boldsymbol{y}}) + \log(\epsilon_{prop}) \tag{7.9}$$

Using the reformulation in equation (7.9) it can be seen that the $\log(\epsilon_{prop})$ should be normally distributed or the ϵ_{prop} log normal distributed.

The predictions from the standard two-compartment model are used to form the ϵ_{prop} using equation (7.8). In Figure 7.9(a) a slight skewness is noticed but the log-transformation removes this skewness. This indicates that an analysis using a proportional error term should be performed.



(a) Histogram of ϵ_{prop} (b) Histogram of $\log(\epsilon_{prop})$

Figure 7.9: Histrogram of the relative error.

The standard two-compartment model is transformed to examine the effect of proportional error compared to additive error. The transformation only occurs in the observation equations. The state equations remain the same.

Log Model

Description: Identical to the standard two-compartment model formulation but with a log transformed observation equation.

State variables:

$$\boldsymbol{x} = [\ C_1 \quad C_2 \quad ISR\ ]^T \tag{7.10}$$

Initial Conditions:

$$C_i = C_1^0 \exp \eta_1 \tag{7.11}$$

$$\boldsymbol{x}_0 = \begin{bmatrix} C_i & \frac{k1}{k2} C_i & k_e C_i \end{bmatrix}^T \tag{7.12}$$

Model:

$$d\boldsymbol{x} = \begin{bmatrix} -(k_1 + k_e) & k_2 & 1 \\ k_1 & -k_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}\ dt + diag \begin{bmatrix} 0 \\ 0 \\ \sigma_{ISR} \end{bmatrix} d\boldsymbol{\omega} \tag{7.13}$$

Output :

$$log(y) = log(\boldsymbol{x}_1) + \epsilon, \qquad \text{where } \epsilon \in N(0, S) \tag{7.14}$$

Parameters to estimate

$$\boldsymbol{\theta} = [\ C_1^0 \quad \sigma_{ISR} \quad S] \tag{7.15}$$

The new model will have a new set of parameters that maximizes the likelihood for the data. The initial starting parameters for the parameter estimation is based on the previous model estimates. The σ_{ISR} parameter is however affected by the reformulation. By looking at Figure 7.9(b) is seen that the residuals are expected to be within 0 ± 0.3. An appropriate starting guess is thus $S = (0.3/2)^2 = 0.02$.

The Log Model has properties that are less convenient. The prediction for the first compartment can for poor choices of parameters become negative. This is not physiologically correct but the model holds no limitations. This will result in a observation prediction that is the logarithm to a negative number and this can not be handled.

The implementation is unable to handle algebraic inequations or equations for the states, so the limitation is a hard-coded in the $\boldsymbol{f}(\cdot)$ function that ensures a positive value in the states. This is not a desirable solution but it is necessary to complete the parameter estimation. The hope is that the optimal parameters will not lead to any negative numbers in the states.

The parameters were estimated and the likelihood profiles can be seen in appendix Figure B.2. In Table 7.4 the parameter estimates can be seen together with the estimated standard deviation. The correlation matrix for the parameters can be seen in table 7.5. By looking at the tables it is seen that the estimates for C_1 and S are fairly certain and non-correlated. On the other hand the estimate for S is correlated with the two others and has a high uncertainty. This indicates that the Log Model has had difficulties separating measurement noise from system noise for ISR.

Parameter	Estimate	Std. deviation	CV
C_1	903.33	121.23	0.13
S	0.00227	0.04652	20.48
σ_{ISR}	5.5526	0.2607	0.05

Table 7.4: Parameter estimates for the Log Model.

	C_1	S	σ_{ISR}
C_1	1		
S	0.379	1	
σ_{ISR}	0.003	-0.374	1

Table 7.5: Correlation matrix for the estimated parameters

The individual predictions can be seen in appendix Figure B.3, note that there are no negative predictions. The distribution of errors and general overview of the residuals can be seen in Figure 7.10. The histogram shows the distribution of the studentized residuals. The histogram and the QQ-plot shows that the studentized residuals are fairly normally distributed and perhaps slightly better than for the additive error model.

The ACF shows some bias in the residual and the peak at $lag = 35$ is still clearly seen. The bias is also seen in the mean profile for the residuals. The model still shows signs of too slow adjustment for prediction. The residuals are positive as the C-peptide levels increase and negative at decrease.

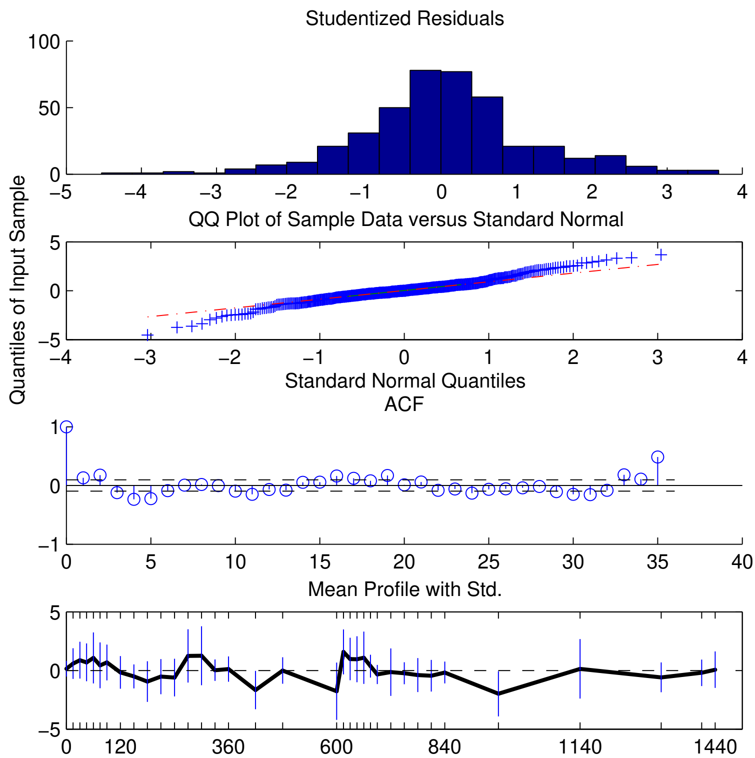


Figure 7.10: Residual analysis of the Log Model.

The Log Model is widely used to describe data since it enables the use of proportional error. The downside is a mathematically unstable model that requires a number of hacks to work during parameter estimation and also it appears to be more difficult for the model to separate measurement and system noise. It has been decided that shortcomings of the Log model outweigh the slight improvement in distribution of the residuals and thus the model with additive noise is concluded to be most adequate.

7.3 Intervention model for ISR

This section will focus on not just estimating but also at the same time modelling the insulin secretion rate (ISR) in order to achieve a lower uncertainty estimate for ISR. The idea is to utilize the knowledge of the three meal times and incorporate it into the model.

An attempt was made modelling the ISR through a predefined input which resembled the peaks after meals and the input was then scaled individually. The complete analysis can be found in appendix B.3.2 but the result was that the decay did not correspond to decay seen in data. A solution that included better control of the 3 peak structure is wanted.

According to the study plan a meal is to be digested over a period no longer than 20 min. but due to the sampling rate of data it will be assumed to last 30 min. Based on a review of the previously estimated ISR curves, it is assumed that ISR is increasing during meal intake and afterwards somewhat slowly decreasing in an exponential decay. This can be modelled with an intervention model with a 30 min. input impulse burst after meal times. It is observed that ISR does not decay toward zero and therefore the model for ISR also contains an basal level for ISR.

The modelling of ISR as an intervention model using state space models enables one to control the order of decay. The desired order of decay for the ISR is 2 hence 1 extra state denoted Q is required.

In order to limit the number of parameters it is chosen to model each of the three meals with identical parameters. It is without doubt a too simple approach since the meals and time of day are likely to influence the dynamics of ISR. The positive coefficients a_1 and a_2 defines the decay from the 2 states and combined they constitute the double exponential decay. The gain for the process (or more loosely speaking the height of the three peak profile) is controlled by a single parameter K and the basal level is controlled by B. Each of these two parameters are modelled as individual parameters, since they are both observed to vary significantly between individuals. As the model still can not be assumed to constitute a sufficient modelling of ISR a stochastic contribution ω is added to the ISR.

The relation of the two states defining the intervention model for ISR is shown in Eq. (7.16) and (7.17). The parameters for the ISR model is estimated along with the remaining parameters.

$$dISR = (-a_1 ISR + a_1 Q + B)dt + d\omega \tag{7.16}$$
$$dQ = (-a_2 Q + a_2 K u_1)dt \tag{7.17}$$

In Figure 7.11 an overview of the ISR model is shown. The input is a 2 dimensional input $u = [\ u_1 \ \ u_2\ ]^T$. The first dimension describes the intervention part by having ones at time points where meals are served.[3] Since the meals are assumed to last 30 minutes and remembering the first order hold on input, the input u_1 is one at time points (0, 15, 240, 600, 615) and zero otherwise. The second dimension in the input is constantly 1 and is a standard work around for a linear model used to add a constant in a differential equation. In the present case the constant creates the basal level for ISR.

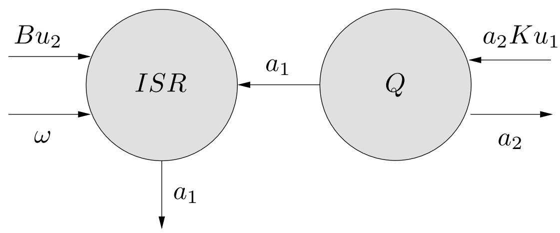


Figure 7.11: Model for ISR with second order dynamics.

The new ISR modelling is appended onto a standard two-compartment model for C-peptide measurements, see Figure 7.12. The model is as usual initiated at steady state, but it is decided to fix the initial concentration in C_1 to C_1^0 = 900pmol/L based on previous results in order to limit the number of model parameters. It is also found necessary to remove the individual estimate of C_1 since the individual optimizations will otherwise be 3-dimensional and thereby too cumbersome.

The modelling of ISR will give this model the property that it enables simulation of the entire system since the random walk is only used to describe deviations from the ISR model and will thus have a mean close to zero. Previous models have had ISR as a complete random walk, which has forced the estimated random walk to be positive. These models are unusable for simulation purposes. This model has ISR modelled and a simulation will thus yield different outcomes of ISR which are biologically feasible.

[3] Meals are served at time points 0, 240 and 600 minutes.

Intervention Model

Description: A two-compartment model with ISR modelled via an intervention model with a double exponential decay for ISR.

State variables:

$$\boldsymbol{x} = \begin{bmatrix} C_1 & C_2 & ISR & Q \end{bmatrix}^T \tag{7.18}$$

Initial Conditions:

$$C_1^0 = 900 \tag{7.19}$$

$$\boldsymbol{x}_0 = \begin{bmatrix} C_1^0 & \frac{k1}{k2}C_1^0 & k_e C_1^0 & 0 \end{bmatrix}^T \tag{7.20}$$

Model:

$$d\boldsymbol{x} = \left(\begin{bmatrix} -(k_1+k_e) & k_2 & 1 & 0 \\ k_1 & -k_2 & 0 & 0 \\ 0 & 0 & -a_1 & a_1 \\ 0 & 0 & 0 & -a_2 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & B\exp(\eta_2) \\ a_2 K \exp(\eta_1) & 0 \end{bmatrix} \boldsymbol{u} \right) dt + \operatorname{diag} \begin{bmatrix} 0 \\ 0 \\ \sigma_{ISR} \\ 0 \end{bmatrix} d\boldsymbol{\omega} \tag{7.21}$$

Output:

$$y = \boldsymbol{C}_1 + \epsilon, \qquad \text{where } \epsilon \in N(0, S) \tag{7.22}$$

Parameters to estimate :

$$\boldsymbol{\theta} = \begin{bmatrix} a_1 & a_2 & \sigma_{ISR} & K & B \end{bmatrix}^T \tag{7.23}$$

The estimated parameters are based on a starting guess which is found by initial experimentation with the model. The coefficients a_1 and a_2 should be positive and chosen to give a reasonable decay profile. The baseline B is found by assuming that the ISR is at steady state at the end of the deconvolved ISR profiles from the previous section. The mean end level is 64.8 pmol/L and it can thus be expected to approximately find $B/a_1 = 64.8$.

The model consists of 5 population and 2 individual parameters. This makes the minimization of the population likelihood function a huge computational task. The optimization has been carried out over several steps where the resulting intermediate likelihood profile curves have been used as new start guesses. The main problem has been noise in the likelihood function which makes gradient based optimization difficult and the problem naturally increases with the size of the parameter space.

The profile likelihood for the found optimal parameters can be seen in appendix Figure B.19. The optimum is found for each parameter and the curves are almost symmetrical around the center. The curvature of the profiles gives a hint to the uncertainty for each parameter but this will be calculated more accurately later on. The optimal ηs are plotted in appendix Figure B.20 and it can be seen that they are close to being normally distributed. This indicates that the population parameters handle the overall variation and the individual parameters only handle the individual variation from the population.

Parameter	Estimate	Std. deviation	CV
a_1	0.02798	0.0038	0.13
a_2	0.01048	0.0015	0.14
σ_{ISR}	6.9861	0.5590	0.08
K	427.63	67.8181	0.16
B	1.7434	0.3378	0.19

Table 7.6: Parameter estimates for the Intervention Model.

	a_1	a_2	σ_{ISR}	K	B
a_1	1				
a_2	-0.7197	1			
σ_{ISR}	0.8038	-0.5952	1		
K	0.1296	-0.3546	0.1161	1	
B	0.6790	-0.3923	0.5472	-0.0236	1

Table 7.7: Parameter correlation estimates for the Intervention Model.

In Table 7.6 it can be seen that the parameters are well estimated and are significantly different from zero. The high correlations indicate that the model parameters have high influence on each other and it might be possible to reformulate the model with fewer parameters.

In Figure 7.13 the predictions can be seen for individuals 1 and 2. The predictions for all 12 individuals can be seen in appendix Figure B.21. It can be seen that the bias around increasing and decreasing ISR levels have disappeared.

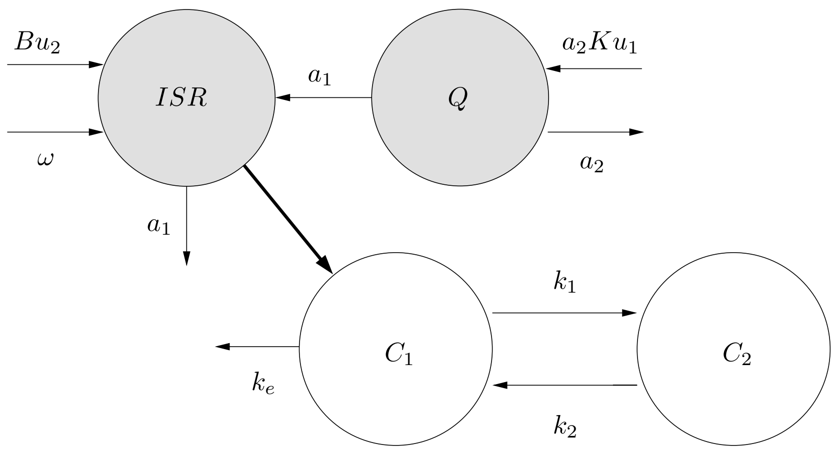


Figure 7.12: Overview of the Intervention Model.

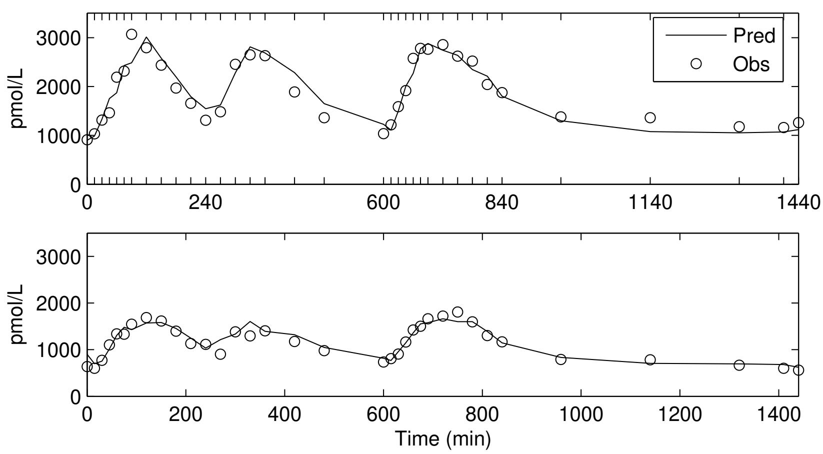


Figure 7.13: Predictions and C-peptide observations for individual 1 and 2.

Due to the fact that the model does not include the individual estimate of C_1^0 the first residual is extremely large, since the first observation is used to find the initial level. The standard deviation of the 1-step prediction residuals for all individuals are 210.4 pmol/L whereas the same for the Standard two-compartment C-peptide Model is 287.2 pmol/L. The overall conclusion on the predictions are that they fit the observations well and show a significant improvement to the previous model.

The residual analysis supports this impression. In Figure 7.14 various examinations of the residuals are shown. The histogram shows some skewness but overall not far from a normal distribution which the QQ-plot verifies. The ACF shows no significant trends in the residuals and the auto-correlation for lag 35 is reduced.

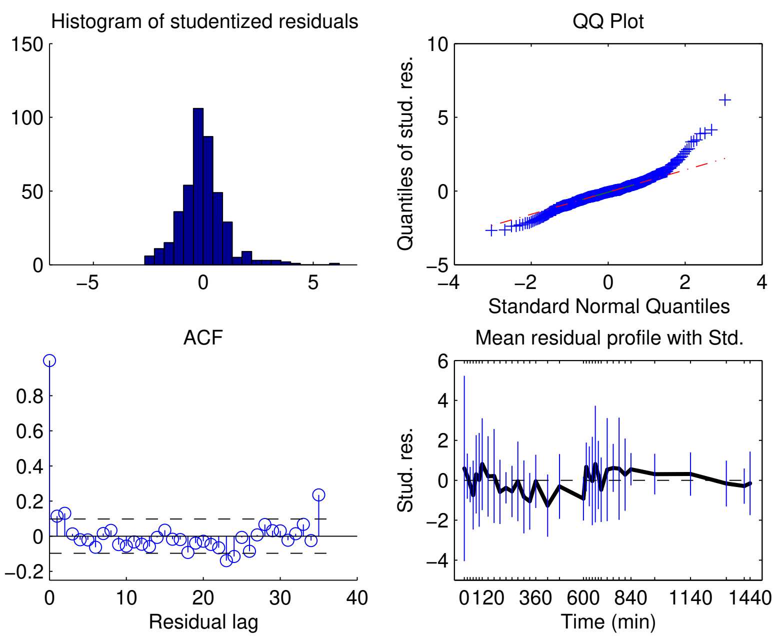


Figure 7.14: Analysis of the prediction residuals from the Intervention Model.

