



FACULTEIT
BIO-INGENIEURSWETENSCHAPPEN



Academiejaar 2004-2005

OPTIMAL EXPERIMENTAL DESIGN FOR CALIBRATION OF BIOPROCESS MODELS: A VALIDATED SOFTWARE TOOLBOX

OPTIMAAL EXPERIMENTEEL ONTWERP VOOR KALIBRERING VAN BIOPROCESMODELLEN: EEN GEVALIDEERDE SOFTWARE TOOLBOX

door

ir. Dirk DE PAUW

*Thesis submitted in fulfillment of the requirements
for the degree of Doctor (PhD) in Applied Biological Sciences*

*Proefschrift voorgedragen tot het bekomen van de graad
van Doctor in de Toegepaste Biologische Wetenschappen*

op gezag van

Rector: **Prof. Dr. A. DE LEENHEER**

Decaan:

Prof. Dr. ir. H. VAN LANGENHOVE

Promotor:

Prof. Dr. ir. P. VANROLLEGHEM

OPTIMAL EXPERIMENTAL DESIGN FOR CALIBRATION OF BIOPROCESS MODELS: A VALIDATED SOFTWARE TOOLBOX

by

ir. Dirk DE PAUW

PhD Review Committee

Dr. E. Ayesa (CEIT, Spain)

Prof. Dr. ir. D. Dochain (Université Catholique de Louvain)

Prof. Dr. ir. S. Marsili-Libelli (University of Florence, Italy)

Prof. Dr. ir. P. Vanrolleghem (Ghent University, promoter)

Prof. Dr. ir. N. Verhoest (Ghent University, secretary)

Other Jury Members

Prof. Dr. ir. G. Hofman (Ghent University, chair)

Prof. Dr. ir. J. Pieters (Ghent University)

Prof. Dr. ir. O. Thas (Ghent University)

ISBN 90-5989-059-0

02.05.2005

The author and promoter give the authorization to consult and to copy parts of this work for personal use only. Every other use is subject to the copyright laws. Permission to reproduce any material contained in this work should be obtained from the author.

De auteur en de promotor geven de toelating dit werk voor consultatie beschikbaar te stellen, en delen ervan te kopiëren voor persoonlijk gebruik. Elk ander gebruik valt onder de beperkingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting uitdrukkelijk de bron te vermelden bij het aanhalen van resultaten uit dit werk.

De promotor

De auteur

Prof. Dr. ir. P. Vanrolleghem

ir. D. De Pauw

Preface

Doctoreren, iets meer dan 4 jaar geleden leek het de droomoplossing: vijf jaar gestudeerd en nog niet veel goesting om te gaan werken, dus nog even uitbollen op de universiteit... Dat is dan wel even anders uitgedraaid moet ik zeggen. Als ik er zo nu op terugkijk is het inderdaad de droomoplossing geweest maar dan om heel andere redenen. De belangrijkste is wellicht dat ik enorm veel bijgeleerd heb. En daar bedoel ik niet enkel mee op wetenschappelijk vlak maar ook op persoonlijk vlak. De laatste 4 jaar hebben me leren inzien wat belangrijk is in het leven, waarin ik echt geïnteresseerd ben, waar mijn grenzen liggen en waar ik naartoe wil. Op wetenschappelijk vlak is er natuurlijk ook heel wat te leren geweest. Ik moet zeggen dat ik mijn niche gevonden heb: daar waar informatica, modelleren en milieu samenkomen. Drie dingen die me altijd al sterk hebben aangetrokken. Ik ben, vrees ik :-), een echte wetenschappelijk “hobbyist” geworden.

Een doctoraat komt heus niet vanzelf tot stand. Er zijn heel wat mensen die bewust en onbewust bijgedragen hebben tot dit werk. Iedereen bedankt daarvoor. Toch zou ik enkele mensen even speciaal willen vermelden.

In de eerste plaats mijn promotor Peter Vanrolleghem. Omwille van de vlotte omgang en communicatie die ik met hem had heb ik hem altijd meer gezien als een briljante collega dan als “baas”. Dat “briljante” heeft dan betrekking op de talloze vergaderingen die we gehad hebben (die welliswaar niet altijd op tijd begonnen): met 2 vragen naar binnen gaan, met 2 antwoorden naar buiten komen samen met nog 10, soms “crazy”, extra ideeën. Maar het zijn dan net die “crazy” ideeën die zich soms ontpoppen tot hele hoofdstukken. Ik zou hem ook enorm willen bedanken voor de vrijheid die hij mij gegeven heeft tijdens mijn onderzoek om mijn eigen ding te doen. Wellicht heeft hij zich dat soms wel wat beklaagd wanneer ik weer vanuit het niets met iets volledig uitgewerkt en geïmplementeerd aan kwam draven. Daarnaast apprecieer ik het ook enorm dat ik de kans gekregen heb om aan een aantal congressen te hebben mogen deelnemen. Dat was altijd een ervaring, op wetenschappelijk vlak, maar zeker ook op persoonlijk vlak.

Dan is er natuurlijk de BIOMATH omgeving nog. Welke onderzoeksgroep

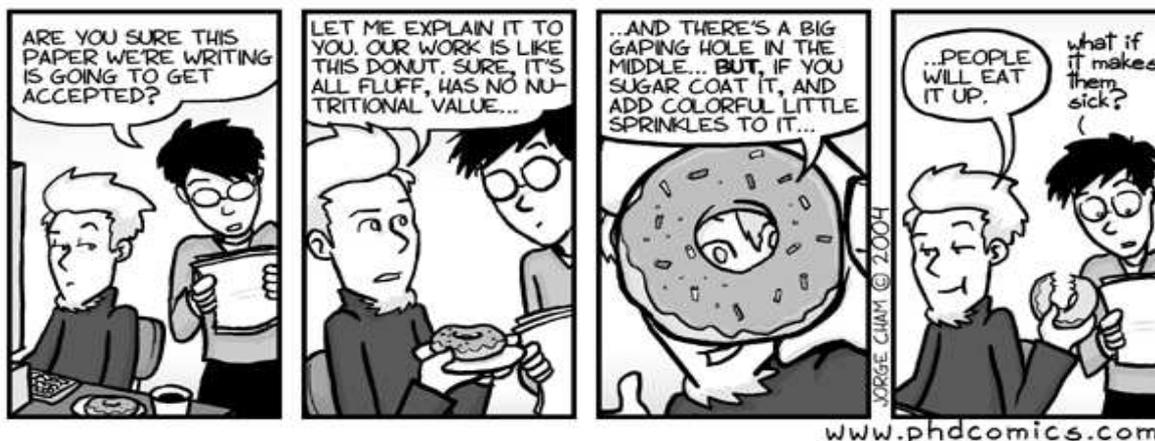
kan zeggen dat er jaarlijks weerkerende evenementen zoals fuiven, weekendjes en cocktails worden georganiseerd. Niet veel denk ik. En dat zegt genoeg over de sfeer die hier heerste de afgelopen jaren. Ik zou hiervoor alle collega's willen bedanken maar natuurlijk ook voor de interessante wetenschappelijke samenwerking en talrijke leuke discussies. De BIOMATH fuif van het jaar 2001 was wellicht één van de belangrijkste gebeurtenissen uit mijn leven tot nu toe (zij die er toen bij waren weten wel waarom). Hetzelfde kan echter niet gezegd worden van de fuif in 2000, want, ik moet eerlijk zijn, die herinnering is nog altijd één groot zwart gat :-).