The improvement in the intervention model compared to previous models is the modelling of the ISR. Previously, the ISR has been modelled as a random walk. However, all the properties of a random walk were not satisfied. The ISR should not be negative and a prior knowledge on the expected behavior is known. In Figure 7.15 the smoothed and the modelled ISR are shown and furthermore the difference which should be the random walk is plotted. The difference has mean zero but the increments should be larger in larger time intervals. The difference for these 2 individuals does not exhibit this behavior but it is closer to a random walk than previous ISR models.

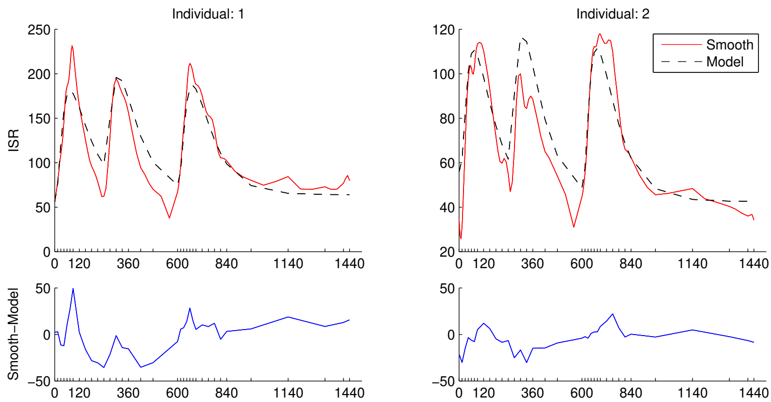


Figure 7.15: Smoothed and model ISR for individual 1 and 2.

The model and smoothed ISR for all individuals can be seen in appendix Figure B.22. The ISR model should be able to fit to the common height of the hills, the increase and decrease in levels and the baseline seen at the end. For all individuals the model for ISR can been seen to fit nicely and especially the baselines with the individually estimated levels are modelled close to the smoothed result. It appears that middle hill is generally slightly lower than the two others, but due to the assumption of equal dynamics for all three meals the model lies above the middle hill.

It is important to note that the estimate of ISR is almost identical to the estimate found using the Standard two-compartment C-peptide Model. This is fairly obvious since the C-peptide models are identical and the small differences in the ISR estimates by the two models only are caused mainly by a different estimate of the measurement error S for the two models. Appendix Figure B.23 shows smoothed estimates of ISR for the two models. However, it was the hope that the modelling of ISR would yield more narrow confidence bands. Figure 7.16 shows ISR for the first two individuals and by comparison with Figure 7.7 it is seen that this is not the case. The reason is that $\sigma_{ISR} = 6.99$ estimate is similar to the previous estimate of 6.06. At first this may seem illogical since the range covered by the random walk is much smaller in the Intervention Model. However, σ is estimated based on the *increments* in the random walk, which is of similar size. The conclusion is thus that the new model does not give more narrow confidence bands, but they are probably more trustworthy since they are based on an estimate from a more true random walk process.

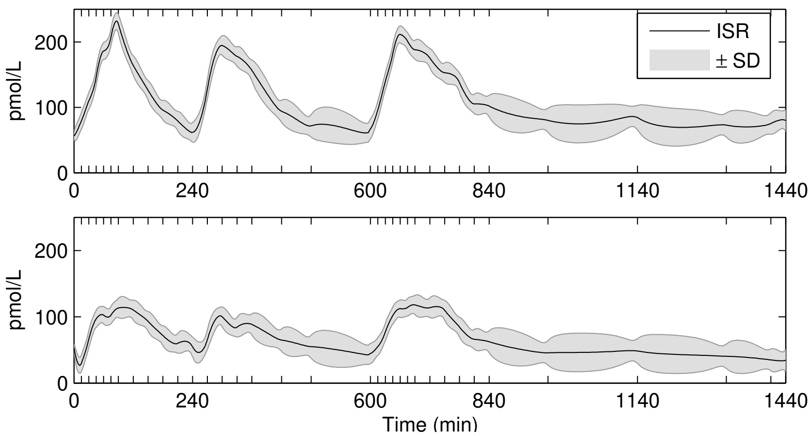


Figure 7.16: Smoothed estimate of ISR for individual 1 and 2.

The model analysis has successfully shown that it is possible to model the C-peptide measurements accurately by including ISR knowledge into an intervention model.

The residual errors were less correlated and the properties of the deconvolved random walk are much closer to assumptions.

The confidence bands around ISR shows no improvement in width but can probably be trusted to a greater extent compared to earlier models. Finally, the model is more adequate for simulation if required.

7.4 Combined models

The previous models of ISR have been based solely on the C-peptide measurements. However, it may also be useful to include the insulin measurements as well in a combined model [Kjems et al. 2001]. Insulin has a very short half-life and is thus usually modelled with a one-compartment model. The hope is that using the insulin and C-peptide measurement in combination will lead to a better estimate of ISR.

7.4.1 Initial approach

It is known that equi-molar amounts of C-peptide and insulin are secreted from the pancreas. The pre-hepatic insulin secretion rate (ISR) are thus equal to the C-peptide secretion rate. It is difficult to asses how much of the insulin is extracted by the liver. For this initial model it is chosen to model the extraction simply by a fixed constant denoted F_c. Using this with the 1-compartment insulin model results in the combined model illustrated in Figure 7.17.

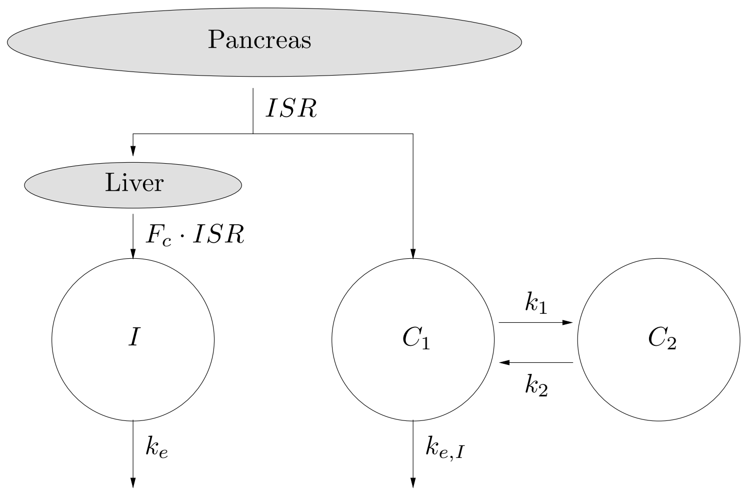


Figure 7.17: Dynamics of the Combined model.

The insulin part of the model introduces two new parameters into the model, namely the liver extraction F and the insulin elimination constant $k_{e,I}$. Both are chosen to be maximum likelihood estimated. It is assumed that the model is initiated in steady state, and this fixes C_2^0 and ISR^0 based on C_1^0 as previously shown in Eq. 7.1 page 56. Also the insulin compartment can be fixed based on C_1^0 since it must hold that $F_c \cdot ISR = k_{e,I} \cdot V$ in steady state. This gives $V = ISR \cdot F_c / k_{e,I} = C_1 k_e \cdot F_c / k_{e,I}$. The resulting model is shown in the model box below.

Initial Combined Model

Description: A combined model with 2 compartments for C-peptide and 1 compartment for insulin. Liver extraction F_c is a fixed constant.

State variables:

$$\boldsymbol{x} = [\, C_1 \quad C_2 \quad I \quad ISR\,]^T \tag{7.24}$$

Initial Conditions:

$$C_i = C_1 \exp(\eta) \tag{7.25}$$

$$\boldsymbol{x}_0 = \left[\, C_i \quad \frac{k_1}{k_2} C_i \quad \frac{k_e}{k_{e,I}} F_c C_i \quad k_e C_i \,\right] \tag{7.26}$$

Model:

$$d\boldsymbol{x} = \begin{bmatrix} -(k_1+k_e) & k_2 & 0 & 1 \\ k_1 & -k_2 & 0 & 0 \\ 0 & 0 & -k_{e,I} & F_c \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}\; dt + diag \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sigma_{ISR} \end{pmatrix} d\boldsymbol{\omega} \tag{7.27}$$

Output:

$$\boldsymbol{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{x} + \epsilon, \qquad \text{where } \epsilon \in N\left(\mathbf{0}, \begin{bmatrix} S_C & 0 \\ 0 & S_I \end{bmatrix}\right) \tag{7.28}$$

Parameters to estimate

$$\boldsymbol{\theta} = [\, C_1^0 \quad F_c \quad k_{e,I} \quad S_C \quad S_I \quad \sigma_{ISR}] \tag{7.29}$$

7.4.1.1 Model estimation and performance

The model is setup with the initial concentration in C_1^0 as individual parameter. The variance of the individual ηs for C_1 are set to $\Omega_{C_1} = 0.15$ and the C-peptide kinetic parameters are set to the Van Cauter values. The remaining population parameters to be estimated are thereby C_1, F_c, $k_{e,I}$, S_C, S_I and σ_{SR}. The estimates are shown in Table 7.8 and the correlation estimates for the parameters are shown in Table 7.9. Most noticeable are the correlation of

0.9956 between F_c and $k_{e,I}$. This shows that the effect of the two parameters cannot be separated, and thus only the estimated fraction $F_c/k_{e,I}$ should be trusted. The high correlation between F_c and $k_{e,I}$ also results in rather high uncertainty estimates which again leads to a high coefficient of variation (CV) for the two parameters.

Parameter	Estimate	Std. deviation	CV
C_1	925.16	113.49	0.12
F_c	0.3047	0.0639	0.21
$k_{e,I}$	0.1410	0.0299	0.21
S_C	7483.9	1328.7	0.18
S_I	13514	952.14	0.07
σ_{SR}	6.1682	0.2774	0.05

Table 7.8: Parameter estimates for Initial Combined Model.

	C_1	F_c	$k_{e,I}$	S_C	S_I	σ_{SR}
C_1	1					
F_c	0.0176	1				
$k_{e,I}$	0.0168	0.9956	1			
S_C	0.0089	0.0027	0.0012	1		
S_I	-0.0065	-0.0534	-0.0528	0.0131	1	
σ_{SR}	0.0078	-0.0354	-0.0343	-0.2925	-0.0188	1

Table 7.9: Parameter correlation estimates.

To verify that the optimal parameters have been found the profile likelihoods have been plotted around the parameter estimates (see appendix Figure B.26). The plot shows that a satisfactory optimum has been found. It should also hold that $\eta \in N(0, \Omega_{C_1})$. In order to limit the parameters to be estimated, Ω_{C_1} was set to 0.15. The sample variance of the estimated ηs equals 0.1425, and is thus close to the specified value. To check the normal distribution, a QQ-plot is made and shown in Figure 7.18. From the figure it is seen that the ηs can be assumed to be normally distributed.

However, it is of concern to see that S_I has been estimated to $13514 = 116.2^2$ whereas $S_C = 86.5^2$. This insulin measurement variance seems much to large, especially knowing that insulin concentrations are always lower than C-peptide concentrations.

This indicates that the insulin part of the combined model is too simple, which is further supported by an analysis of the studentized residuals found in Figure B.24. The histogram and QQ-plots show that the residuals are fairly well approximated by a normal distribution although they show a tendency toward too heavy tails. The insulin residual histogram appears skewed with a few very

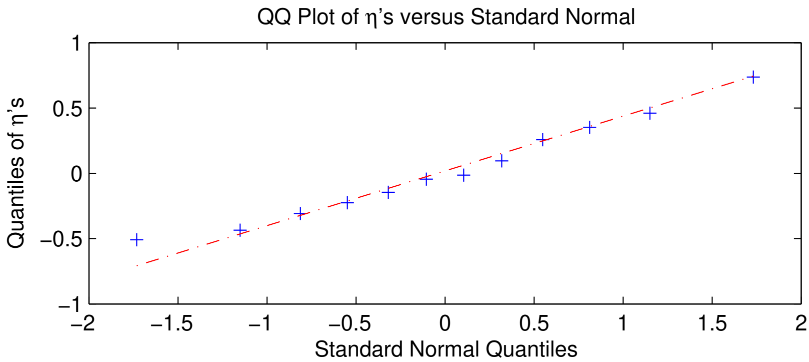


Figure 7.18: QQ-plot of individually estimated η-values.

large residuals and (as a consequence) a too large proportion of negative residuals. This is also seen in the autocorrelation for the insulin residuals, which are positive for all lags. These observations indicates that the insulin model is insufficient and is generally over-estimating.

The effect of the insufficient insulin model is that no extra accuracy is obtained for ISR using insulin measurements together with C-peptide measurements. The reason is that insulin measurements are given a low emphasis in the Kalman state update of ISR due to a large S_I. This results in basically unchanged estimates of the ISR compared to the Standard 2-compartment C-peptide Model.

The most probable deficiency of the combined model is the use of a constant liver insulin extraction proportion. Many studies have shown that a more complex model should be used e.g. Michaelis-Menten saturation as discussed in Section 2.1.1. Instead of using one of these models, a choice is made to instead use the combined model to *estimate* the dynamics of the liver extraction. This can be done by including F as a state and this will be the focus of the next proposed model.

7.4.2 Combined model to estimate liver extraction

The initial approach above to construct a combined model indicated a need for a time varying liver extraction, $F(t)$. This possible by including $F(t)$ in the state equation in order to be able to estimate it over time. $F(t)$ is modelled as a random walk similar to the approach for ISR. The insulin elimination constant is chosen to be a fixed number. In the model above $k_{e,I}$ was found to be 0.1410 and in a similar study by Kjems et al. (2001) using the same model also on type 2 diabetic patients, it was found that $k_{e,I} = 0.355$. It is chosen to use the latter value, since the analysis of the model above showed that the estimate of $k_{e,I}$ was highly correlated with the estimate of F.

In order to limit the parameter space for the optimization the estimate of C_1^0 is fixed to 900 $pmol/L$ and σ_{ISR} is fixed to 6.2. This is done since the C-peptide part of the model is not expected to change. The initial state C_1^0 is still estimated individually. The initial guess for the level of F is set to $F_0 = 0.2$, which is a little lower than the estimate of the constant F_c in the previous model.

The general layout of the model has not changed, and is thus still the one shown in Figure 7.17. The idea is that ISR will be estimated using the C-peptide part of the model as usual, and with an estimated ISR it will be possible to estimate $F(t)$ by using the insulin measurements and having a fixed insulin elimination constant. The model is non-linear due to the multiplication of the two states F and ISR in the state equations. The model details are shown in the box below.

Liver Extraction Model

Description: A combined model with 2 compartments for C-peptide and 1 compartment for insulin. Liver extraction F is modelled as a random walk.

State variables:

$$\boldsymbol{x} = [\ C_1 \quad C_2 \quad I \quad ISR \quad F\ ]^T \tag{7.30}$$

Initial Conditions:

$$C_i = C_1^0 \exp(\eta) \tag{7.31}$$

$$\boldsymbol{x}_0 = \begin{bmatrix} C_i & \frac{k_1}{k_2}C_i & \frac{k_e}{k_{e,I}}F_0C_i & k_eC_i & F_0 \end{bmatrix}^T \tag{7.32}$$

Model:

$$d\boldsymbol{x} = \begin{bmatrix} -(k_1+k_e)C_1 + k_2C_2 + ISR \\ k_1C_1 - k_2C_2 \\ -k_{e,I}I + F \cdot ISR \\ 0 \\ 0 \end{bmatrix} dt + diag\begin{pmatrix} 0 \\ 0 \\ 0 \\ \sigma_{ISR} \\ \sigma_F \end{pmatrix} d\boldsymbol{\omega} \tag{7.33}$$

Output:

$$\boldsymbol{y} = \begin{bmatrix} C_1 \\ I \end{bmatrix} + \epsilon, \qquad \text{where } \epsilon \in N\left(\mathbf{0}, \begin{bmatrix} S_C & 0 \\ 0 & S_I \end{bmatrix}\right) \tag{7.34}$$

Parameters to estimate:

$$\boldsymbol{\theta} = [\ S_C \quad S_I \quad \sigma_F\ ] \tag{7.35}$$

7.4.2.1 Model estimation

The population parameter estimates are shown in table 7.10. The estimate of S_C, the C-peptide measurement noise, seems reasonable as does the σ_F which is the Wiener noise gain for F. However the estimate of the insulin measurement noise for $S_I = 0.1$ is much to low. For practical purposes the insulin concentration measurement error is assumed to be around 1%-10%. The mean of all insulin measurements are 221pmol/L and it would thus roughly be expected to find a maximum error in the range from 2.21pmol/L to 22.1pmol/L. If this maximum error is seen as a 95% confidence interval this corresponds finding S_I in the range from 1 to 120[4]. There may be several factors causing this unreasonably low estimate of S_I, e.g. the starting guess for $F_0 = 0.2$ and the initialization of the state covariance. However, the most important thing is that the model for F is too flexible and it is thus not possible to separate the measurement noise from the variation of F.

Parameter	Estimate	Std. deviation	CV
S_C	4013.8	700.88	0.17
S_I	0.1	N/A	N/A
σ_F	0.0290	0.0003	0.01

Table 7.10: Parameter estimates for combined model with F as state.

	S_C	S_I	σ_F
S_C	1		
S_I	N/A	1	
σ_F	-0.0140	N/A	1

Table 7.11: Correlation estimates.

An analysis of the residuals (see Figure B.27) shows similar results as seen for the first combined model. It appears as if the improved model for F has removed some of the bias in the insulin residuals mainly for the last half of the time interval.

Figure 7.19 shows the profile likelihood function around the optimal parameters. The figure shows that the estimates of S_C and σ_F are accurate and robust. On the other hand there appears to be significant noise in the likelihood profile for S_I although it still seems to have a nice minimum around 10^{-2}. Due to this noise around the parameter estimate it was not possible to estimate the uncertainty for the estimate of S_I or find correlation with the other two parameters.

A look at the insulin predictions reveals a peculiar effect of the model. In Figure 7.20 the one step prediction of insulin measurements for individual 1 is seen. Due to the low estimate of insulin measurement noise the prediction update after measurement is almost equal to the last measurement. What is more surprising is the fact that the prediction remains constant between measurements. This seems to hold for all time intervals and is the same for all individuals.

[4] $2 \cdot \sqrt{S_I} = 221 \cdot 10\% \Rightarrow S_I = 120$

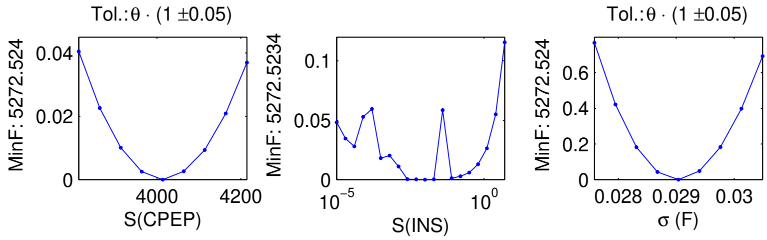


Figure 7.19: Likelihood function plotted around optimal parameters.

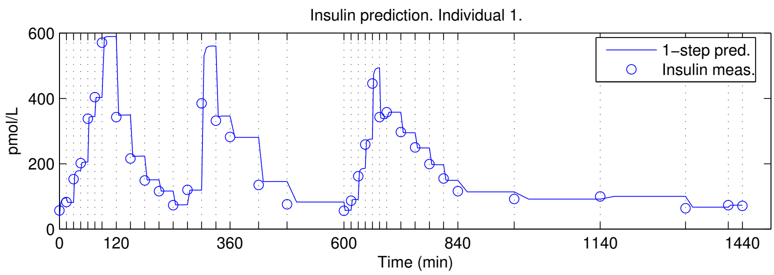


Figure 7.20: 1-step prediction of insulin measurements for individual 1.

A closer look at the model can tell more about what is going on. The insulin prediction if found by calculating the ODE solution for the state $X_{k+1|k}$ after the state is updated by the last measurement, $X_{k|k}$. By looking at the state equations (Eq. 7.33) it is seen that for the insulin prediction to remain constant, it must hold that $F_{k|k}ISR_{k|k} = k_{e,I}I_k$ (since $I_k \approx I_{k|k}$ due to the low S_I). F and ISR remains constant between measurements since $dF/dt = dISR/dt = 0$. Thus if F is updated such that $(F_{k|k}ISR_{k|k})/(k_{e,I}I_k) = 1$, then the prediction of insulin is constant. In Figure 7.21 this fraction has been found and shown with a mean profile for all individuals over time. The figure shows that it is in fact very close to 1 with minor deviations around meal times.

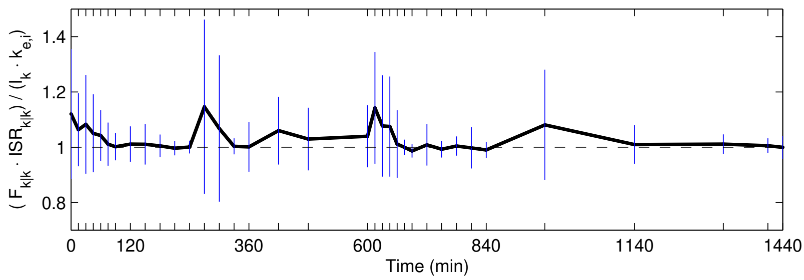


Figure 7.21: Mean profile for the fraction between the amount running in and out of the insulin compartment at the beginning of each time interval.

The conclusion on the analysis is that a less flexible model for F is needed such that it is possible to separate the liver extraction dynamics from the measurement noise.

7.4.3 Improved model for liver extraction

The analysis in the previous section has shown that the first Liver Extraction Model yields poor performance and that this most likely is due to a too flexible model for F. In order to solve this problem a slight change in the model setup for the liver extraction is made. It is chosen to model *the derivative* of F as a random walk instead of directly F as done before. This can be achieved by introducing a new state named X such that

$$\frac{dF}{dt} = X \tag{7.36}$$

$$dX = \sigma_X d\omega \tag{7.37}$$

The change in the model for F causes the increments of the derivative of F to be penalized by the Wiener noise gain σ instead of directly the increments of F. This results in a less flexible model for F where fluctuations are constrained. The change is incorporated into the existing model for the liver extraction which results in the model shown in the following.

Constrained Liver Extraction Model

Description: A combined model with 2 compartments for C-peptide and 1 compartment for insulin. Liver extraction F is constrained by modelling dF/dt as a random walk.

State variables:

$$\boldsymbol{x} = [\ C_1 \quad C_2 \quad I \quad ISR \quad F \quad X\ ]^T \tag{7.38}$$

Initial Conditions:

$$C_i = C_1^0 \exp(\eta_1) \tag{7.39}$$

$$\boldsymbol{x}_0 = \left[\ C_i \quad \frac{k_1}{k_2}C_i \quad \frac{k_e}{k_{e,I}}F_0C_i \quad k_eC_i \quad F_0 \quad X_0\ \right]^T \tag{7.40}$$

Model:

$$d\boldsymbol{x} = \begin{bmatrix} -(k_1+k_e)C_1 + k_2C_2 + ISR \\ k_1C_1 - k_2C_2 \\ -k_{e,I}I + F \cdot ISR \\ 0 \\ X \\ 0 \end{bmatrix} dt + diag \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sigma_{ISR} \\ 0 \\ \sigma_X \end{pmatrix} d\boldsymbol{\omega} \tag{7.41}$$

Output:

$$\boldsymbol{y} = \begin{bmatrix} C_1 \\ I \end{bmatrix} + \epsilon, \qquad \text{where } \epsilon \in N\left(\mathbf{0}, \begin{bmatrix} S_C & 0 \\ 0 & S_I \end{bmatrix}\right) \tag{7.42}$$

Parameters to estimate:

$$\boldsymbol{\theta} = [\ S_C \quad S_I \quad \sigma_X\ ] \tag{7.43}$$

The parameters to be estimated are S_C, S_I and σ_X. F_0 is fixed to 0.4 based on the mean of the smoothed estimate of $F(t=0)$ for all individuals from the previous model. X_0 is fixed to 0.01. The model estimates are shown in Table 7.12 and 7.13. The likelihood profiles and a residual analysis can be found in Figure B.30 and B.31. The figures verifies that a good optimum has been found and that no large deviations from normality of the residuals are present in the model.

Parameter	Estimate	Std. deviation	CV
S_C	5505.6	855.4	0.16
S_I	392.7	73.9	0.19
σ_X	$6.437 \cdot 10^{-4}$	$0.401 \cdot 10^{-4}$	0.06

Table 7.12: Parameter estimates for the Constrained Liver Extraction Model.

	S_C	S_I	σ_X
S_C	1		
S_I	0.0068	1	
σ_X	-0.0512	-0.3449	1

Table 7.13: Correlation estimates.

From the correlation matrix it is seen that S_I and σ_X are strongly negatively correlated. The reason is that a lower S_I yields a more fluctuating F and thereby a higher σ_X, hence giving a negative correlation between the two. However, due to the constrained model for F it is for this model possible to separate the two and find reasonable estimates.

As expected the Constrained Liver Extraction Model finds an ISR which is almost identical to the one found using just a C-peptide model. The smoothed liver extraction F is shown in Figure 7.22 for individual 1 and 2. The plots illustrates that the liver extraction cannot be assumed constant. The proportion sent through the liver, F, is below one over the entire time interval as it naturally should be. This also holds for 8 out of the remaining 10 individuals. For the two last individuals F varies between 0.5 and 1.8. This is however not of great concern, because F and $k_{e,I}$ are correlated and it is thus probably just indicating that $k_{e,I}$ for this particular individual is set to high.

It is expected that the estimate of F with the Constrained Liver Extraction Model should be more smooth compared to the previous unconstrained model. A comparison of the two models are shown in Figure 7.23 where the constrained movement is clearly seen. Moreover the confidence interval for F is also smoother due to the constrained movement of F.

It is of interest to see how ISR and F relates to each other. Appendix Figure B.32 shows the two plotted against each other. There is an obvious linear relationship between them. By testing for equal slope using a standard General Linear Model

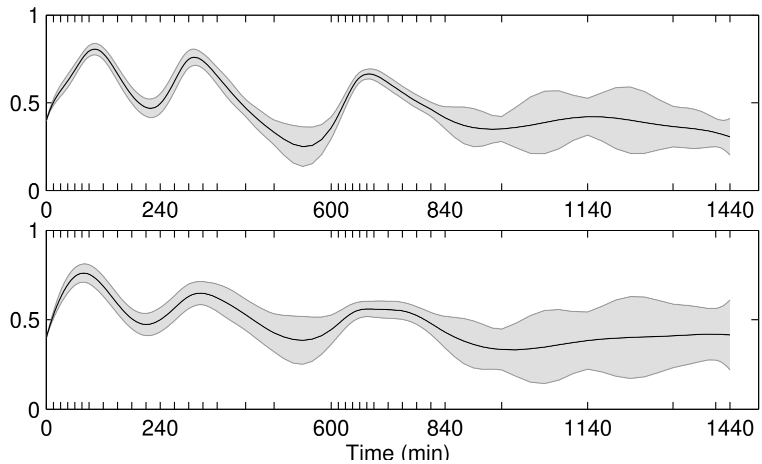


Figure 7.22: Smoothed liver extraction ±SD for individual 1 and 2.

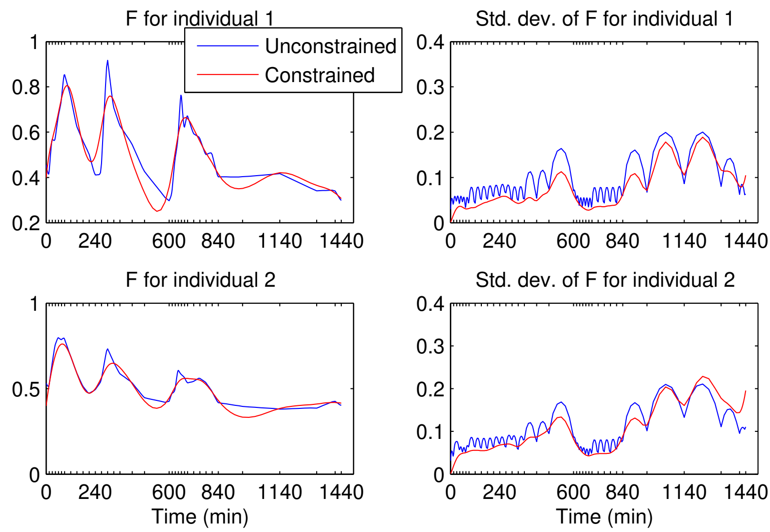


Figure 7.23: Comparison of estimation of F using the constrained and unconstrained model.

$$
\begin{aligned}
H_0 : F_{ij} &= \mu_i + \alpha \cdot ISR_{ij} + \epsilon_{ij} \\
H_1 : F_{ij} &= \mu_i + \alpha_i \cdot ISR_{ij} + \epsilon_{ij}
\end{aligned}
$$

where $i = 1...12$ and $j = 1...35$, the slopes are found to be significantly different (H_0 is rejected with $p < 0.001$). It should be noted though that this simple test does not take correlation between measurements into account, which would increase the p-value for the test.

Having found a linear relationship between ISR and F it is also of interest to see if more information about their co-variation can be revealed such as the presence of a hysteresis loop. To study this a phase plot of smoothed estimates of ISR and F for the first individual are plotted in Figure 7.24 (4 more are found in appendix Figure B.33). The smoothed estimates are subsampled with 5 extra points between measurements and is split up with one plot for each meal. Based on these plots it is not possible to draw any further conclusions on the relation between the two.

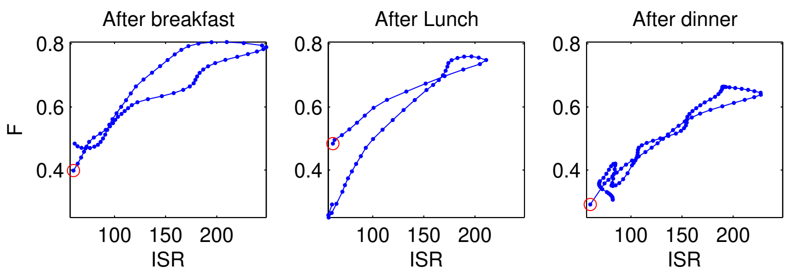


Figure 7.24: Phase plot of F vs ISR for the 1st individual; the circle is 1st obs.

In Figure 7.25 the smoothed confidence bands for ISR, F and X can be seen. They are scaled to resemble each other to visualize the delays between them. The insulin secretion rate has the lowest uncertainty just before an observation arrives. The F is estimated based on ISR and insulin observations. This means that F is optimally estimated just in between the minimal ISR uncertainty and observation time. X is the change in F and is best estimated just before the optimal for F. All these relations can be more clearly seen in the figure.

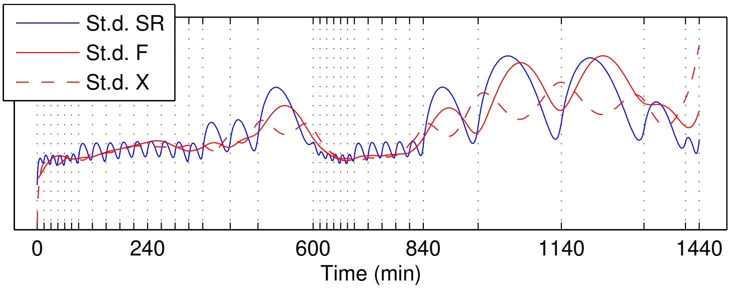


Figure 7.25: Relatively scaled comparison of the smoothed standard deviation for the first individual for state ISR, F and X.

7.4.4 Modelling summary

The PSM prototype has been used to develop different models for insulin and C-peptide measurements. Different models were proposed for deconvolution of the insulin secretion rate with different properties.

The intervention model included a model for the ISR thereby improving the residual and Wiener process properties. The deconvolved ISR and uncertainty are not significantly improved but the result is probably more trustworthy.

Finally, a model which exploited the knowledge on equi-molar secretion of C-peptide and insulin was analysed. The model makes use of the deconvolved ISR to estimate the time varying dynamics of the liver extraction. The final model proposed had a less flexible model for the extraction rate. This model did not over-fit the data but on the contrary showed a tendency towards more bias in the predictions. The analysis showed that this model is able to successfully estimate the liver extraction of insulin.

Chapter 8

Future Work with PSM

The work with implementing the prototype has resulted in several experiences. Moreover, the process with model building afterwards has provided experiences on functionalities that would be beneficial to include. These experiences should be considered before proceeding with the next version of PSM.

A large but necessary step is to move to another programming language. Matlab is ideal for numerical implementations but it lacks in speed and parallel computing options. The standard within scientific programming is Fortran[1] although other languages such as C++ or java are potentially faster but have shortcomings in other areas. Important factors when choosing programming language for this task is its ability to handle linear algebra calculations and the accessibility to already available modules - minimizers and ODE-solvers. PSM does not use any toolboxes in Matlab to obtain functionality and the main work is done through standard matrix calculations.

The numerical capabilities in Fortran are well ahead of competitors and furthermore the standard has for many years been Fortran in the academic world. This results in a great deal of scientific modules already being available in Fortran. An example is the used Quasi-Newton minimizer *ucminf* already implemented in Fortran and several ODE-solvers exists eg. ODEPACK[2]. Further investiga-

[1]Fortran 95 is the current Fortran version

[2]http://www.llnl.gov/CASC/odepack/ or http://www.netlib.org/odepack/

tion of an ideal ODE-Solver is advised as it is a requirement that it is thread safe. Based on these considerations Fortran would be the language of choice for a new implementation of PSM.

The program should also take full use of parallelism as the computational task is substantial. Several alternatives have emerged within recent years and limits are constantly expanded. The GRID technology could be an obvious choice as it offers enormous amounts of computer power at a low cost. The small data sizes which are common for PK/PD trials, are also ideal for GRID computing. The problem arises in the granularity of the parallelism. Many of the calculations in PSM are interlinked and in PSM there are several levels ideal for parallelization. The APL parallelization already implemented is ideal for the GRID due to long computational times and little data transfer.

The parameter estimation pr. individual is based on multiple Kalman filter calculations which are fast small calculations. Numerous small interlinked calculations are preferred to be performed on a High-Performance Computer setup compared to a GRID setup due to the overhead in communication on network connections.

A High-Performance Computing setup consists of several processors with access to shared memory. This makes communication fast and provides a vast amount of computer power as well. A High-Performance setup is typically more expensive compared to a GRID setup but can most often be found in scientific institutions. Furthermore, the trends from the processor industry is a move towards dual-core system or hyper threading making parallelism available on small systems. An interesting awaited arrival is the CELL processor which forms the basis of the PlayStation 3. It is targeted towards home entertainment but if its potential can be exploited within linear algebra the effects will be immense.

The optimal platform for a future implementation would be a High-Performance Computing setup also know as a Shared-Memory system. Shared-Memory Parallelism can be implemented in Fortran using the OpenMP[3] package. This package supports parallelism through meta tags that will make portability to single processor units easy. Very low level parallelism can be obtained by tuning compiler options. Some new compiles are able to create parallel calculations by automatically analyzing the code but the largest improvement is achieved using manual parallelization.

Another big challenge lies within the model specification. In Matlab the model specification is through script files which are not a part of the actual compiling. A simple and powerful model would be model specification through a text file e.g.

[3]www.openmp.org

similar to the NONMEM model specification. This approach requires a lexical parser that is able to translate the model specification into objects that can be evaluated just as functions can. This text file could also hold important values, e.g. for termination criteria for the optimization. The model specification should enable quick changes from the user such as which parameters to estimate, limits for parameters and what output to return from the estimation.

The model specification could be extended with a GUI interface making options and model specification easy accessible. The GUI should then prepare the model specification file for the user and start the desired functions.

Finally, a large effort should be put into the output of the program. The potential users of the program are from a wide range of fields – PK/PD, construction, chemical, physicist and statisticians each group having their own favorite program to analyse data. The output should comply with the needs from a wide range of users. The data format should be in a universal format that most programs are able to import. This a difficult task as the output consists of many different data types – predictions, covariance matrices and parameter estimates. Some research should be put into analyzing the different programs and their capabilities before deciding on a format. A potential candidate is netCDF[4] which is a universal data format that looks promising but no consensus has yet been reached on a common standard.