Wie ook niet mag ontbreken zijn mijn ouders. Ik zou ze allereerst willen bedanken omdat ze hun genetisch materiaal zodanig gecombineerd hebben zodat ik daar het resultaat van geworden ben. Maar, een mens is natuurlijk meer dan het genetische materiaal. Daarom zou ik hen ook willen bedanken voor de manier waarop ze mij hebben opgevoed, voor de vrijheid, steun en het vertrouwen dat ik altijd gekregen heb in al mijn beslissingen die ertoe geleid hebben dat ik dit werk tot een goed einde heb kunnen brengen.

Als laatste zou ik nog iemand willen bedanken die me ontzettend nauw aan het hart ligt, mijn Kathyke. Ik mag me ontzettend gelukkig prijzen iemand ontmoet te hebben waarmee ik zo ontzettend op dezelfde golflengte zit. Het lijkt soms wel "griezelig" telepatisch. Er is volgens mij maar één vlak waarop we niet zo goed overeenkomen, en dat is onze appreciatie voor de omgevingstemperatuur :-). Bedankt mijn schatje, voor al de leuke, romantische en mooie momenten die we al samen hebben beleefd, en voor de liefde en steun die ik van jou gekregen heb. Zonder jou had ik het wellicht halverwege het doctoraat al voor bekeken gehouden. Nog eens bedankt voor alles hé ...

Met onderstaande cartoon zou ik dan het wellicht meest gelezen stuk van mijn doctoraat willen beëindigen. Die weerspiegelt een beetje de bedenkingen die ik heb bij veel wetenschappelijke publicaties die ik de afgelopen jaren bestudeerd heb. Rest mij u alleen nog veel leesplezier te wensen.

Gent, April 2005
Dirk De Pauw



List of abbreviations and symbols

Dimensions

#	number (e.g. mol)
l	length
m	mass
t	time

Abbreviations

Name	Description	Dimension
DAE	Differential and Algebraic Equation	
EAST	Enhanced Academic Simulation Tool	
FIM	Fisher Information Matrix	
GA	Genetic Algorithm	
MOGA	Multi-Objective Genetic Algorithm	
MRE	Maximum Relative Error	
ODE	Ordinary Differential Equation	
OED-PE	Optimal Experimental Design for Parameter Estimation	
OUR	Oxygen Uptake Rate	$\text{m.l}^{-3}.\text{t}^{-1}$
SA	Sensitivity Analysis	
SAE	Sum of Absolute Errors	
SRE	Sum of Relative Errors	
SSE	Sum of Squared Errors	
VFA	Volatile Fatty Acids	
WDVE	WEST distributed virtual experimentation	
WEST	Wastewater treatment plant Engine for Simulation and Training	

Roman symbols

Name	Description	Dimension
A_{clar}	clarifier surface	l^2
b_H	decay rate of heterotrophic biomass	t^{-1}
C	parameter estimation error covariance matrix	
C_H	parameter estimation error covariance matrix	
C_{tic}	total inorganic carbon concentration	$\#.l^{-3}$
$e_{\hat{\theta}}$	relative parameter estimation error	%
f_{ns}	non-settleable fraction	-
f_P	inert particulate fraction of the biomass	-
H	Hessian matrix	
H_{clar}	clarifier height	1
J	objective function	
k_1	yield for substrate degradation	-
k_2	yield for VFA production	$\#.m^{-1}$
k_3	yield for VFA consumption	$\#.m^{-1}$
k_4	yield for CO_2 production (by acidogenic biomass)	$\#.m^{-1}$
k_5	yield for CO_2 production (by methanogenic biomass)	$\#.m^{-1}$
k_6	yield for NH_4 production	$\#.m^{-1}$
k_h	hydrolysis rate constant	t^{-1}
K_a	equilibrium constant	$\#.l^{-3}$
K_b	equilibrium constant	$\#.l^{-3}$
K_d	decay rate	t^{-1}
K_{I2}	volatile fatty acids inhibition constant	$\#.l^{-3}$
K_{NH_4}	ammonium half-saturation constant	$m.l^{-3}$
K_O	oxygen half-saturation constant (for heterotrophic biomass)	$m.l^{-3}$
$K_{O\ NH}$	oxygen half-saturation constant for ammonium oxidizers	$m.l^{-3}$
$K_{O\ PAO}$	oxygen half-saturation constant for phosphate accumulating organisms	$m.l^{-3}$
K_S	substrate half-saturation constant	$m.l^{-3}$
K_{S_1}	organic substrate half-saturation constant	$m.l^{-3}$
K_{S_2}	VFA half-saturation constant	$\#.l^{-3}$
K_X	particulate COD half-saturation constant	$m.l^{-3}$
N	number of data points / individuals	
p	number of parameters	
Q	measurement error covariance matrix	
Q_C	carbon dioxide (CO_2) production rate	$l^3.t^{-1}$
Q_f	feed pump flow rate	$l^3.t^{-1}$
Q_M	methane (CH_4) production rate	$l^3.t^{-1}$
Q_{PHA}	rate constant for PHA storage	t^{-1}
Q_{PP}	rate constant for polyphosphate storage	t^{-1}
Q_T	total gas production rate	$l^3.t^{-1}$
R	correlation matrix	

S	substrate concentration	m.l^{-3}
S_1	organic substrate concentration	m.l^{-3}
S_2	volatile fatty acids concentration	$\#.1^{-3}$
S_f	feed concentration	m.l^{-3}
S_{NH}	ammonium concentration	m.l^{-3}
S_{NH4}	ammonium concentration	m.l^{-3}
S_{NO2}	nitrite concentration	m.l^{-3}
S_{NO3}	nitrate concentration	m.l^{-3}
S_O	oxygen concentration	m.l^{-3}
$S_{O,set}$	oxygen set point	m.l^{-3}
S_{PO4}	phosphate concentration	m.l^{-3}
S_S	readily biodegradable substrate concentration	m.l^{-3}
t_f	feed time	t
V	reactor volume	l^3
v_0	maximum settling velocity	l.t^{-1}
V_{anox}	feed volume to the first four SBR anoxic phases	l^3
X	biomass concentration	m.l^{-3}
X_1	acidogenic biomass concentration	m.l^{-3}
X_2	methanogenic biomass concentration	m.l^{-3}
$X_{B,A}$	autotrophic biomass concentration	m.l^{-3}
$X_{B,H}$	heterotrophic biomass concentration	m.l^{-3}
X_H	heterotrophic biomass concentration	m.l^{-3}
X_{NH4}	autotrophic biomass concentration	m.l^{-3}
Y	yield coefficient	-
Y_A	yield coefficient for autotrophic biomass	-
Y_H	yield coefficient for heterotrophic biomass	-
Y_{HNO3}	anoxic yield coefficient for heterotrophic biomass	-
Y_{NH4}	yield coefficient for autotrophic biomass	-
Y_{PO4}	yield coefficient for phosphate release	-
Z	total alkalinity	$\#.1^{-3}$

Greek symbols

Name	Description	Dimension
α	fraction of biomass in the liquid phase	-
δ	parameter estimation confidence interval	
δ^{msqr}	parameter sensitivity ranking	
ϵ	residual	
$\eta_{NO_3 HET}$	reduction factor for nitrate denitrification	-
θ	parameter	
Θ	parameter domain	
$\hat{\theta}$	parameter estimate	
λ	eigenvalue	
λ_{min}	smallest eigenvalue	

List of symbols

λ_{max}	largest eigenvalue	
μ	growth rate	t^{-1}
μ_{1max}	maximum growth rate of acidogenic bacteria	t^{-1}
μ_{2max}	maximum growth rate of methanogenic bacteria	t^{-1}
μ_{mA}	maximum growth rate for autotrophic biomass	t^{-1}
μ_{mH}	maximum growth rate for heterotrophic biomass	t^{-1}
μ_{max}	maximum growth rate	t^{-1}
$\mu_{max,H}$	maximum growth rate for heterotrophic biomass	t^{-1}
$\mu_{max,NH4}$	maximum growth rate for autotrophic biomass	t^{-1}
μ_{PAO}	maximum growth rate for phosphate accumulating organisms	t^{-1}
ξ	perturbation factor	
σ	standard deviation	
σ^2	variance	
τ	OUR transient term time constant	t
ψ	experimental degrees of freedom	
Ψ	domain of experimental degrees of freedom	
ψ_{API}	API-optimal experiment	
ψ_D	D-optimal experiment	
ψ_{ED}	ED-optimal experiment	
ψ_{EID}	EID-optimal experiment	
ψ_{GSID}	global SID-optimal experiment	
ψ_{LSID}	local SID-optimal experiment	
ψ_{MMD}	MMD-optimal experiment	

Contents

1	General concepts and problem statement	1
1.1	Mathematical modelling	1
1.2	Parameter estimation	4
1.2.1	Objectives of parameter estimation	4
1.2.2	Assessing parameter estimation quality	9
1.2.3	Justifying the linear approximation for nonlinear models	16
1.2.4	Drawing confidence ellipses in practice	16
1.3	Optimal experimental design	17
1.3.1	Background	17
1.3.2	Optimal experimental design for parameter estimation (OED-PE)	18
1.3.3	Iterative optimal experimental design procedure	22
1.4	Problem statement and thesis outline	27
2	Software development and description of test models	31
2.1	Software development	31
2.1.1	The WEST modelling and simulation software	31
2.1.1.1	Introduction	31
2.1.1.2	WEST(++) software architecture	33
2.1.2	The EAST simulation software	35
2.1.2.1	Modification of the WEST(++) experimentation environment	35
2.1.2.2	EAST experiment architecture	36
2.1.2.3	EAST simulation architecture	37
2.1.2.4	EAST optimization (parameter estimation) ar- chitecture	38

2.1.2.5	EAST sensitivity analysis architecture	39
2.1.2.6	EAST optimal experimental design for parameter estimation (OEDPE) architecture	42
2.1.2.7	EAST LabVIEW optimal experimental design architecture	46
2.1.2.8	EAST robust optimal experimental design architecture	47
2.1.3	Unifying WEST(++) and EAST in Tornado	48
2.2	Short description of test models	48
2.2.1	Batch and fed-batch Monod kinetics model	48
2.2.2	Anaerobic digestion model AM2	49
2.2.3	One-step nitrification model	50
2.2.4	Sequencing batch reactor (SBR) model	52
2.2.5	Two step nitrification/denitrification SBR model	54
2.2.6	Benchmark model	56
3	Local sensitivity analysis	59
3.1	Introduction	59
3.2	Sensitivity analysis	60
3.3	Practical aspects of the finite difference method	63
3.3.1	Theoretical background	63
3.3.2	Selection of test models	64
3.3.3	Quantifying the sensitivity calculations quality	67
3.3.4	Comparing different quality criteria	69
3.3.5	Selection of a useful criterion threshold value	71
3.3.6	Influence of integrator accuracy	73
3.4	The complex-step derivative approximation method	75
3.4.1	Theoretical background	75
3.4.2	Practical implementation issues	76
3.4.3	Numerical example	77
3.4.4	Computational cost comparison	78
3.5	Conclusions	79
4	Use of the Hessian and the FIM in parameter estimation	81
4.1	Introduction	81
4.2	Numerical Hessian matrix calculation techniques	82
4.2.1	Introduction	82
4.2.2	The quadratic approximation technique	83
4.2.3	Richardson's extrapolation	85

4.2.4	Combining the Richardson’s extrapolation method and the complex-step derivative approximation technique	88
4.2.5	Comparison of the numerical Hessian calculation techniques	89
4.3	Relationship between the Hessian and the FIM	92
4.3.1	Theoretical background	92
4.3.2	Numerical illustration	93
4.4	Selecting identifiable parameters	95
4.4.1	Overview of existing methods	95
4.4.1.1	Identifiability	95
4.4.1.2	Selecting practically identifiable parameter subsets	96
4.4.2	An improved procedure for parameter subset selection based on FIM related properties	102
4.4.2.1	Proposed procedure	102
4.4.2.2	Critical evaluation of the proposed procedure and comparison with other techniques	104
4.4.2.3	Testing the proposed procedure	105
4.5	Conclusions	115
5	Single-objective optimal experimental design	117
5.1	Introduction	117
5.2	The optimal experimental design optimization problem	118
5.3	Using classical optimization algorithms to solve the OED problem	122
5.3.1	Short introduction	122
5.3.2	The Simplex optimization algorithm	123
5.3.3	The Praxis optimization algorithm	124
5.3.4	Drawbacks of classical optimization algorithms	124
5.4	Using genetic algorithms to solve the OED problem	125
5.4.1	Classical genetic algorithms	125
5.4.1.1	Working principle and terminology	125
5.4.1.2	Problems encountered when using binary-coded GAs	128
5.4.2	Real-coded genetic algorithms	129
5.4.2.1	Background and working principle	129
5.4.2.2	Real-coded crossover and mutation operators	129
5.4.2.3	Combined discrete and continuous optimization problems	132
5.5	Designing measurement campaigns for SBRs	132
5.5.1	Introduction	132