A final time estimate for a new implementation of PSM is not given as it depends on too many factors, for example choices of implementation, programming experience and to what extent available modules can be used.

[4]http://www.unidata.ucar.edu/software/netcdf/

Chapter 9

Discussion

This thesis covers a number of fields such as statistics, numerical computation, PK/PD Modelling and diabetes. Results and perspectives will be discussed in this chapter leading to the final conclusion in the next chapter.

The first part of the thesis presents an approach for using stochastic state space models in a mixed-effects setup. The model is based on a Kalman filter approach by assuming Gaussian conditional densities and that the individual likelihood function can be approximated by a 2nd order Taylor expansion in the FOCE approximation. An alternative approach is to use the Markov Chain Monte Carlo (MCMC) simulation to estimate the population likelihood funcion. This approach does not rely on the two assumptions. However, the computational task of the MCMC simulation is extensive and simulation studies suggest that the two assumptions yields a good approximation as long as the system is not highly non-linear [Singer 2002]. The results from an analysis of studentized residuals for the various models studied in this thesis suggested that the conditional densities are well approximated by a Gaussian distribution.

The implementation of the chosen approach encompasses two standard problems of solving differential equations and minimization of an objective function. The implemented prototype makes use of available modules to solve these problems - an ODE-solver and a minimizer.

The prediction in the Extended Kalman filter consists of coupled ordinary differential equations where an analytical solution is generally not available. The solution must thus be calculated numerically by an ODE-solver. For the prototype it has been chosen to use a standard Matlab interface which opens up for easy interchange between ODE-solvers. The ODE-solver of choice in the prototype is based on a robust algorithm which is able to handle stiff systems. The ODE-solver is a central part of the approach and the chosen solver is a compromise between accuracy and robustness and overall speed of the prototype. The importance of a robust ODE-solver is most clearly present in more complex models where the degree of non-linearity will make an accurate prediction more demanding. A robust ODE-solver with variable step length and error measurement is able to adopt and decrease step length when necessary whereas a simpler algorithm will just predict poorly.

The second module required is a minimizer. It is used to minimize the negative likelihood function for parameter estimation on both population and individual level. The individual likelihood functions are minimized to estimate the random effects η and the population likelihood is used to estimate the population parameters. The combination of an optimization of several optimizations results in an extremely demanding computational task. The minimizers' termination criteria and the step length for the gradient approximation influences the result. A further investigation of the linked effects could hopefully reveal valuable information. The current options are based on empirical investigations that resulted in robust results in the analysed models.

The outcome of the project has been a functional prototype that was validated and used in model building. A side product are numerous experiences on numerical issues and considerations on useful options that resulted in a set of recommendations for the next implementation. Fortran has been suggested as the evident choice of programming language mainly because of numerical capabilities and easy implementable parallelization. A key issue in the implementation is to consider parallelization since a considerable reduction in computation time is achievable.

The final prototype PSM was successfully shown to yield parameter estimates consistent with CTSM and NONMEM and to be able to handle SDEs correctly. However, the validation with NONMEM showed differences in approximate population likelihood values. This was investigated thoroughly but no conclusion on the difference could be made. This was due to the fact that no documentation was sufficiently detailed on the NONMEM likelihood calculation to establish a statistical and/or numerical reason for the observed difference. The difference in APL was accepted as the maximum likelihood parameter estimation yielded identical parameter estimates.

A further investigation revealed issues concerning the likelihood ratio test. The log-likelihood ratios found by NONMEM were consistently larger than for the prototype. As a consequence, NONMEM will more often reject the reduced model. This is a problem in modelling as different conclusions can be drawn based on the choice of software. Today NONMEM is the defacto standard for PK/PD modelling and therefore a more detailed specification of the evaluation of the likelihood function would be valuable. This is however not likely to come about in near future as NONMEM is a commercial software based on proprietary code.

The actual use of PSM to build models has resulted in several pharmacokinetic models with different characteristics. The PK models have been modelled with insulin and/or C-peptide measurements. The models have shown how the variation can be split into population and individual parameters and also how the noise is divided into uncorrelated observation noise and correlated system noise (Wiener process).

The first models focused on extracting the insulin secretion rate (ISR) by deconvolution with the stochastic state space model. It was found to give reliable estimates of the ISR. However, the estimated confidence band did not seem entirely realistic since the confidence band was seen to be relatively wide at the end of the experiment where the ISR had stabilized. This effect is a result of ISR being modelled entirely as a random walk with variance proportional to time intervals. An intervention model was proposed to model the behavior of ISR with the diffusion term handling the deviations from this ISR model. The intension with the model was to reduce the confidence by modelling the mean of the process which was successfully achieved. However, the confidence bands were still relatively wide but it was found that the properties of the deconvolved random walk were much closer to a theoretical random walk. For this reason the latter confidence band may be more trustworthy.

A combined model for deconvolution of the liver extraction rate was proposed. The model used both insulin and C-peptide measurements to determine the ISR. Given the ISR, the liver extraction rate can be estimated in a non-linear model setup using insulin measurements. The analysis of the model has shown that deconvolution is possible also for non-linear models and it was found that the model for the liver extraction had to be constrained to be able to estimate the measurement noise.

Chapter 10

Conclusion

A functional prototype has been developed with the ability to handle stochastic differential equations in a mixed-effects setup. The maximum likelihood approach was used in parameter estimation on individual and population level. The Extended Kalman filter was used to create approximate conditional densities required for the likelihood function when working with SDEs.

The numerical implementation uses an accurate and robust standard Matlab ODE-solver for the prediction in the non-linear case. This choice was based on a compromise between speed and accuracy and resulted in a reliable model prediction for all tested PK/PD models.

The optimization of the maximum likelihood function was performed with a Quasi-Newton method and/or a pattern search. The parameter estimation was possible but a computational cumbersome task. It was found that noise in the objective function disturbed the parameter estimation but optimal parameter estimates were still able to be found and could be validated through profile plots.

The final prototype PSM was validated with CTSM and NONMEM. Since no standard solution exists for handling SDEs with mixed effects the validation was divided into separate parts. The SDE extension was validated with CTSM for a single subject and the population part was validated with NONMEM. The single subject validation was performed on a two-compartment model where the individual likelihood functions were compared with CTSM. The differences in likelihood values were within expectation of numerical differences. The comparison of the smoother for the linear and non-linear cases with CTSM showed that the results were also acceptably similar. The validation with NONMEM showed different approximate population likelihood (APL) values but it was found that the optimal parameters were almost identical. The difference in APL values resulted in different loglikelihood ratio tests which could result in different modelling results. It was not possible to come to any conclusion on what caused this difference since the NONMEM likelihood function is not publicly documented.

The prototype was used to analyse models within Pharmacokinetic modelling. The data set consisted of measurements of insulin and C-peptide from 12 subjects which made it ideal for mixed-effects modelling.

The insulin secretion rate was the focus of the first model which was used to form estimates of ISR by deconvolution. In this model ISR was assumed to be a random walk.

The second model included a priori knowledge on the meal times and durations in order to model the behavior of the ISR. This was achieved through a classic intervention model incorporated into the stochastic state space model. It was concluded that the ISR can be modelled well with this model and the diffusion term is able to account for deviations in ISR from the model.

A combined model for deconvolution of the liver extraction rate F was proposed. The model used both insulin and C-peptide measurements to determine the ISR. The initial model had F modelled as a pure random walk which resulted in a model where the insulin measurement noise and system noise on F could not be separated. A less flexible constrained F was introduced by adding another state to the model and redefining the derivative of F to the new state. The model with the constrained F produced more reliable results for the liver extraction and at the same time making it possible to estimate the insulin measurement noise. The liver extraction rate for one individual was estimated to be above one at certain times. This contradicts theory but is an effect of a shared elimination constant for all individuals. An interesting extension could be to estimate the elimination constant in a separate study and incorporate this into the model.

The overall conclusion based on the project is that a functional prototype incorporating SDEs into a non-linear mixed effects model is now available and it has been shown to be a flexible tool for working with models within PK/PD. Hopefully, the development of this program will continue in order to aid and guide in clinical drug development as well as in other areas.

Appendix

Appendix A

Matlab Implementation

The population model based on stochastic differential equations (SDE's) has been implemented in Matlab. It has been done both based on the Kalman Filter and Extended Kalman filter to be able to handle linear as well as non-linear models.

A.1 Considerations

The Matlab implementation has been based on a set of functions which all together are responsible for calculating population likelihood values by calling either `APL_KF` or `APL_EKF`. Figure A.2 and A.1 illustrates the functional relationship for the linear and non-linear version.

The idea behind the functional setup is to make the implementation simple to change and improve. It is thus simple to e.g. use another minimization algorithm for the individual optimizations or another ODE solver for the EKF, since the functions have a standard input/output interface.

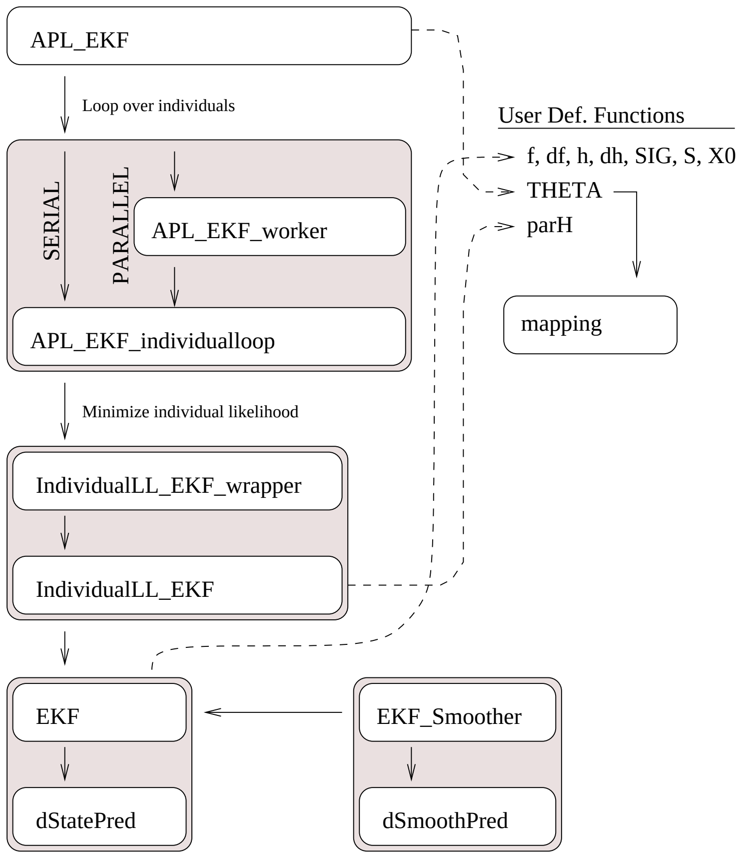


Figure A.1: Overview of functions for the non-linear model.

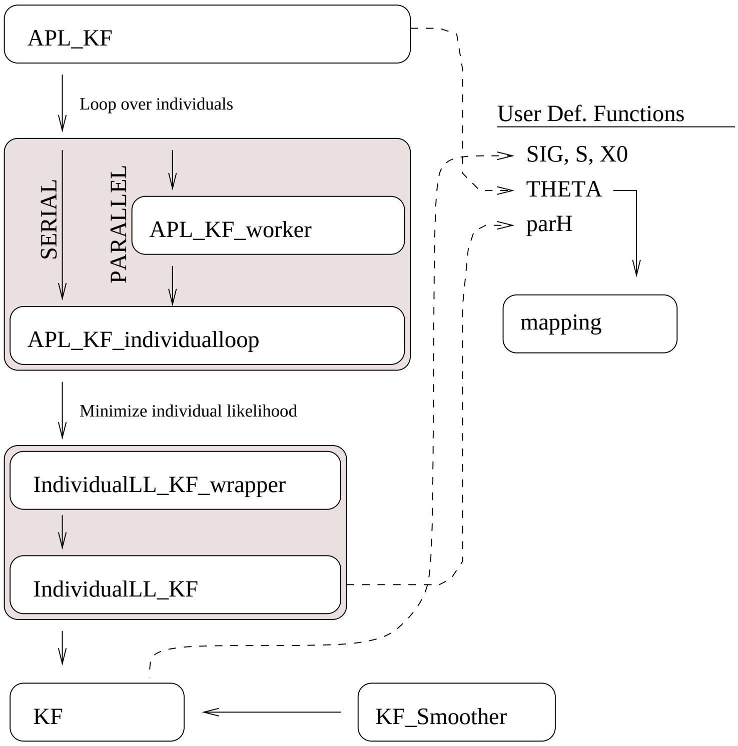


Figure A.2: Overview of functions for the linear model.

A.1.1 Non-linear case

The non-linear model has been implemented in the most general case, as specified below:

$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, t, \boldsymbol{\theta})dt + \boldsymbol{\sigma}(\boldsymbol{u}_t, t, \boldsymbol{\theta})d\boldsymbol{\omega}_t \tag{A.1}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k, \boldsymbol{\theta}) + \boldsymbol{e}_k \tag{A.2}$$

where $t \in \mathbb{R}$ is time, $\boldsymbol{x}_t$ is a vector in $\mathbb{R}^s$ of state variables, $\boldsymbol{u}_t$ is a vector in $\mathbb{R}^m$ of input variables, $\boldsymbol{y}_k$ is a vector in $\mathbb{R}^l$ of output variables, $\boldsymbol{\theta}$ is a vector of parameters, $\boldsymbol{f}(\cdot)$, $\boldsymbol{\sigma}(\cdot)$ and $\boldsymbol{h}(\cdot)$ are non-linear functions, $\{\boldsymbol{\omega}_t\}$ is a standard Wiener process and $\{\boldsymbol{e}_k\}$ is a white noise process with $\boldsymbol{e}_k \in N(\boldsymbol{0}, \boldsymbol{S}(\boldsymbol{u}_k, t_k, \boldsymbol{\theta}))$.

A.1.2 Linear case

The linear model is a special case of the non-linear model. It has been implemented to achieve high computation speed in Matlab, since the Kalman Filter can be solved explicitly without the use of an ODE solver. The speedup is easily a factor 100 or more. Also for the sake of computation speed it has been implemented as a time and partly parameter invariant linear model, which can be described as follows:

$$d\boldsymbol{x}_t = (\boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t)dt + \sigma(\boldsymbol{\theta})d\boldsymbol{\omega}_t \tag{A.3}$$

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}\boldsymbol{u}_k + e_k \tag{A.4}$$

where t is time, $\boldsymbol{x}_t$ is a state vector in $\mathbb{R}^n$, $\boldsymbol{u}_t$ is an input vector in $\mathbb{R}^m$, $\boldsymbol{y}_k$ is an output vector in $\mathbb{R}^l$, $\boldsymbol{\theta}$ is a vector of parameters, $\boldsymbol{A}$ is a matrix $\mathbb{R}^{n\times n}$, $\boldsymbol{B}$ is a matrix $\mathbb{R}^{n\times m}$, $\boldsymbol{\sigma}(\cdot)$ is a matrix $\mathbb{R}^{n\times n}$, $\boldsymbol{C}$ is a matrix $\mathbb{R}^{l\times n}$ and $\boldsymbol{D}$ is a matrix $\mathbb{R}^{l\times m}$, $\{\boldsymbol{\omega}_t\}$ is an n-dimensional standard Wiener process and $\{\boldsymbol{e}_k\}$ is an l-dimensional white noise process with $\boldsymbol{e}_k \in N(\boldsymbol{0}, \boldsymbol{S}(\boldsymbol{u}_k, t_k, \boldsymbol{\theta}))$.

The linear case has been implemented with zero-order hold on inputs, meaning that input is assumed constant between observations. In the current implementation it is only possible to use input if A has full rank.

A.2 Function specification

Throughout the implementation a number of standard data types has been used, which are listed in Table A.1.

Input	Name	Type	Default
THETA	Population param.	Vector	
phi/theta	Individual param.	Vector	
eta	Individual param.	Matrix $(v \times 1)$	
X	State	Matrix $(s \times 1)$	
Y	Observations	Matrix $(l \times n \times N)$	
Yi	Obs. for 1 ind.	Matrix $(l \times n)$	
Yk	Obs. for 1 ind. at t=k	Matrix $(l \times 1)$	
U	Input	Matrix $(m \times n \times N)$	
Ui	Input for 1 ind.	Matrix $(m \times n)$	
Uk	Input for 1 ind. at t=k	Matrix $(m \times 1)$	
T	Time	Matrix $(n \times 1)$	
t	1 timepoint	Scalar	
fktList	User def. functions	Cell array	
LB, UB	Bounds for THETA	$(p \times 1)$	No bounds
GUIFlag	Display progress		False
MPIFlag	Parallel computation		False

Table A.1: Overview of common input data types.

A.2.1 Non-linear

The non-linear model is setup by the user by defining a list of required functions. The user may choose the function names, and they must be listed in a cell array in the correct order shown in Table A.2. As it can be seen from the table, the program does not calculate the partial derivatives, but they must instead be derived and specified by the user.

Function number 8 and 9 are wrapper functions for the parameters. Function #8 takes in the population parameters defined in THETA and returns OMEGA (Ω) and theta. The parameters are mapped if lower and upper bounds are specified. theta contains the individual parameters η (set to NaN) and population parameters, exept for those used in OMEGA. Function #9 inserts eta values into theta, which is then called phi.

No.	Function	Input	Output	Reference
1	$\boldsymbol{f}$	X, U, t, phi	Matrix ($s \times 1$)	Eq. A.1
2	$\partial \boldsymbol{f}/\partial \boldsymbol{x}_t$	X, U, t, phi	Matrix ($s \times s$)	
3	$\boldsymbol{h}$	X, U, t, phi	Matrix ($l \times 1$)	Eq. A.2
4	$\partial \boldsymbol{h}/\partial \boldsymbol{x}_t$	X, U, t, phi	Matrix ($s \times s$)	
5	$\boldsymbol{\sigma}$	Uk, t, phi	Matrix ($s \times s$)	Eq. A.1
6	$\boldsymbol{S}$	Uk, t, phi	Matrix ($l \times l$)	Eq. A.2
7	$\boldsymbol{X}_0$	Uk,t,phi	Matrix ($s \times 1$)	
8	THETA	THETA, (LB, UB)	OMEGA, theta	
9	parH	eta, theta	phi	Eq. (4.1)

Table A.2: En tabel

APL	Approximate Population Likelihood
etaList	Estimated η's for each individual
optimStat	Infomation about individual optimizations

Table A.3:

A.2.1.1 APL_EKF

```
[APL,etaList,optimStat] =
  APL_EKF(THETA,Y,U,T,fktList,LB,UB,GUIFlag,MPIFlag)
```

This function calculates the Approximate Population Likelihood.