5.5.2	Selection of the identifiable parameters	133
5.5.3	Optimization of measurement times	137
5.5.3.1	Problem description	137
5.5.3.2	Measurement points expressed as continuous variables	137
5.5.3.3	Measurement points expressed as discrete variables	140
5.5.4	Optimization of experimental manipulations	143
5.5.5	Combined optimization of experimental manipulations and measurement times	146
5.5.6	Practical significance of the optimal experimental designs 150	
5.6	Conclusions	152
6	Multi-objective optimal experimental design	155
6.1	Introduction	155
6.2	Multi-objective optimization	155
6.3	Classical methods for solving the multi-objective optimization problem	157
6.4	Multi-objective genetic algorithms	158
6.4.1	Introduction	158
6.4.2	Basic requirements for Pareto-based multi-objective GAs	158
6.4.3	NSGA-II	159
6.5	Designing SBR measurement campaigns using multi-objective GAs	161
6.5.1	Problem description	161
6.5.2	Algorithm settings	162
6.5.3	Multi-objective D-optimal and modified E-optimal design	163
6.5.3.1	Objectives	163
6.5.3.2	Experimental degrees of freedom	163
6.5.3.3	Results and discussion	163
6.5.4	Experimental cost versus D-optimal design	168
6.5.4.1	Objectives	168
6.5.4.2	Experimental degrees of freedom	168
6.5.4.3	Results and discussion	168
6.6	Conclusions	170
7	Automatic optimal experimental design	173
7.1	Introduction	173
7.2	An automatic optimal experimental design methodology	173

7.3	Application to a hybrid respirometer	176
7.3.1	Experimental setup	176
7.3.2	LabVIEW communication	178
7.3.3	Identification of a one-step nitrification model	180
7.3.4	Experimental degrees of freedom and constraints	181
7.3.5	Objective selection	182
7.3.6	Results and discussion	182
7.3.7	Evaluating the optimization of the experimental degrees of freedom	184
7.4	Conclusion	186
8	Robust experimental design	187
8.1	Introduction	187
8.2	Overview of existing techniques	188
8.2.1	Sequential design	188
8.2.2	Discrete design	188
8.2.3	Expected value approach	189
8.2.4	Maximin approach	190
8.3	Simulation study	191
8.3.1	System setup and data generation	191
8.3.2	Experimental degrees of freedom and objective	192
8.3.3	Initial design and calibration	192
8.3.4	Implemented algorithms and settings	194
8.3.4.1	Expected value approach	194
8.3.4.2	Maximin approach	196
8.3.4.3	Sensitivity index approach	196
8.3.5	Results and discussion	198
8.4	Conclusions	204
9	General conclusions and future perspectives	207
9.1	Goal of this chapter	207
9.2	Problem 1: Software availability	207
9.3	Problem 2: Local sensitivity analysis	208
9.4	Problem 3: Parameter identifiability	209
9.5	Problem 4: Global optimization in optimal experimental design	212
9.6	Problem 5: Multi-objective optimization in experimental design	213
9.7	Problem 6: Automation of experimental design	214
9.8	Problem 7: Robust experimental design	214
9.9	Final thoughts	215

Contents

Bibliography	219
Summary	233
Samenvatting	237
Curriculum vitae	241

Chapter 1

General concepts and problem statement

1.1 Mathematical modelling

A mathematical model can best be described as a mathematical description of a real process. These processes can be simple physical systems like a weight attached to a spring, more complex biological systems like wastewater treatment plants or very complex systems like the global climate. The reason why models are built for these processes can be very diverse. Models can for example be used in education, to increase the knowledge about the process, to predict and control the future behaviour of the system and to test certain hypotheses that can not be tested on the real process.

Generally, a model transforms a certain input (u) into an output (y) by some defined relationship. The outputs are the variables that are of interest to the user and the inputs are the disturbances and manipulations that affect the outputs. A wastewater treatment plant will be used as an example here. In this system, untreated wastewater coming from the sewer system enters the plant and is processed in different physical and biological steps to produce purified water. The input for a model of such a system would be the composition and quantity of the incoming wastewater while one of the outputs of the model could be the quality of the water flowing out of the treatment plant.

The relationship between input and output defines the basic structure and the type of the model. The model structure is defined by the equations which relate the input to the output. These equations consist of variables (states, inputs and outputs) and parameters. Parameters are model quantities that tend to be unknown and need to be estimated from prior knowledge or data. Variables and parameters can be arranged in different ways to form

the final equations leading to different types of models. Models for which the variables evolve over time are called dynamic, while models which are not dynamic are called static or steady-state. Models can also be called linear or nonlinear with respect to their variables and/or parameters. The evaluation of linearity can be performed by differentiating the equations with respect to the variable or the parameter in question and by evaluating whether the derivative is still function of the same variable or parameter. If this is the case, the model is said to be nonlinear in the variable or parameter. Linearity has an important impact on the way the model can be solved. For linear models, analytical solutions can be obtained, while nonlinear models often require numerical techniques in order to solve the model equations.

The biotechnological models used in this thesis can be classified as nonlinear and dynamic. The basic model structure is given by Equation 1.1.

$$\begin{aligned}\frac{dx}{dt} &= f(x, \theta, u, t), & x(t_0) &= x_0 \\ y &= g(x, \theta, u, t)\end{aligned}\tag{1.1}$$

In this equation x is a vector of state variables, θ a vector of parameters, y a vector of outputs, u a vector of inputs and t the independent variable. The dynamic nature of this model is reflected in the fact that it is formulated as a differential equation indicating the change of state variables x with respect to the independent variable t . In order to solve these equations initial values of x at time 0 have to be given: x_0 .

In order to illustrate various aspects of modelling, parameter estimation and experimental design in this chapter, a simple bioprocess model will be used in which microbial biomass is growing on a limiting substrate. The rate at which the bacteria grow, is modelled by the well known Monod equation (Equation 1.2).

$$\mu(S) = \frac{\mu_{max} \times S}{K_S + S}\tag{1.2}$$

This equation is one of the most commonly used expressions for describing saturation phenomena in different biological, chemical, pharmacological and medical processes. It was first derived for enzyme catalyzed reactions by Michaelis and Menten in 1913 and applied for the first time in microbiology by Monod (1949) and is therefore also called the Monod model. Monod used it to describe the growth rate μ (h^{-1}) of microorganisms. This growth is related to the substrate concentration S ($g.l^{-1}$) and is characterized by a maximum growth rate μ_{max} (h^{-1}) and a saturation constant K_S ($g.l^{-1}$) at which the growth rate is half of the maximum growth rate. This is illustrated in Figure 1.1 for a maximum growth rate of $0.39 h^{-1}$ and a saturation constant of $0.06 g.l^{-1}$.

Since its first application by Monod, it has been the most frequently used model for describing microbial growth in all kinds of applications, for example: batch, fed-batch and continuous fermentation processes (Bastin and Dochain, 1990), waste water treatment processes (Dochain and Vanrolleghem, 2001) and anaerobic digestion (Batstone et al., 2002).

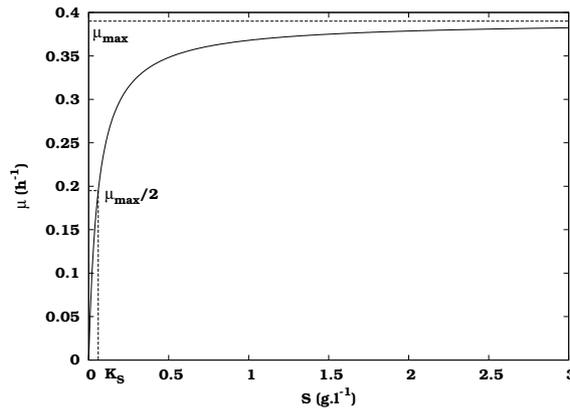


Figure 1.1: Microbial growth rate μ (h^{-1}) as a function of substrate concentration S ($g.l^{-1}$) based on Monod kinetics with $\mu_{max} = 0.39 h^{-1}$ and $K_S = 0.06 g.l^{-1}$.

Let us now consider a stirred fed-batch bioreactor in which biomass is grown on a limiting substrate. The term fed-batch indicates that the substrate is not only supplied at the start of the process (batch) but is also gradually fed to the reactor during the course of the process (fed-batch). This process can be modelled by Equations 1.3 to 1.5.

$$\frac{dX}{dt} = (\mu - K_d) \times X - \frac{Q_f}{V} \times X \quad (1.3)$$

$$\frac{dS}{dt} = -\frac{\mu}{Y} \times X + \frac{Q_f}{V} \times (S_f - S) \quad (1.4)$$

$$\frac{dV}{dt} = Q_f \quad (1.5)$$

In this simple model the process starts off as a batch process with a certain initial volume V_0 (l), substrate S_0 ($g.l^{-1}$) and biomass X_0 ($g.l^{-1}$) concentration. At a certain time t_f (h), a pump is switched on and substrate with a concentration S_f ($g.l^{-1}$) is fed to the reactor using a constant pump flow rate Q_f ($l.h^{-1}$). Equation 1.3 describes the change of the biomass concentration X ($g.l^{-1}$) with respect to time and is characterized by growth, decay and dilution. The biomass growth rate μ can be described by Equation 1.2, the decay process is modelled by a constant decay rate K_d (h^{-1}) and dilution is the result of the increasing reactor volume due to the influent feed stream. The change of substrate concentration S ($g.l^{-1}$) is given by Equation 1.4. It decreases due to microbial consumption and increases as a result of the input feeding stream. The amount of substrate which is converted into biomass is determined by the yield coefficient Y (gX/gS). The change of reactor volume (Equation 1.5) is obviously equal to the influent pump flow rate. In Figures 1.2 and 1.3 the evolution over time of the substrate and biomass concentration in the reactor is shown. In order to simulate these profiles, the model differential equations had to be integrated using a numerical

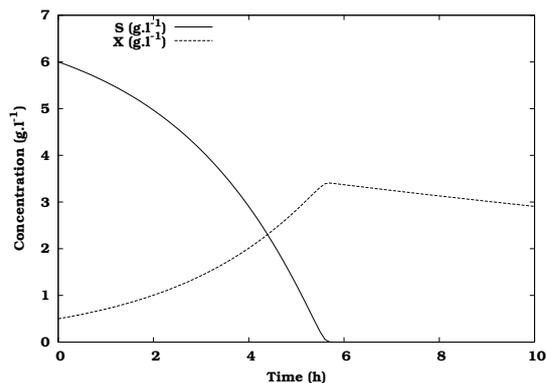


Figure 1.2: Time evolution of the substrate (S) and biomass (X) concentration during a batch experiment.

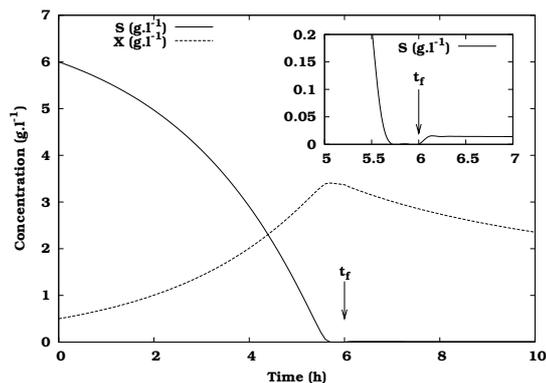


Figure 1.3: Time evolution of the substrate (S) and biomass (X) concentration during a fed-batch experiment. The inset magnifies the substrate concentration profile at the start of substrate feeding (t_f).

solver or integrator using the model parameters and initial conditions listed in Table 1.1. These values were taken from Baltes et al. (1994) who performed batch and fed-batch fermentation experiments with the yeast *Trichosporon cutaneum*. The batch experiment (Figure 1.2) shows a decrease in substrate concentration due to the biomass growth and a corresponding increase in biomass concentration. At around 5.5 h the substrate is depleted and from then on the biomass is subject to decay only, indicated by the gradual decrease in biomass concentration. The fed-batch experiment (Figure 1.3) shows identical concentration profiles in the first part of the experiment but at $t_f = 6$ h the feeding of substrate is started and its concentration increases slightly (visible in the inset of Figure 1.3). This causes the biomass to continue to grow although this is not noticeable in the figure due to the dilution effect of the feeding stream.

1.2 Parameter estimation

1.2.1 Objectives of parameter estimation

Before being able to use a model for any practical purpose it first needs to be calibrated. This means that the parameter values need to be chosen or estimated in such a way that the agreement between the measured data and the model prediction is as good as possible. Given a certain data set we intuitively feel that some parameter sets are very unlikely, i.e. those for which the model does not look like the data at all, while others are very likely, i.e. the ones for which the model closely resembles the data. In order to translate this feeling into mathematics, an appropriate objective function

Table 1.1: Parameter values and initial conditions for the batch and fed-batch growth model (from Baltes et al., 1994).

Parameter	Value	Unit
μ_{max}	0.39	h^{-1}
K_S	0.06	$g.l^{-1}$
Y	0.54	$gX.gS^{-1}$
K_d	0.037	h^{-1}
t_f	batch: ∞ / fed-batch: 6	h
S_f	4	$g.l^{-1}$
Q_f	2.4	$l.h^{-1}$
Initial condition	Value	Unit
X_0	0.5	$g.l^{-1}$
S_0	6	$g.l^{-1}$
V_0	15	l

needs to be constructed which quantifies the deviation of the model from the data. A frequently used function is the “least squares” or “sum of squares” objective function given by Equation 1.6.

$$J(\theta) = \sum_{i=1}^N (y_i - y_i(\theta))^2 \tag{1.6}$$

The objective $J(\theta)$ is the sum over all N data points of the squared difference between the model prediction $y_i(\theta)$ and the measurement y_i . If parameters are chosen close to the true model parameters (which are unknown), the value of $J(\theta)$ will tend to become small. The parameters of the model thus need to be adjusted in order to minimize the objective function value resulting in the p (number of estimated parameters) best-fit parameters ($\hat{\theta}$). The search for these best-fit parameters can be based on trial and error and human expertise, but more conveniently an automatic parameter estimation algorithm or optimizer can be used. The topic of optimization algorithms will not be discussed here in detail, but reference is made to Dochain and Vanrolleghem (2001) for an overview of existing algorithms.