A.2.1.2 APL_EKF_worker

APL_EKF_worker is a script which is distributed to different CPUs for parallel computation of individual likelihood.

A.2.1.3 APL_EKF_individualloop

```
[LiPart_i eta_i optimStat_i] =
  APL_EKF_individualloop(theta,fktList,T,Yi,Ui,OMEGA,GUIFlag)
```

APL_EKF_individualloop is a function which computes the individual likelihood. It has been made to ensure that the serial and parallel version will perform the exact same calculations.

A.2.1.4 IndividualLL_EKF_wrapper

```
[Li GRAD] =
  IndividualLL_EKF_wrapper(eta,theta,fktList,T,Yi,Ui,OMEGA,GradStep);
```

IndividualLL_EKF_wrapper is a wrapper function for IndividualLL_EKF. It returns the function value and gradient to be used by an optimizer.

A.2.1.5 IndividualLL_EKF

```
Li = IndividualLL_EKF(eta,theta,fktList,T,Yi,Ui,OMEGA)
```

IndividualLL_EKF is a function, which runs the Extended Kalman Filter and calculates the Individual Log-Likelihood.

A.2.1.6 EKF

```
[LL,o] =
  EKF(phi,fktList,T,Y,U,LB,UB,GUIFlag)
```

EKF is function which returns the input filtered by the Extended Kalman Filter. The first output is the negative log-likelihood for a single individual model, which is not relevant for the population model. The second output is a struct array with all relevant computations, as shown i table A.4

EKF can handle missing observations stored as NaN. Missing observations are handled by setting the corresponding diagonal element in the observation covariance matrix to $10^{3}00$. This ensures correct prediction by EKF, but the likelihood function will not be correct. This is not a problem however, since the purpose of implementing missing observations is to be able to observe uncertainty between measurement timepoints, which is done after estimating the model.

A.2.1.7 dStatePred

```
dZ = dStatePred(t,Z,U,phi,fktList,dimX,Index)
```

LL	Negative log-likelihood
o	Output object
o.Yp	Space prediction
o.R	Space prediction covariance
o.Xf	Filtered state
o.Pf	Filtered state covariance
o.Xp	State prediction
o.Pp	State prediction covariance
o.KfGain	Kalman filter gain
o.sub_foreward	Used for smoothing

Table A.4: Output from EKF.

dStatePred is the function which is called by the ODE solver in EKF to predict the devellopment of the state and state covariance. They are solved simultaniously by introducing a new variable $\boldsymbol{Z} = (\boldsymbol{X} \;\; \boldsymbol{P}_u)$, where $\boldsymbol{P}_u$ is the vector containing the upper triangular elements of $\boldsymbol{P}$. This is done because $\boldsymbol{P}$ is symmetric, and it is thus only necessary to let the ODE solver work on one triangular part.

A.2.1.8 EKF_Smoother

```
[o Tsubs] = EKF_Smoother(phi,fktList,T,Y,U,subs)
```

EKF_Smoother computes smoothed estimates of the state. The input subs is a variable ≥ 0 which inserts missing observations between measurements, in order to easily let the EKF estimate the state and uncertainties between measurements. By specifying subs > 0 a number of missing observations are inserted linearly distributed between measurements, and the output Tsubs is the new time vector to be used for plotting etc.

The backward state prediction in the smoother requires $\boldsymbol{f}(\cdot)$ evaluated at the forward state estimate between observations. Since there is no fixed step length when solving the ODE's with an ODE solver, the forward filter estimate of the state is estimated by linear interpolation of the solution of the state development from the forward EKF. The ODE solver will automatically decrease the step length when there are fast changes in the solution, and therefore a linear interpolation of the solution should in general be close to the true development in the state. This technique enables a standard ODE solver to be used in both the forward and backward Kalman filter together with an easy estimate of the state prediction at any given timepoint.

A.2.1.9 dSmoothPred

```
dZ = dSmoothPred(t,Z,Uk,phi,fktList,dimX,Index,sub_foreward)
```

dSmoothPred is the function used by the ODE solver in EKF_Smoother for the backward filtering. It is similar in structure to dSmoothPred and also uses the trick with a vector containing the state and upper covariance matrix elements.

A.2.2 Linear

The linear model setup is almost identical to the non-linear model. The user must specify the some of the same functions as in the non-linear case. The system dynamics are given in the time and parameter invariant matrices A, B, C and D. Table A.5 gives an overview of the usersupplied functions.

No.	Function	Input	Output	Reference
1	$\boldsymbol{\sigma}$	phi	Matrix $(s \times s)$	Eq. A.1
2	$\boldsymbol{S}$	phi	Matrix $(l \times l)$	Eq. A.2
3	$\boldsymbol{X}_0$	phi	Matrix $(s \times 1)$	
4	THETA	THETA, (LB, UB)	OMEGA, theta	
5	parH	eta, theta	phi	

Table A.5: User defined functions for the linear model.

The linear functions are all similar to the non-linear, except from the fact that they also take A, B, C and D as input argument. The call to the main function that calculates the population log-likelihood value is looks like

```
[APL,etaList,optimStat] =
  APL_KF(THETA,fktList,A,B,C,D,T,Y,U,LB,UB,GUIFlag,MPIFlag)
```

A.2.2.1 KF_Smoother

```
[o Tsubs] =
  KF_Smoother(theta,fnk,A,B,C,D,T,Y,U,subs)
```

The smoother in the linear case uses an algorithm by Bryson & Fraiser [Kailath et al. 2000].

A.2.3 Simulation

```
[T,X,Y] =
  Simulation(fktList,phi,U,dt,SamplePoints)
```

Simulation is a function that based on the functions from the non-linear model setup is able to simulate data.

dt	Step length when solving the differential equation for the state.
SamplePoints	Sample points where observations are stored.

Table A.6: Input to Simulation.

To make the simulation accurate, a small *dt* must be chosen for subsampling the desired sample points. All sample points must be a product of dt.

The function Simulation returns T, $\boldsymbol{X} = \{\boldsymbol{X}_{t_i} | t_i \in T\}$, $\boldsymbol{Y} = \{\boldsymbol{Y}_{t_i} | t_i \in T\}$ and an output object containing the added Wiener and measurement noise.

A.2.4 Common functions

```
x = mapping(xtilde,LB,UB,inv)
```

The linear and non-linear versions share a common mapping function. The mapping function is used to to give bounds for parameters in the optimization of the likelihood function.

A mapping function is of the type $\Theta_i = f(X_i), \quad f : \mathbb{R} \rightarrow [L_i; U_i]$, where L_i and U_i are the lower and upper bound respectively for the i'th parameter. The use of a mapping function makes the standard unconstrained minimizer more robust since it is constrained from trying extreme extreme values of the parameters, e.g. a negative variance.

Two different mapping functions defined in Eq. A.5 and A.6 were considered. f_e is a logistic mapping and f_a is mapping based on the inverse tangent.

$$\theta_i = \quad f_e(X_i) = \frac{exp(X_i)U_i + L_i}{exp(X_i) + 1} \tag{A.5}$$

$$\theta_i = \quad f_a(X_i) = \frac{arctan(X_i) + \pi/2}{\pi}(U_i - L_i) + L_i \tag{A.6}$$

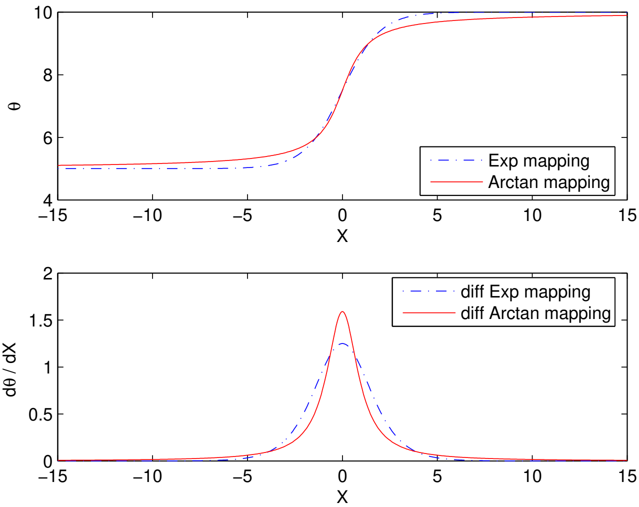


Figure A.3: Comparisson of logistic and arctan mapping.

The functions and their derivatives are plotted for $(L_i, U_i) = (5, 10)$ in Figure A.3. From the upper plot it is seen that f_a approaches its asymptotes slower than f_e. This is an advantage, because this means that the gradient with respect to X will not go toward zero as fast. Based on this analysis the mapping based on Arctan is chosen.

Appendix B

ISR

B.1 2 compartment C-peptide model

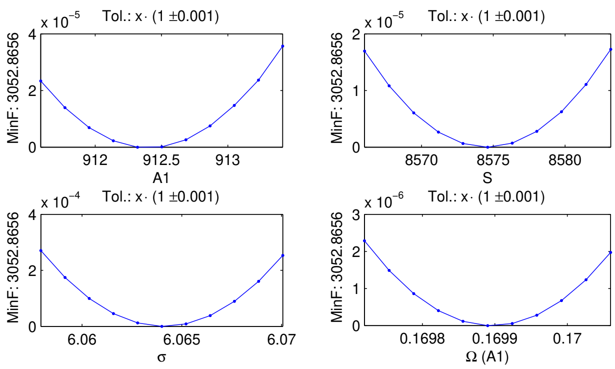


Figure B.1: APL for $\theta_{opt} \pm 5\%$.

B.2 2 compartment C-peptide log Model

Figure B.2 shows the APL curves for the estimated parameters in the Log Model. The plot shows that a reasonable minimum has be found. The width of the plot is ±5%.

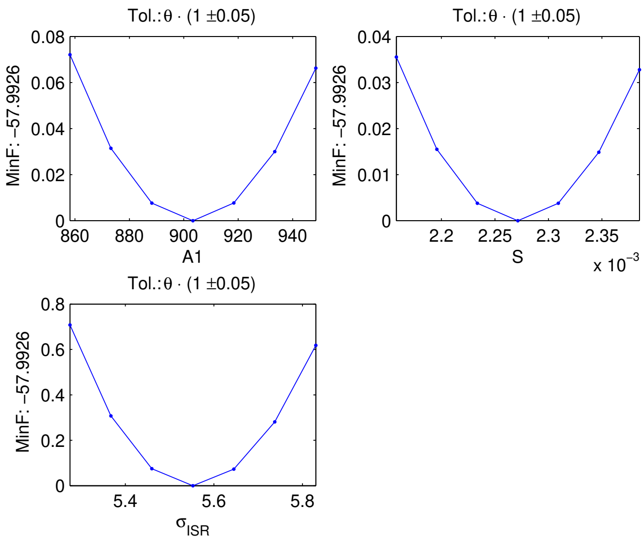


Figure B.2: APL for $\theta_{opt} \pm 5\%$.

–

Figure B.3 shows the log model 1-step predictions for the CPEP measurements. The predictions have been inverted from the logarithmic transformation by use of the exponential function.

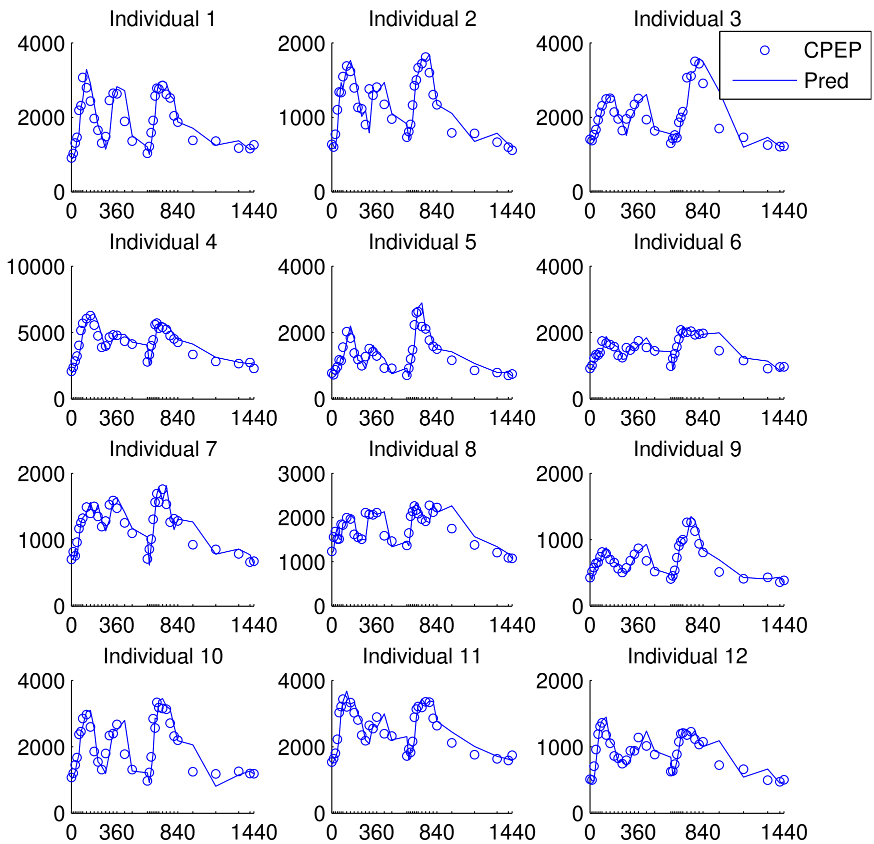


Figure B.3: Individual CPEP predictions.

B.3 Modelling of ISR

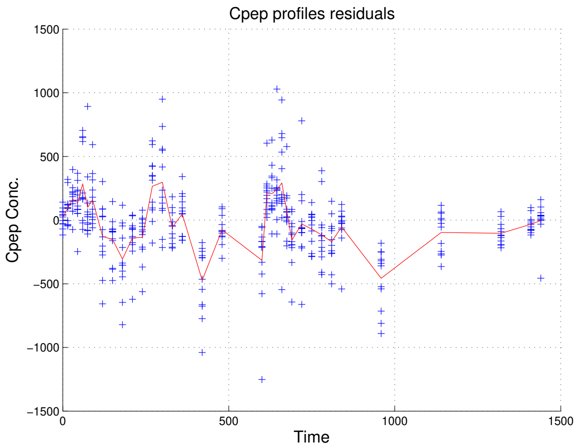


Figure B.4: Residuals plotted vs. time

In figure B.4 the model deficiency is more clearly seen. The model estimates too low when the Cpep concentrations are going up and when the concentration is coming down again the estimates tend to be too high. The prediction is based on the current summed ISR and predicted rise in the glucose levels just after a meal are not modelled.

The ISR profile is extracted as a mean of the summed ISR adjusted for the length of the time interval. This input will incorporate the knowledge of a known increase in cpeptide concentrations after meals.

The summed ISR's smoothed variance is shown in figure B.6. The variance is parameter specific an depends on the length of the time intervals.

B.3.1 Second Stage

The ISR profile is used as an input in the second model formulation.

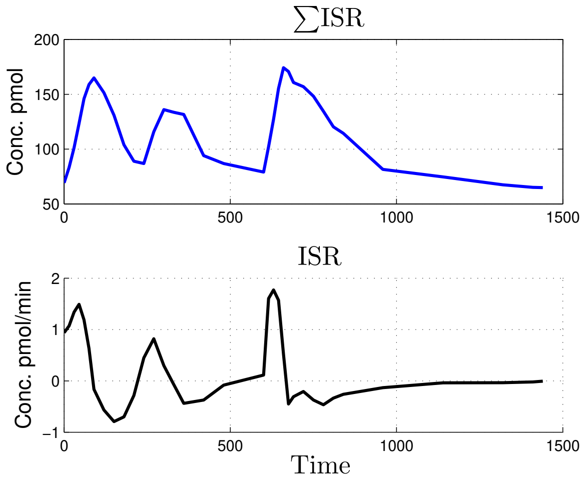


Figure B.5: $\sum$ISR and ISR profile

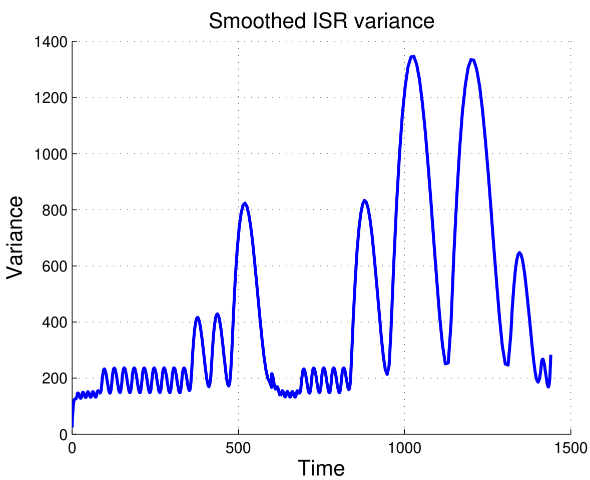


Figure B.6: ISR smoothed variance

Mean Input Model

Description: A 2 compartment model with the ISR profile as input.

Initial Conditions:

$$A_i = 900\, e^{\theta_1} \tag{B.1}$$

$$\boldsymbol{x}_0 = \begin{pmatrix} C_1 \\ C_2 \\ ISR \end{pmatrix} = \begin{pmatrix} A_i \\ \frac{k1}{k2} A_i \\ k_e A_i \end{pmatrix} \tag{B.2}$$

Model :

$$\begin{aligned} d\boldsymbol{x} &= \left(\begin{bmatrix} -(k_1+k_e) & k_2 & 1 \\ k_1 & -k_2 & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \cdot \\ \cdot \\ e^{\theta_2} \end{bmatrix} u_t \right) dt \\ &\quad + \begin{bmatrix} 1e^{-3} & & \\ & 1e^{-3} & \\ & & \theta_4 \end{bmatrix} d\boldsymbol{\omega} \end{aligned} \tag{B.3}$$

Output :

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \boldsymbol{x} + \epsilon, \qquad \text{where} \epsilon \in N(0, \theta_3) \tag{B.4}$$

The second stage model is almost identical to the first one except the population parameter in the starting point is changed into a individual scaling of the ISR input.