A topic which has not been discussed so far is measurement error or noise. It is a well-known fact that all data is corrupted by measurement error to a certain extent which means that a model will never perfectly fit to the measurements even if the model structure is correct. This also means that with each measurement point an uncertainty is associated. Measurements which are accompanied by large measurement errors can be regarded as uncertain, while measurements with small measurement errors are more to be “trusted”. This notion of uncertainty can be added to the parameter estimation objective function resulting in a “weighted least squares” objective function (Equation 1.7).

$$J(\theta) = \sum_{i=1}^N \frac{1}{\sigma_i^2} (y_i - y_i(\theta))^2 \quad (1.7)$$

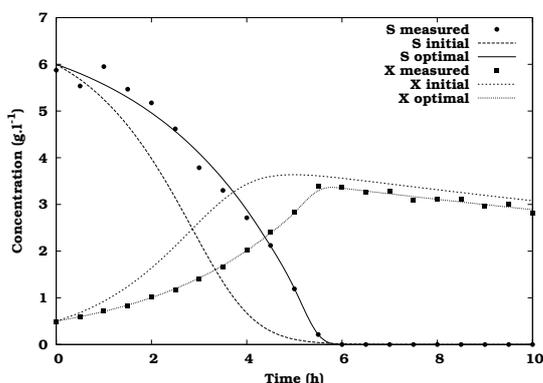
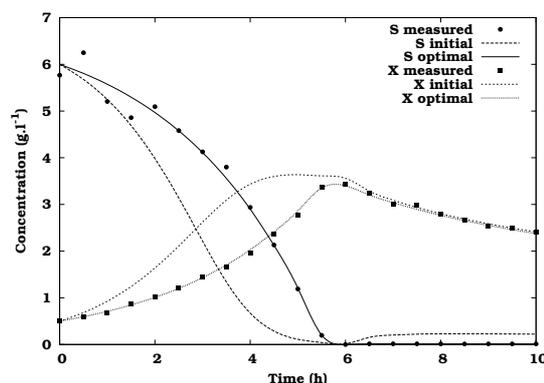
In this objective function σ_i^2 is the variance (square of the standard deviation σ_i) associated with the measurement and $1/\sigma_i^2$ acts as a weight in order to compensate for the possible uncertainty between the measurements. Small measurement error variances correspond with large weights which will amplify the difference between model and measurement and result in significant contributions to the final objective function value. On the other hand, deviations from the model predictions for measurements with a large variance (large measurement error) will contribute less to the overall objective function because of smaller weights. In the case where all measurement errors are equal, the weighting factor $1/\sigma_i^2$ can be removed from Equation 1.7 resulting in the ordinary least squares objective function (Equation 1.6).

The batch and fed-batch model described in Section 1.1 will now be used to illustrate these basic aspects of parameter estimation. Baltes et al. (1994) investigated the error characteristics of the substrate and biomass measurements and found that the errors were proportional to the measured concentrations. For this illustration, measurement errors of 5 and 2 % were assumed for substrate and biomass respectively, values that closely resemble the error characteristics determined by Baltes et al. (1994). Using these error characteristics and the parameters and initial conditions listed in Table 1.1 artificial data was generated for the batch and the fed-batch case. Based on these data three model parameters were estimated: μ_{max} , K_S and Y . The decay rate constant K_d was assumed to be known.

The results of the estimation procedure are shown in Table 1.2. Figures 1.4 and 1.5 show the generated data, the model simulations with the initial parameters and the model simulations using the best-fit parameters. From the figures it is clear that the model simulations using the initial parameter values (from which the parameter estimation was started) do not fit the measured data very well. On the other hand, the simulated substrate and biomass concentrations using the best-fit parameters show a close agreement with the generated data. This can also be seen from the initial and final objective function values listed in Table 1.2. Looking at the estimated parameter values themselves, it can be concluded that for the batch experiment only μ_{max} and Y could be estimated adequately, while K_S remained far from the true value. This result is also found by many other researchers who have tried to estimate the Monod parameters from batch data. Although it has been theoretically proven that both μ_{max} and K_S can be estimated from ideal batch measurements (without noise corruption) (Holmberg, 1982), the parameters in general cannot be correctly determined from noisy measurements. From the fed-batch data however, all parameters could be estimated adequately. The fact that the objective function value at the minimum is smaller for the batch case than for the fed-batch case is a result of the use of proportional errors in the objective function. With this type of error characteristics, small values of the measurements result in small errors

Table 1.2: Estimated parameters based on artificially generated data for a batch and fed-batch experiment.

Parameter	True value	Initial value	Batch estimate	Fed-batch estimate
μ_{max}	0.39	1	0.391	0.392
K_S	0.06	3	0.0759	0.0599
Y	0.54	0.6	0.536	0.545
Initial objective value			10485.41	778570.18
Final objective value			27.62	36.94


Figure 1.4: Artificially generated substrate and biomass measurements for a batch experiment together with the model simulations using the initial parameters and the best-fit parameters.

Figure 1.5: Artificially generated substrate and biomass measurements for a fed-batch experiment together with the model simulations using the initial parameters and the best-fit parameters.

and thus large weights. However, values below 1 mg.l^{-1} were assumed to be unmeasurable and were assigned a very small weight. As a consequence, the substrate measurements after 5 h in the batch case did not contribute to the final objective function value while the substrate measurements after the start of the reactor feeding in the fed-batch experiment did contribute to the objective function value because the concentrations remained above the detection limit. This caused the final objective function value in the fed-batch case to be higher.

To better understand why the optimization algorithm was unable to find the correct value for K_S , Figures 1.6 and 1.7 need to be compared. These figures show plots of the objective function value for the batch and the fed-batch case respectively in the neighbourhood of the minimum which was found through parameter estimation. The figures were constructed by defining a grid of values for two parameters while the third parameter was kept at its best-fit value. For each of these grid points the objective function

value was calculated and used to construct the three-dimensional surfaces shown in Figures 1.6b and 1.7b for the parameter combination μ_{max} and K_S . The two-dimensional projections or contour plots of these surfaces are shown in Figures 1.6a and 1.7a. In these figures the lines represent levels of equal objective function values similar to elevation contours on a map. The contour plots for other parameter combinations are shown in Figures 1.6c,d and 1.7c,d.

On the three-dimensional surface for the batch case (Figure 1.6b) a very narrow flat valley can be seen. Each combination of μ_{max} and K_S values in this valley corresponds to a model simulation which is able to describe the measured data equally well. This poses problems for the optimization algorithms because no clear direction can be found in which the objective function value improves and this results in a premature convergence of the algorithm and wrongly estimated parameters. In contrast to that, the contour plot of μ_{max} and Y (Figure 1.6c) shows no valley but a nicely shaped minimum easily detectable by the optimization algorithm. For the fed-batch case (Figure 1.7), the valley for μ_{max} and K_S has changed into a clearly defined minimum indicated by the more elliptical contour lines around the minimum. This change in the objective function surface illustrates why μ_{max} and K_S can be estimated from fed-batch data and not from batch data.

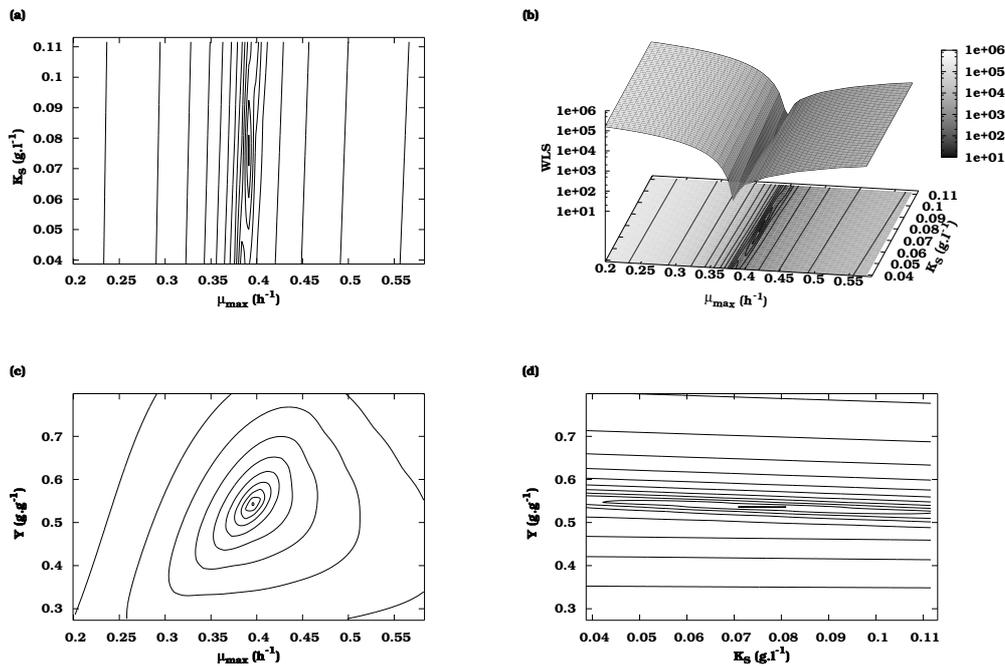


Figure 1.6: Objective function plots for the batch parameter estimation case. Contour plots of the objective value around the minimum for two parameters while the third parameter is kept at its best-fit value are shown in (a), (c) and (d). A three-dimensional representation of the contour plot shown in (a) is given in (b). In this graph, the WLS axis gives the values of the weighted least squares objective function.

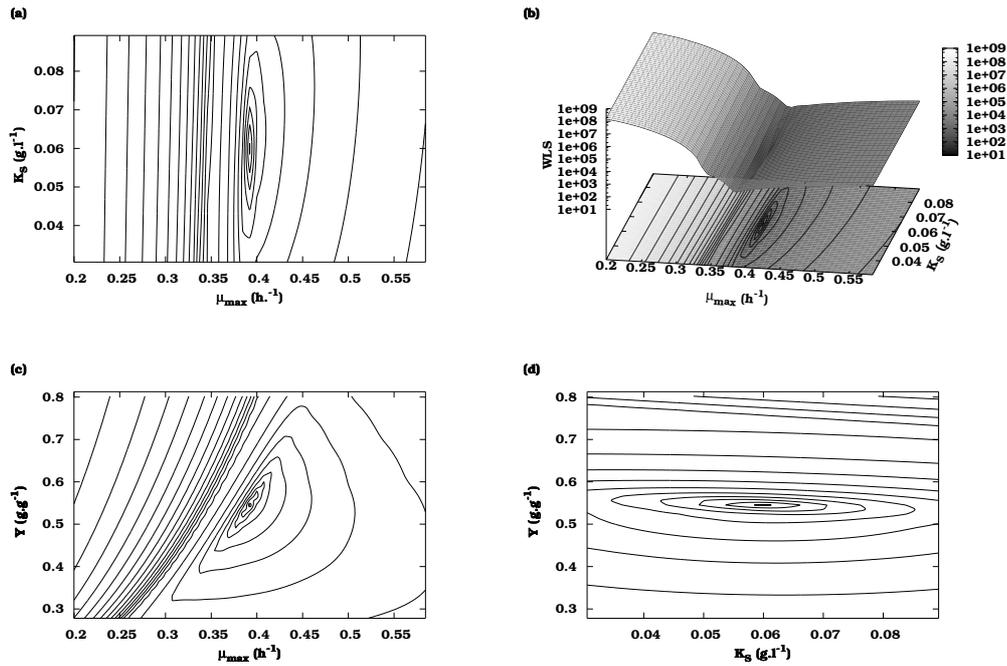


Figure 1.7: Objective function plots for the fed-batch parameter estimation case. Contour plots of the objective value around the minimum for two parameters while the third parameter is kept at its best-fit value are shown in (a), (c) and (d). A three-dimensional representation of the contour plot shown in (a) is given in (b). In this graph, the WLS axis gives the values of the weighted least squares objective function.

1.2.2 Assessing parameter estimation quality

Fitting the model to the data and finding the best-fit parameters ($\hat{\theta}$) is not the end of parameter estimation. In order for the parameter values to be meaningful they need to be accompanied by a confidence region. This is the region in which it can be expected that, with a certain probability, the true parameters lie. In other words, for a 95 % confidence region, there is a 95 % chance that the true parameter values fall within this region. As discussed in the previous section, the objective function (Equation 1.7) gives an indication how close the model fits the experimental data. It is thus justifiable to base the confidence region on the contours of $J(\theta)$. An exact confidence region is given by Equation 1.8 (Seber and Wild, 1989).

$$\{\theta : J(\theta) \leq c \times J(\hat{\theta})\} \quad (c > 1) \quad (1.8)$$

The confidence region is thus bounded by the value $c \times J(\hat{\theta})$ where $J(\hat{\theta})$ is the value of the objective function at the minimum. The constant c is generally unknown but approximations can be made for large enough number

of data points N by (Seber and Wild, 1989):

$$\left\{ \theta : J(\theta) \leq \left(1 + \frac{p}{N-p} \times F_{p, N-p}^{1-\alpha} \right) \times J(\hat{\theta}) \right\} \quad (1.9)$$

where $F_{p, N-p}^{1-\alpha}$ is the upper α critical level of the F -distribution with p (number of estimated parameters) and $N - p$ degrees of freedom. This region has the required $100 \times (1 - \alpha)\%$ asymptotic confidence level. As this region is very difficult to determine for nonlinear models, some extensive techniques have been proposed by Vanrolleghem and Keesman (1996) and Lobry et al. (1991) in order to determine it. Vanrolleghem and Keesman (1996) used a Monte Carlo technique and Lobry et al. (1991) proposed to use a contraction method starting from a confidence region which encloses the exact confidence region.