The prediction residuals are shown for the second model as for the first one in figure B.7. It is seen that there is no obvious trends in the residual plot.

The ACF for the prediction residuals from the second model can be seen in figure B.8. There is clear improvement in the auto correlation.

Finally the smoothed variance from the ISR profile is compared to the one from the first model.

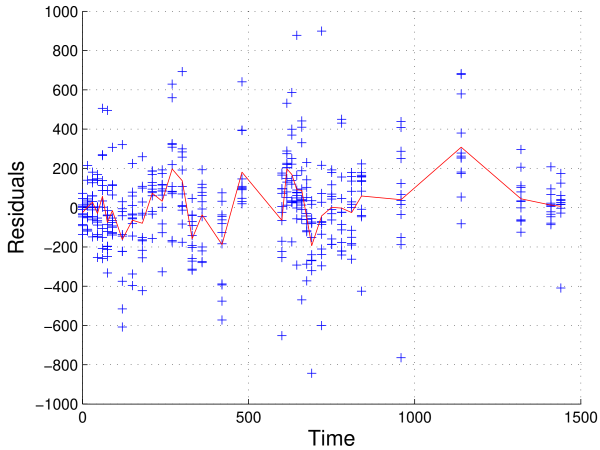


Figure B.7: Prediction residuals from the second model

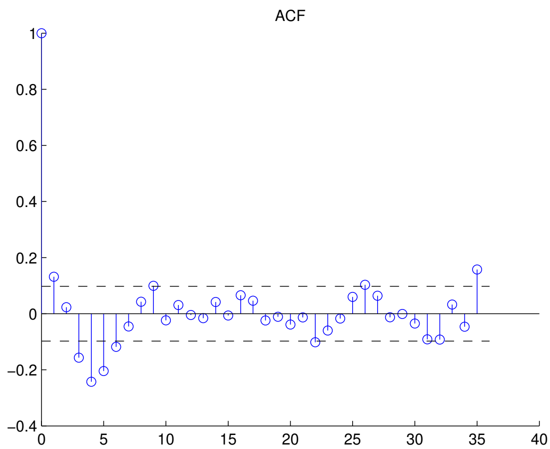


Figure B.8: ACF for the second model

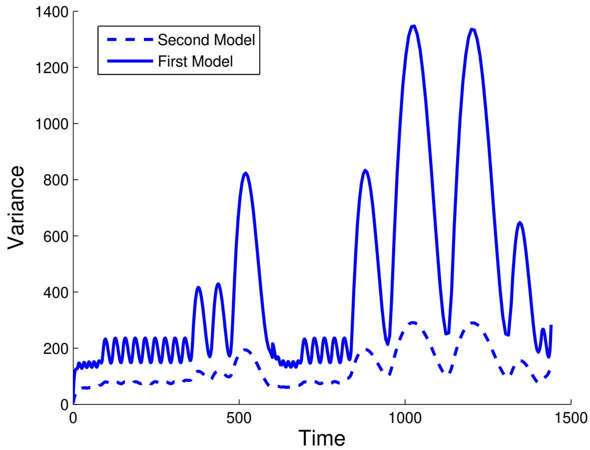


Figure B.9: Comparison of the smoothed variances

B.3.2 Extended Mean Input Model

The previous modelling of the insulin secretion rate has been a two stage model. There are statistical arguments against this kind of modelling as the data is used twice. The first stage is used to extract the mean ISR. This extracted ISR is then used as an input to a model that returns the ISR.

Instead of using the ISR from the first stage a ISR is constructed using mathematical formulaes. This curve is built on the knowledge on how the secretion rate should be. It could be considered a *A priori* knowledge.

The secretion of insulin goes up immediately the food is spotted. Just as the secretion of saliva goes up immediately. The maximum of the secretion is relative quickly archived and as the glucose levels in the blood goes down the secretion stops. The curve sought is a fast rising curve with a slower decrease. The χ^2-distribution is used as curve.

$$P_r(x) = \frac{x^{r/2-1}e^{-x/2}}{\Gamma(\frac{1}{2}r)2^{r/2}} \tag{B.5}$$

The formulae (B.5) can be found in [Conradsen 1999] or on the internet [1]. The

[1] http://mathworld.wolfram.com/Chi-SquaredDistribution.html

number r is the degrees of freedom in the density. It has been found that $r = 7$ gave a sequence as the secretion is believed to progress. The curve was moved and scaled by simple variable transformation and then subsampled according to the samplings in the dataset. It can be expected that 3 humps should be in the ISR as 3 meals were served during the trial. The created curves can be seen in figure (B.10).

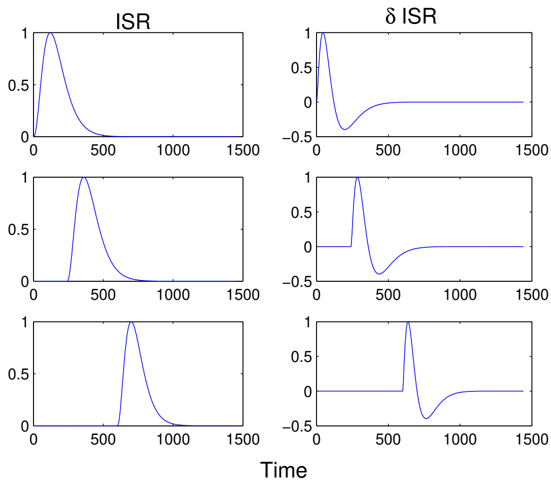


Figure B.10: Scaled χ^2-distributions and the derivative.

The 3 humps were subsampled inorder for them to be used as discrete input to the model.

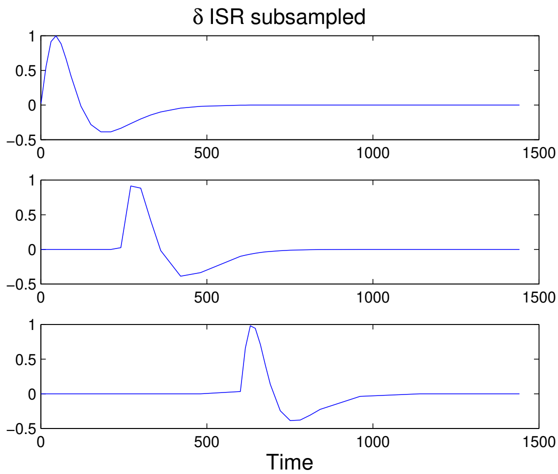


Figure B.11: Subsampled derivatives - Input to the model.

Extended Mean Input Model

Description: A standard 2 compartment model with individualized starting point in the initial conditions. The ISR over time intervals is modelled as a random walk with magnitude as parameter and the observation variance is a parameter as well. 3 dimensional input each scaled with individualized parameters.

Initial Conditions:

$$A_i = 900\exp(\eta_1) \tag{B.6}$$

$$\boldsymbol{x}_0 = \begin{pmatrix} C_1 \\ C_2 \\ ISR \end{pmatrix} = \begin{pmatrix} A_i \\ \frac{k1}{k2}A_i \\ k_e A_i \end{pmatrix} \tag{B.7}$$

Model :

$$d\boldsymbol{x} = \left(\begin{bmatrix} -(k_1+k_e) & k_2 & 1 \\ k_1 & -k_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} & & \\ & & \\ \eta_2 & \eta_3 & \eta_4 \end{bmatrix} \boldsymbol{u} \right) dt + \begin{bmatrix} 1e^{-3} & & \\ & 1e^{-3} & \\ & & \theta_2 \end{bmatrix} d\boldsymbol{\omega} \tag{B.8}$$

Output :

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \boldsymbol{x} + \epsilon, \qquad \text{where} \epsilon \in N(0, \theta_1) \tag{B.9}$$

B.3.2.1 Results

The model is used on the data set and the parameters are estimated using *ucminf*. The minimization were very time costly due to the 4 individualized parameters that has to be found for each subject in each population calculation. The residuals are analyzed again to give an indication of the model fit.

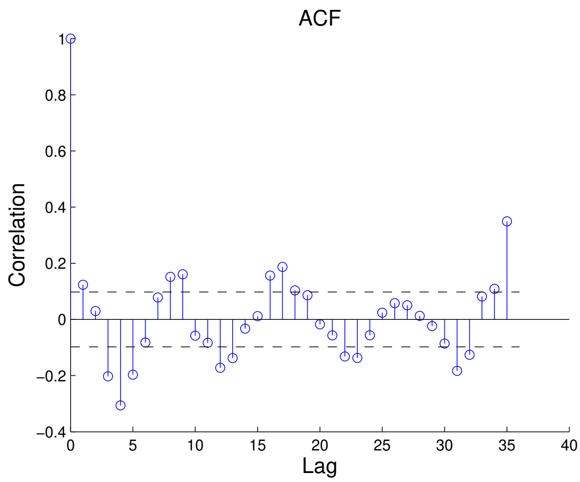


Figure B.12: Auto Correlation of the residuals from model.

The ACF in figure B.12 shows high resemblance to the ACF from the initial standard 2 compartment model. There is still clear trends and some variance to account for in the model.

A histogram and a QQ-plot was constructed to check if the residuals can be assumed to be normally distributed. The plots can be seen in figure B.13.

The a priori derivative of the ISR and the obtained states from the filtering are compared in figure B.14. The a priori derivative was constructed so it would mimic the filtered ISR. The a priori derivative is scaled accordingly to the individual parameters and hereafter integrated and has offset in the steady state solution for the ISR state. The difference between the 2 curves are the unmodelled secretion of Insulin which should be handled by the stochastic part

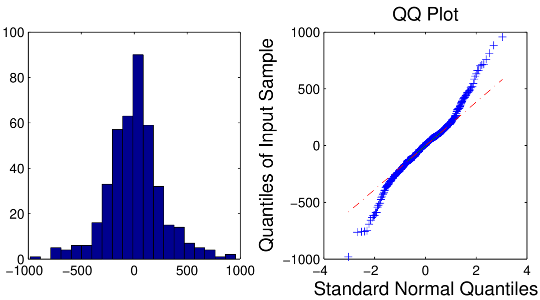


Figure B.13: Histogram and QQ plot of the residuals from Model 3

of the state space model. It can be seen that the ISR originating from the input peaks too late and should decrease faster. The scaling factor is disturbed by the slow decrease so the top is lowered to account for the long tails.

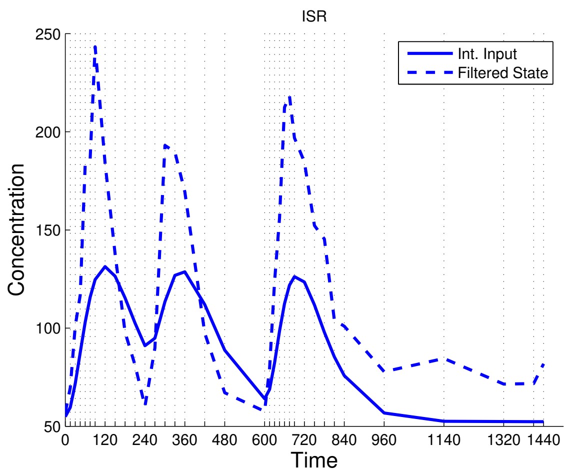


Figure B.14: The individualized summed input and the filtered ISR.

The smoothed variance is plotted along with the smoothed variance from the other 2 models in figure B.15. It can be seen that the two stage model is still the best choice based on the smoothed variances.

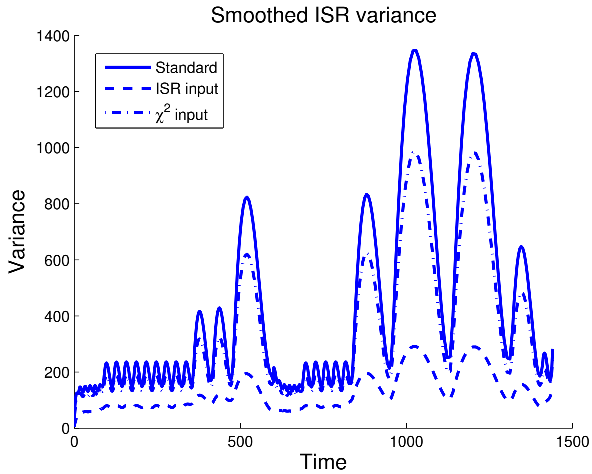


Figure B.15: Smoothed ISR from the 3 models.

B.3.3 Cross Validation

The two stage model has weakness in the data usage. The data is used twice and this could cause problems with the releability of the estimates. This is tested via cross validation. The data is split into 4 groups of 3 individuals. The parameters are now estimated on 9 individuals and used to predict for the remaining 3. The model used in the parameter estimation is the two compartment model with input. The during the training the input is scaled accordingly to the 9 training individuals and during prediction with the 3 validation individuals.

The 4 groups each provide 3 sets of prediction residuals which are augmented and analyzed.

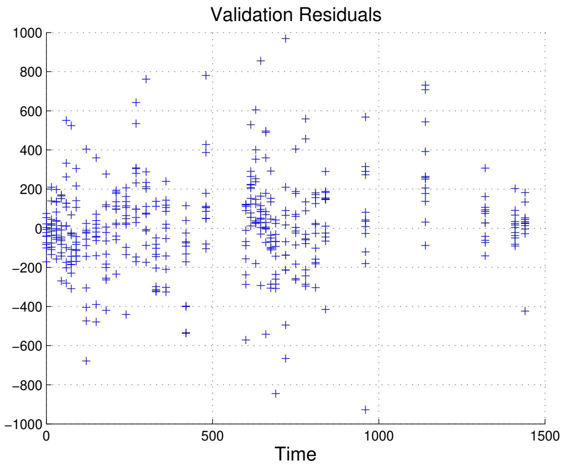


Figure B.16: Prediction residuals from validation groups.

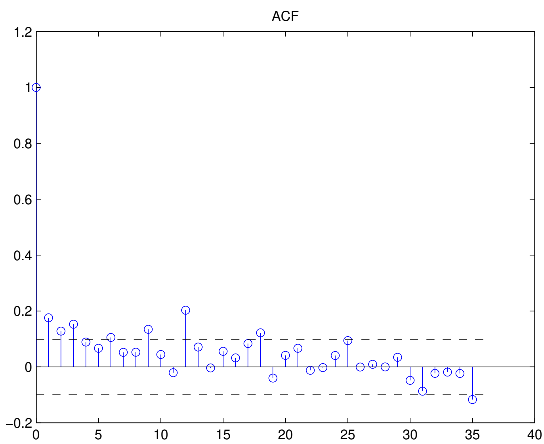


Figure B.17: ACF og the validation Prediction residuals.

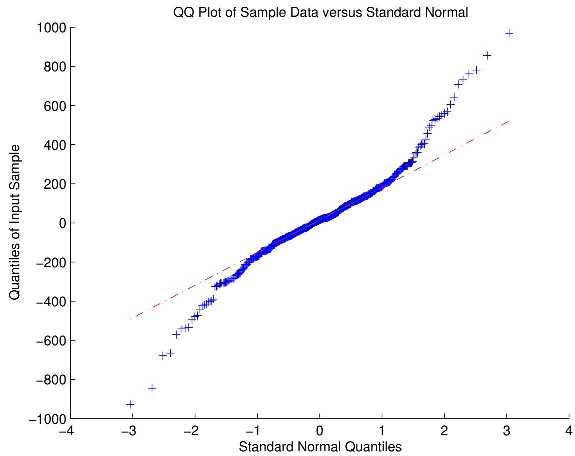


Figure B.18: QQ-Plot of the validation residuals.

B.4 Intervention Model

Figure B.19 shows the profile of the APL function for the found optimum of the 5 population parameters in the intervention model.

In figure B.20 the corresponding optimal ηs for the found optimal parameters are examined for the assumption that they are normally distributed by plotting them in QQ-plot.

In figure B.21 the predictions from the intervention model are plotted with the C-PEPTIDE observations.

In figure B.22 the model and smoothed ISR can be seen for all individuals.

In figure B.23 the smoothed estimates of ISR is compared for the Intervention Model and the Standard 2-compartment C-peptide Model. It should be seen that they are very similar.

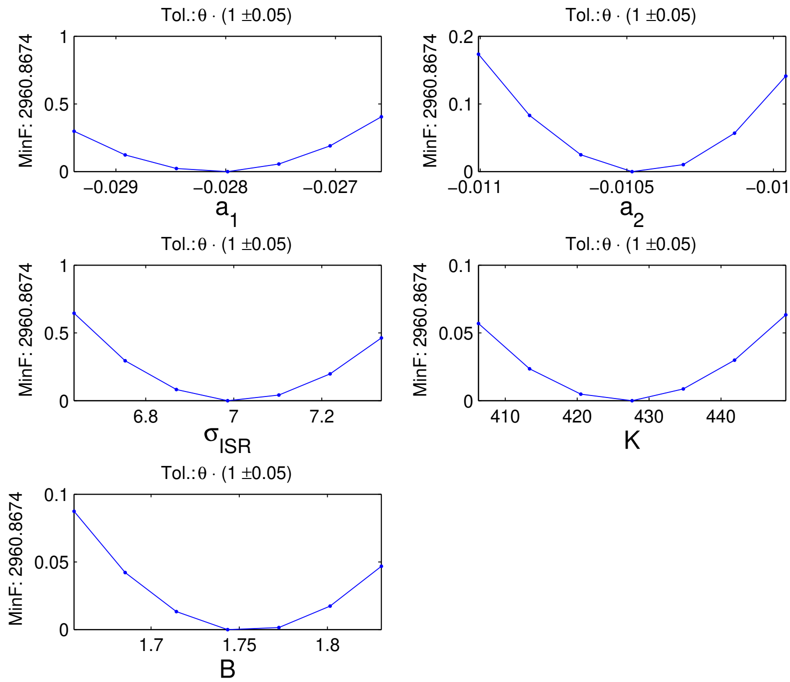


Figure B.19: Profile of APL for the found optimum in parameters.

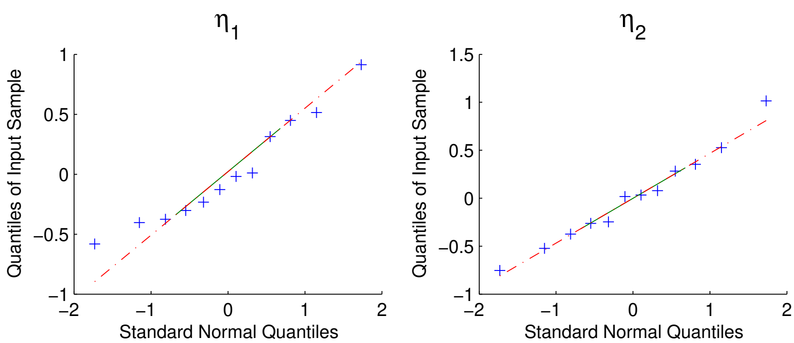


Figure B.20: QQ-plot of the found optimal etas..

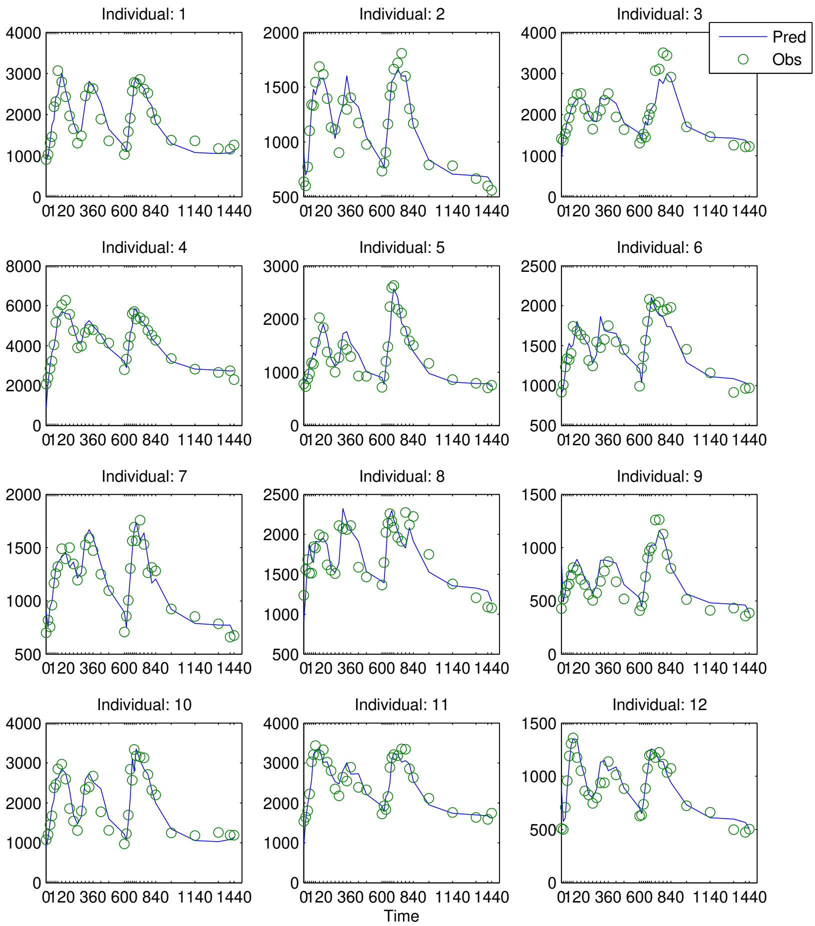


Figure B.21: Predictions and C-PEPTIDE observations for each of the 12 individuals.

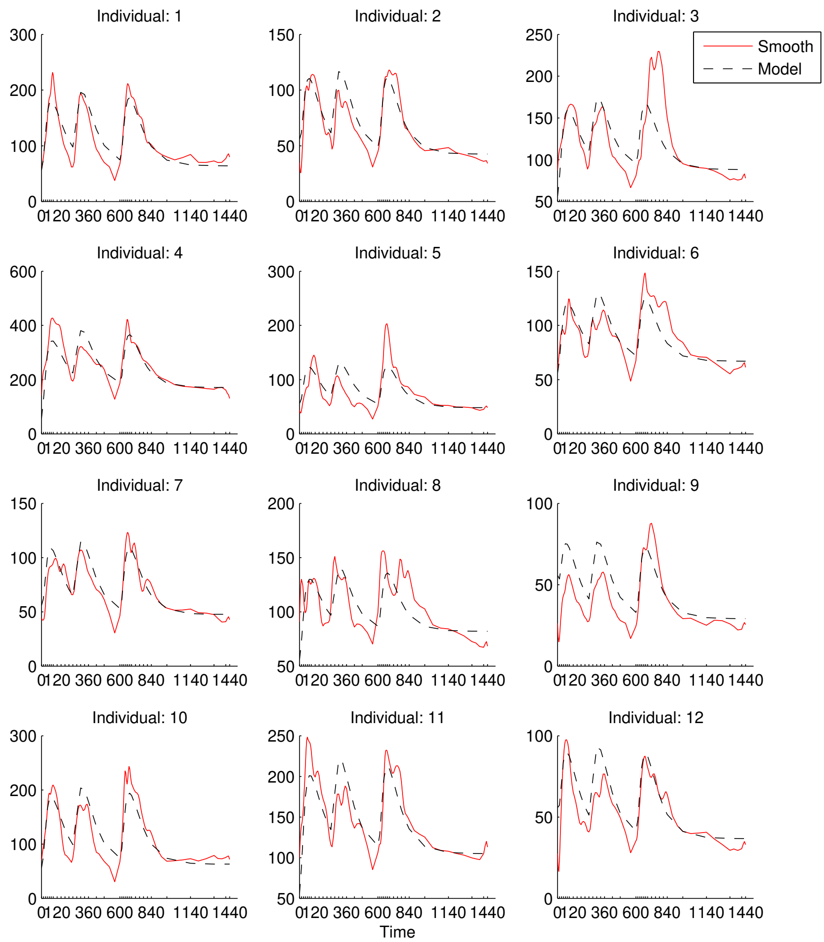


Figure B.22: Model and smoothed ISR for all individuals.

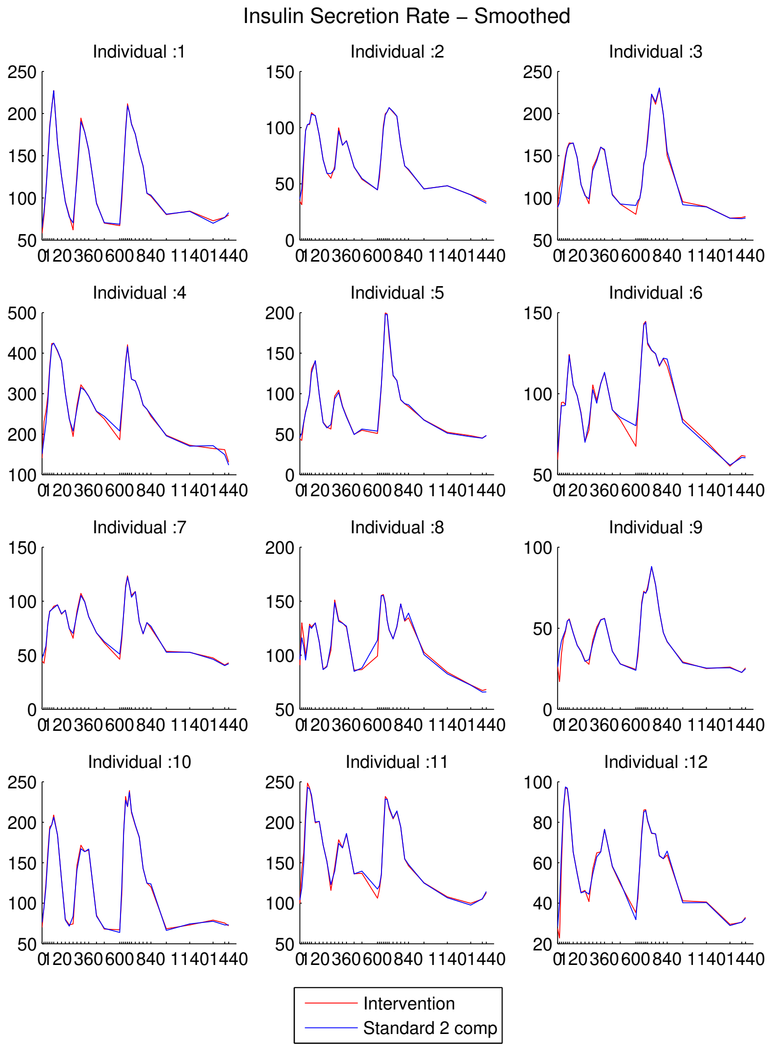


Figure B.23: Smoothed ISR for the standard 2-comp. and intervention model.

B.5 Model validation for the combined model

Figure B.24 shows residual analysis for the combined model with 2D-data (C-peptide and Insulin) found in Section 7.4.1. Figure B.25 shows the 1-step predictions for the model for all 12 individuals. Figure B.26 verifies that the correct optimum of the likelihood function has been found.

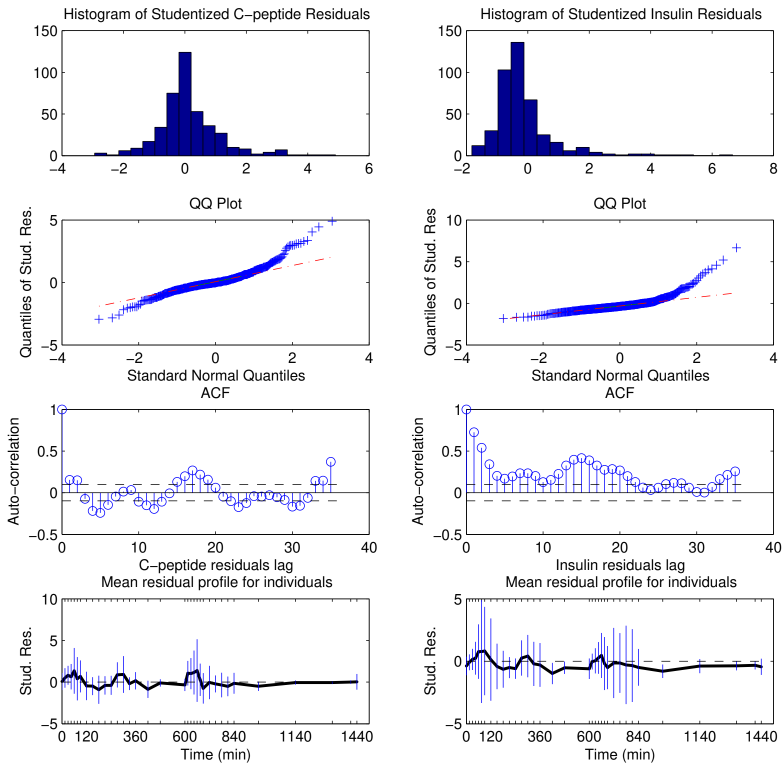


Figure B.24: Residual analysis for C-Peptide and Insulin measurements.

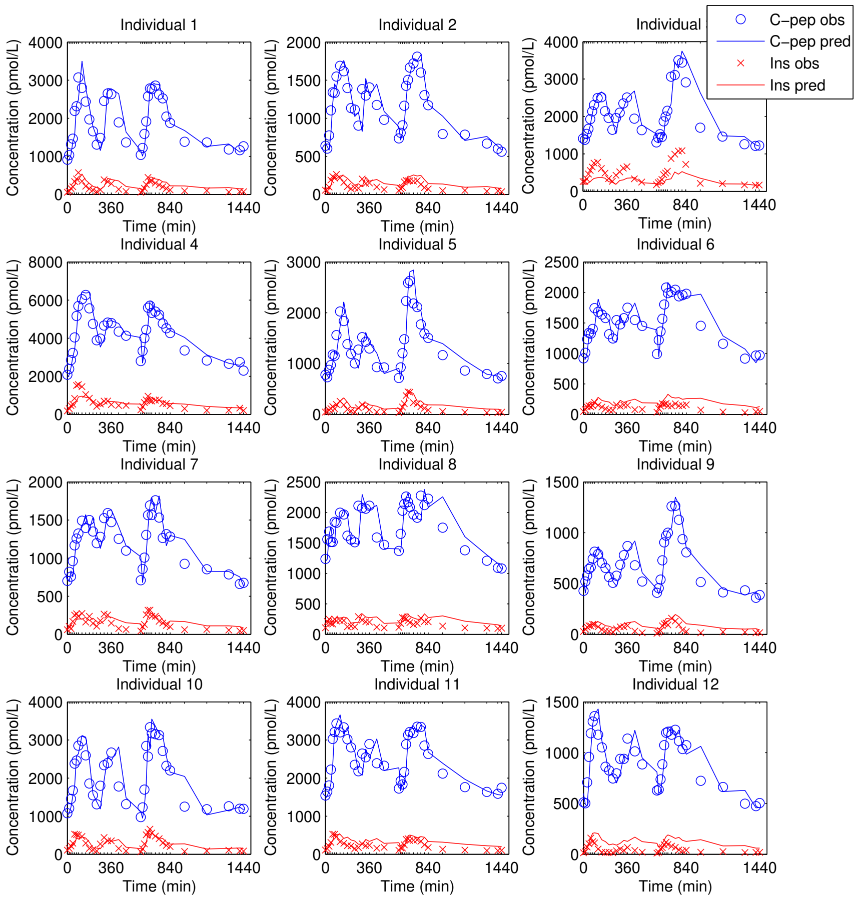


Figure B.25: 1-step predictions for C-peptide and Insulin measurements using the combined model.

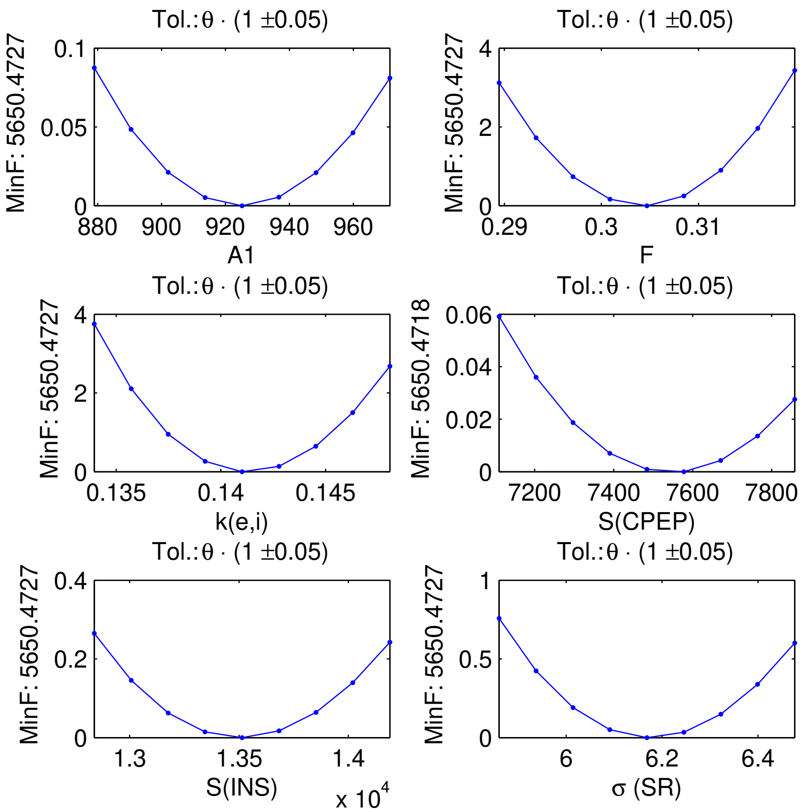


Figure B.26: Likelihood function plotted $\pm 5\%$ of optimal parameters.

B.6 Liver Extraction Model

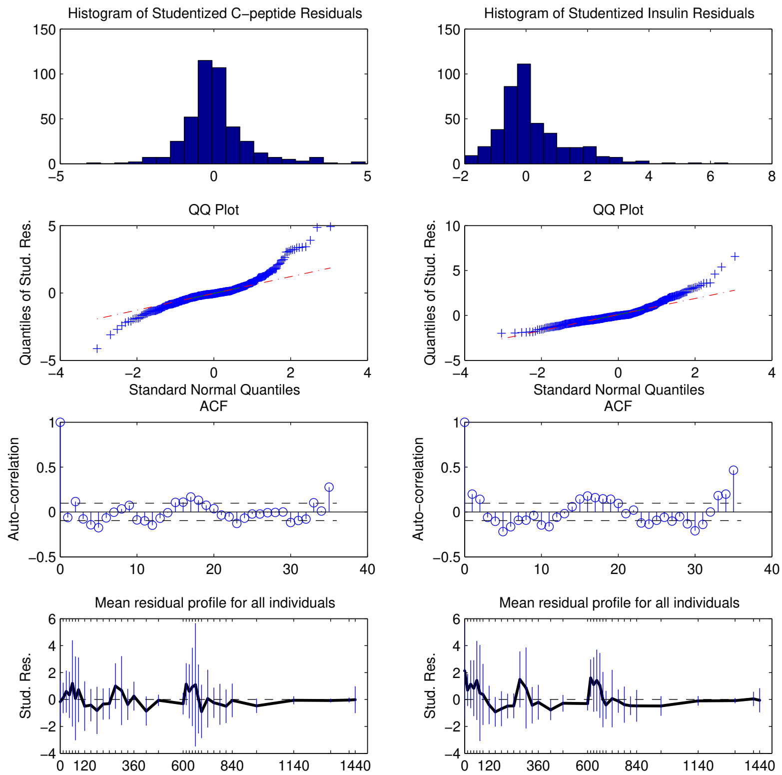


Figure B.27: Residual analysis for C-Peptide and Insulin measurements for the Liver Extraction Model.

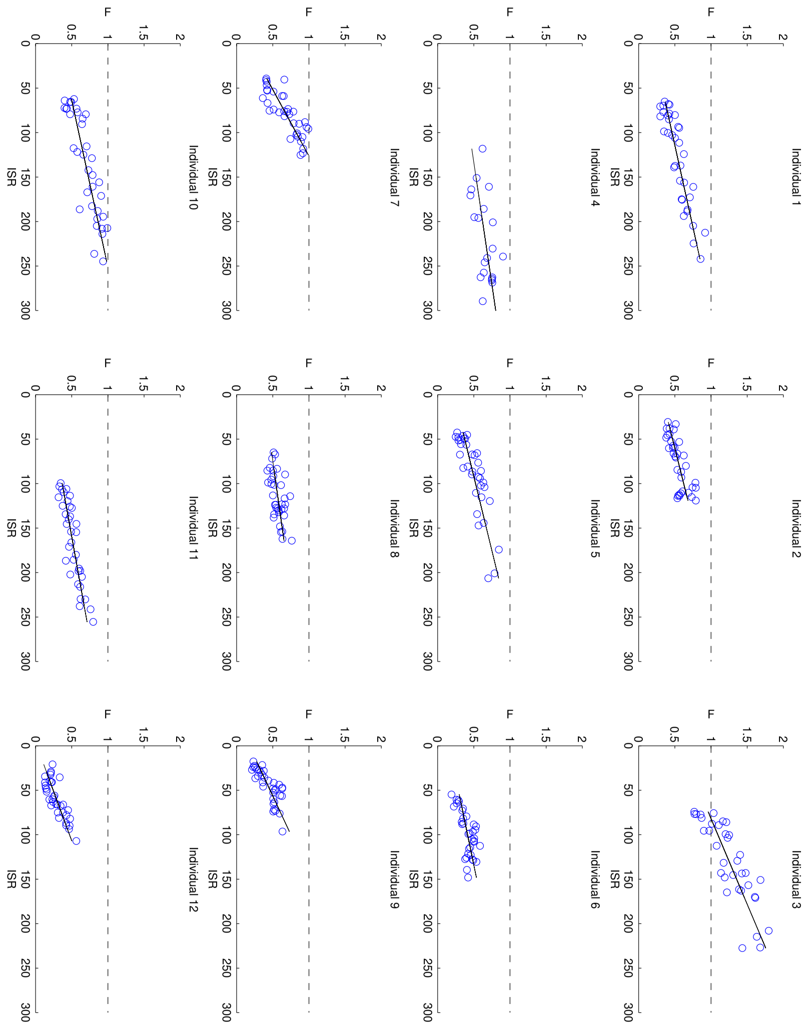


Figure B.28: Relation between F and ISR for all 12 individuals for the Liver Extraction Model.

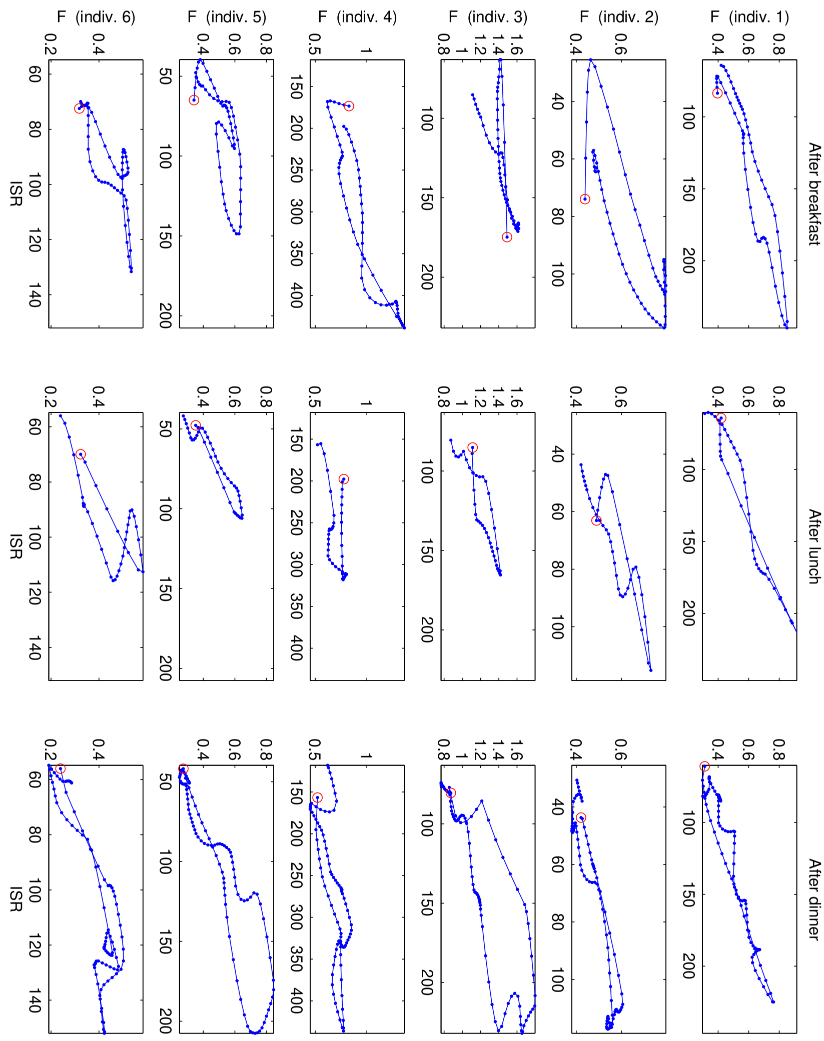


Figure B.29: Phase plot of F and ISR for 6 individuals. Smoothed estimates of F and ISR is used with 5 subsamples bestween each measurement.

B.7 Constrained Liver Extraction Model

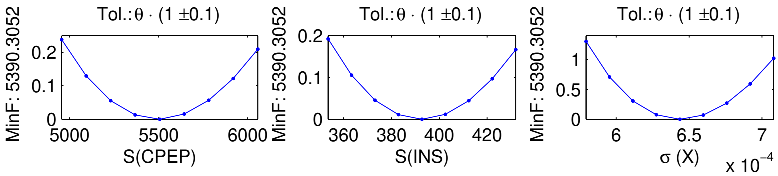


Figure B.30: Likelihood profiles for the Constrained Liver Extraction Model.

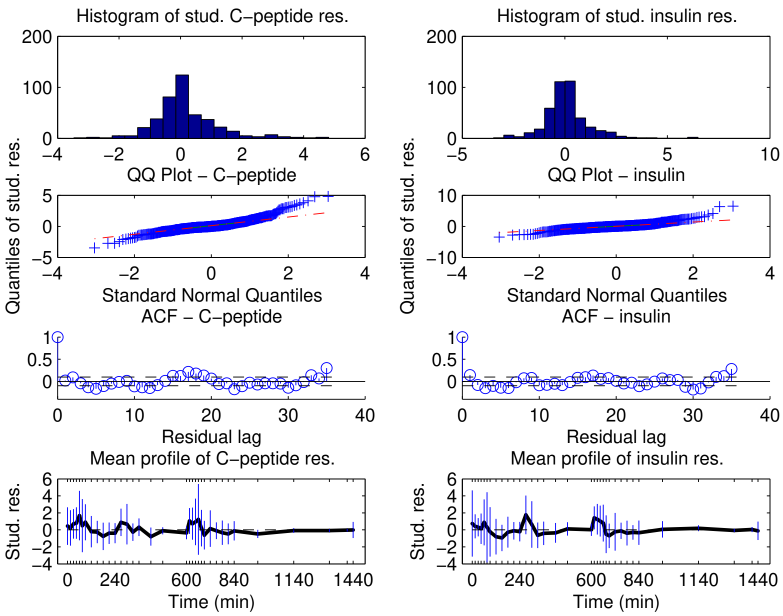


Figure B.31: Residual analysis for the Constrained Liver Extraction Model.

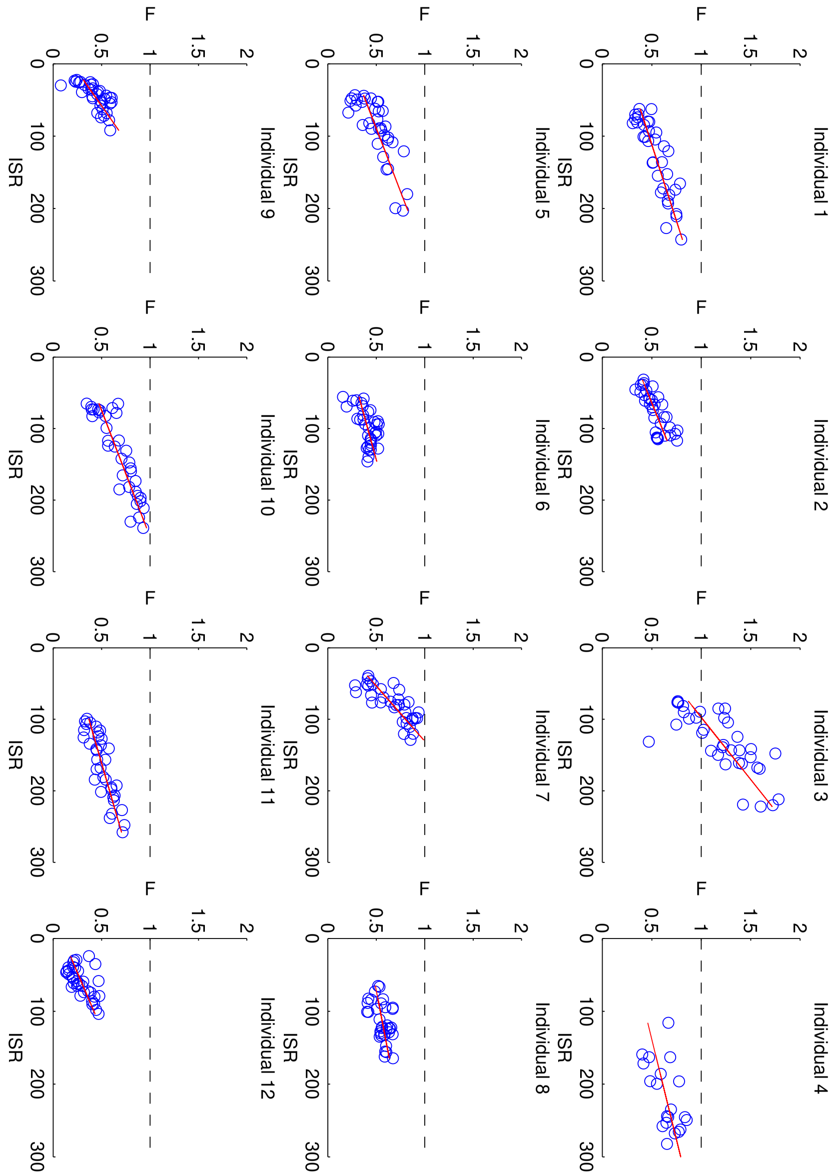


Figure B.32: Relation between F and ISR for all 12 individuals for the Constrained Liver Extraction Model. Red line is individually fitted slope.

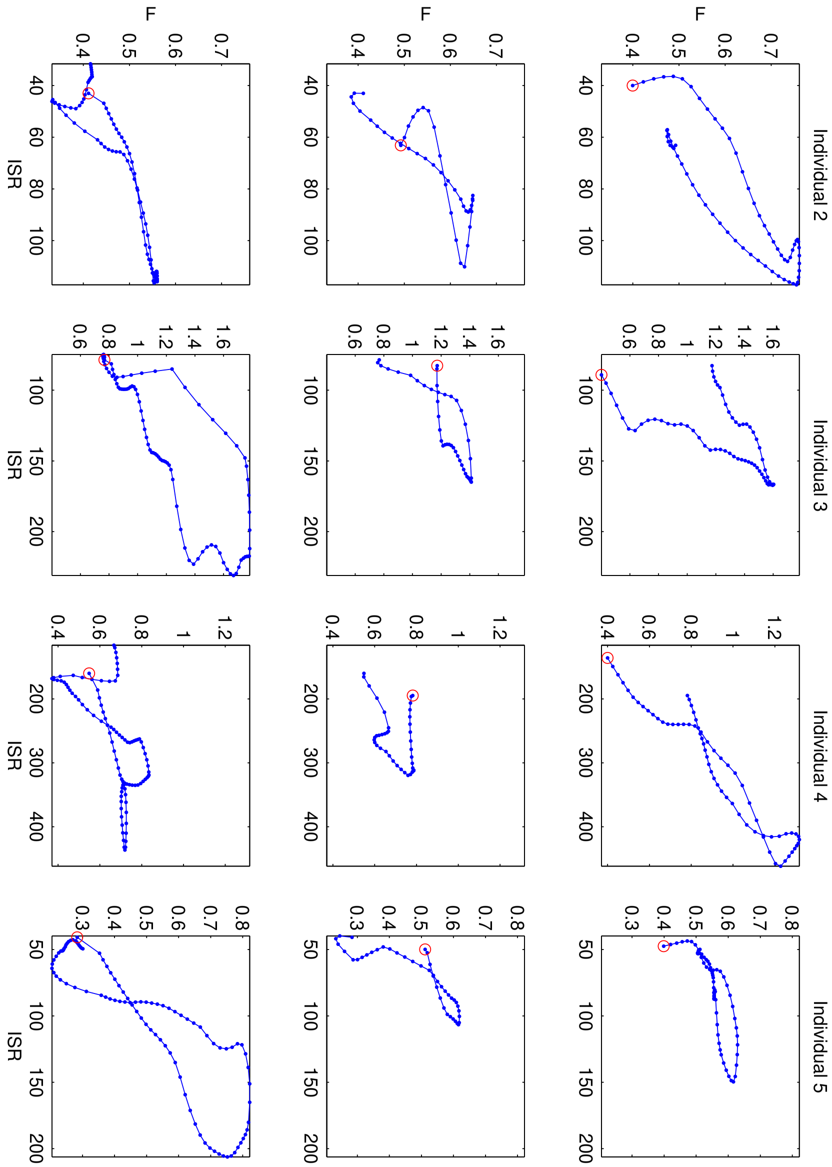


Figure B.33: Phase plot of F and ISR for 4 individuals. Smoothed estimates of F and ISR is used with 5 subsamples bestween each measurement.

Bibliography

Bickel, P. J. & Doksum, K. A. (1976), *Mathematical statistics*, Holden-Day Inc., San Francisco, Calif. Basic ideas and selected topics, Holden-Day Series in Probability and Statistics.

Cauter, E. V., Mestrez, F., Sturis, J. & Polonsky, K. (1992), 'Estimation of insulin secretion rates from c-peptide levels. comparison of individual and standard kinetic parameters for c-peptide clearance.', *Diabetes* **41**(3), 368–77.

Conradsen, K. (1999), *En Introduktion til Statistik*, IMM.

Dennis, J. J. E. & Schnabel, R. B. (1983), *Numerical methods for unconstrained optimization and nonlinear equations*, Prentice Hall Series in Computational Mathematics, Prentice Hall Inc., Englewood Cliffs, NJ.

FDA (1999), 'Guidance for industry - population pharmacokinetics', web. http://www.fda.gov/cber/gdlns/popharm.htm.

Frandsen, P. E., Jonasson, K., Nielsen, H. B. & Ingleff, O. (2004), Unconstrained optimization, Teaching material in the cource Optimization and Datafitting at DTU.

Gabrielsson, J. & Weiner, D. (1997), *Pharmacokinetic and Pharmacodynamic Data Analysis: Concepts and Applications*, second edn, Kristianstads Boktryckeri.

Gelb, A., Joseph F. Kasper, J., Raymond A. Nash, J., Price, C. F. & Arthur A. Sutherland, J. (1982), *Applied Optimal Estimation*, seventh edn, The MIT Press.

Jazwinski, A. H. (1970), *Stochastic processes and filtering theory*, Vol. 64, Academic Press., New York.

Kailath, T., Sayed, A. H. & Hassibi, B. (2000), *Linear Estimation*, Prentice-Hall.

Kalman, R. (1960), 'New approach to linear filtering and prediction problems', *American Society of Mechanical Engineers – Transactions – Journal of Basic Engineering Series D* **82**(1), 35–45.

Kalman, R. E. & Bucy, R. S. (1961), 'New results in linear filtering and prediction theory', *Trans. ASME Ser. D. J. Basic Engrg.* **83**, 95–108.

Kjems, L. L., Vølund, A. & Madsbad, S. (2001), 'Quantification of beta-cell function during ivgtt in type ii and non-diabetic subjects: assessment of insulin secretion by mathematical methods', *Diabetologia* **44**, 1339–1348.

Kloeden, P. E. & Platen, E. (1992), *Numerical solution of stochastic differential equations*, Vol. 23 of *Applications of Mathematics (New York)*, Springer-Verlag, Berlin.

Kristensen, N. R. & Madsen, H. (2003), Continuous time stochastic modelling - ctsm 2.3 mathematics guide, Technical report, Technical University of Denmark.

Kristensen, N. R., Madsen, H. & Ingwersen, S. H. (2004), A deconvolution method for linear and nonlinear systems based on stochastic differential equations, poster presented at: Population Approach Group in Europe (PAGE), 13th meeting.

Kristensen, N. R., Madsen, H. & Ingwersen, S. H. (2005), 'Using stochastic differential equations for pk/pd model development', *Journal of Pharmacokinetics and Pharmacodynamics* **32**, 109–41.

Øksendal, B. (1995), *Stochastic Differential Equations - An Introduction with Applications*, fourth edn, Springer.

Overgaard, R. V., Jonsson, N., Tornoe, C. W. & Madsen., H. (2005), 'Non-linear mixed-effects models with stochastic differential equations: implementation of an estimation algorithm.', *Journal of Pharmacokinetics and Pharmacodynamics* **32**(1), 85–107.

Pharsight (2004), *WinNonlin User Manual version 3.1*, deconvolution edn, Pharsight.

Racine-Poon, A. & Wakefield, J. (1998), 'Statistical methods for population pharmacokinetic modelling', *Statistical Methods in Medical Research* **7**(1), 63.

Singer, H. (2002), 'Parameter estimation of nonlinear stochastic differential equations: simulated maximum likelihood versus extended Kalman filter and Itô-Taylor expansion', *J. Comput. Graph. Statist.* **11**(4), 972–995.

Sparacino, G., Pillonetto, G., Capello, M., Nicolao, G. D. & Cobelli, C. (2002), 'Winstodec: A stochastic deconvolution interactive program for physiological and pharmacokinetic systems', *Computer Methods and Programs in Biomedicine* **67**, 67–77.

Sundheds-ministeriet (2003), Handlingsplan for diabetes, www.im.dk.

Thyregod, P. & Madsen, H. (2004), *An Introduction to General and Generalized Linear Models*, 0.9 edn, IMM, Lyngby.

Tornøe, C. W. (2005), Population pharmacokinetic/pharmacodynamic modelling of the hypothalamic-pituitary-gonadal axis, PhD thesis, IMM - DTU.

Warberg, J. (1991), *Human Fysiologi - En grundbog*, second edn, Polyteknisk Forlag.

Welch, G. & Bishop, G. (2004), An introduction to the kalman filter, Department of Computer Science - University of North Carolina at Chapel Hill.

WHO (2006), World health organisation, country and regional data, http://www.who.int/diabetes/facts/world_figures/en/.