Rather than trying to find the exact confidence region, most often linear approximations are used to construct the confidence region. For linear models the objective function contours are of a quadratic form meaning that the two-dimensional confidence regions are ellipses and the three-dimensional confidence regions are ellipsoids. For nonlinear models Equation 1.7 is not exactly quadratic, but for sufficiently small deviations $\theta - \hat{\theta}$, it can be approximated by a second order series expansion of $J(\theta)$ around the best-fit parameters $\hat{\theta}$ given by Equation 1.10.

$$J(\theta) \approx J(\hat{\theta}) + \frac{1}{2} \times (\theta - \hat{\theta})^T \times \left. \frac{\partial^2 J}{\partial \theta \partial \theta^T} \right|_{\hat{\theta}} \times (\theta - \hat{\theta}) \quad (1.10)$$

The first order term in this expansion is missing because the first derivative of the objective function to the parameters equals 0 at the minimum. Substituting Equation 1.10 in Equation 1.9 gives us a new expression for the (quadratic) confidence region:

$$(\theta - \hat{\theta})^T \times C_H^{-1}(\hat{\theta}) \times (\theta - \hat{\theta}) \leq p \times F_{p, N-p}^{1-\alpha} \quad (1.11)$$

where $C_H^{-1}(\hat{\theta})$ is called the inverse of the parameter estimation error covariance matrix defined by:

$$C_H(\hat{\theta}) = 2 \times \frac{J(\hat{\theta})}{N-p} \times H(\hat{\theta})^{-1} \quad (1.12)$$

in which $\frac{J(\hat{\theta})}{N-p}$ is an approximation of the residual variance σ^2 . This approximation is based on the fact that, assuming the model is correct, the residuals will be random errors and the average of these squared residuals is an estimate of the error variance. In the above equation, $H(\hat{\theta})^{-1}$ is the inverse Hessian matrix defined by:

$$H(\hat{\theta}) = \left. \frac{\partial^2 J}{\partial \theta \partial \theta^T} \right|_{\hat{\theta}} \quad (1.13)$$

The error covariance matrix of the parameter estimates $C_H(\hat{\theta})$ as defined by Equation 1.12 is the cornerstone for assessing the parameter estimation quality and has the general form:

$$C = \begin{bmatrix} \sigma_{\theta_1}^2 & cov(\theta_1, \theta_2) & \dots & cov(\theta_1, \theta_p) \\ cov(\theta_2, \theta_1) & \sigma_{\theta_2}^2 & & \\ \vdots & & \ddots & \\ cov(\theta_p, \theta_1) & & & \sigma_{\theta_p}^2 \end{bmatrix} \quad (1.14)$$

The diagonal elements of this matrix are the variances (σ^2) of the errors of the parameter estimates and the off-diagonal elements are the covariances between the parameter estimation errors. Based on these variances and covariances the elements of the linear correlation matrix can be calculated as:

$$R_{ij} = \frac{cov(\theta_i, \theta_j)}{\sqrt{\sigma_{\theta_i}^2 \times \sigma_{\theta_j}^2}} \quad (1.15)$$

These elements give a measure for the linear correlation between two parameters and can range between -1 and 1 . If the linear correlation coefficient (R_{ij}) is close to -1 or 1 the parameters are said to be highly correlated, while correlation coefficients close to 0 imply low correlations.

As already discussed, the parameter estimation error covariance matrix can be used to construct confidence regions in various dimensions. The confidence region of a single parameter ($\hat{\theta}_i$) is a one dimensional interval δ_i typically calculated as (Seber and Wild, 1989):

$$\delta_i = \pm t_{N-p}^\alpha \times \sqrt{C_{ii}} \quad (1.16)$$

where t_{N-p}^α is the two-tailed Student t-distribution for the given confidence level $1 - \alpha$ and $N - p$ degrees of freedom (e.g. 95% confidence: $\alpha = 0.05$). Additionally, a relative parameter estimation error $e_{\hat{\theta}_i}$, expressed as a percentage of the parameter value, can be calculated by using Equation 1.17.

$$e_{\hat{\theta}_i} = \frac{\delta_i}{\hat{\theta}_i} \times 100 \quad (1.17)$$

Confidence regions in two dimensions are represented by confidence ellipses and can be drawn based on the parameter estimation error covariance matrix using the following parametric equation:

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \sqrt{p \times F_{p, N-p}^{1-\alpha}} \times \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}^T \times \begin{bmatrix} \sqrt{\frac{1}{\lambda_1}} & 0 \\ 0 & \sqrt{\frac{1}{\lambda_2}} \end{bmatrix} \\ \times \begin{bmatrix} \cos(\phi) \\ -\sin(\phi) \end{bmatrix} + \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \end{bmatrix} \quad (1.18)$$

where the angle ϕ ranges from 0 to 2π . λ_i is the eigenvalue belonging to the eigenvector W_i [W_{i0} W_{i1} ... W_{ip}] resulting from an eigenvector and eigenvalue decomposition of the inverse covariance matrix C^{-1} given by:

$$C^{-1} = W \times L^{-1} \times W^T \quad L = \text{diag}(\lambda) \quad (1.19)$$

The eigenvalues from this decomposition are proportional to the size of the axes of the ellipse, while the eigenvectors determine the orientation of the ellipse axes.

Similar to confidence ellipses, confidence regions in three dimensions are represented by confidence ellipsoids and are drawn based on the following parametric equation:

$$\begin{aligned} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} &= \sqrt{p \times F_{p, N-p}^{1-\alpha}} \times \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{32} & W_{33} \end{bmatrix}^T \times \begin{bmatrix} \sqrt{\frac{1}{\lambda_1}} & 0 & 0 \\ 0 & \sqrt{\frac{1}{\lambda_2}} & 0 \\ 0 & 0 & \sqrt{\frac{1}{\lambda_3}} \end{bmatrix} \\ &\times \begin{bmatrix} \cos(\phi) \times \sin(\varphi) \\ \sin(\phi) \times \sin(\varphi) \\ \cos(\varphi) \end{bmatrix} + \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \hat{\theta}_3 \end{bmatrix} \end{aligned} \quad (1.20)$$

where ϕ ranges from 0 to 2π and φ ranges from 0 to π . A visual interpretation of the confidence regions can only be provided for a maximum of 3 parameters. If more parameters are estimated, conditional confidence regions can be constructed by sub-sampling the complete Hessian matrix, i.e. selecting only those elements from the Hessian matrix corresponding to the parameters of interest. This partial Hessian matrix can then be inverted and used to calculate the covariance matrix from which the confidence intervals, ellipses or ellipsoids can be constructed. Geometrically this operation corresponds to making the intersection between the (hyper)ellipsoid and the (hyper)plane passing through the best-fit value of the parameters which are not considered.

To make these aspects of parameter estimation a bit more concrete, Table 1.3 lists confidence information for both the batch and the fed-batch parameter estimation problem. In this table the covariance (Equation 1.14) and correlation (Equation 1.15) matrices of the parameter estimates are given, together with the single parameter estimation confidence intervals (Equation 1.16) and the parameter estimation errors (Equation 1.17).

Figures 1.8 and 1.9 show the one-, two- and three-dimensional confidence regions for the batch and fed-batch case respectively. Each of these figures shows the individual 95 % confidence intervals and 95 % confidence ellipses (with axes) for parameter combinations $\mu_{max} - K_S$ (Figures 1.8a and 1.9a), $\mu_{max} - Y$ (Figures 1.8c and 1.9c) and $K_S - Y$ (Figures 1.8d and 1.9d). Figures 1.8b and 1.9b show the three dimensional 95 % confidence ellipsoids. The confidence ellipses were constructed by selecting the elements from the

(3×3) Hessian matrix that corresponded to the required parameter pair. From this partial Hessian matrix the covariance matrix for the parameter pair was calculated and used to draw the ellipses.

Some conclusions that were made from the contour and surface plots of the objective function (Figures 1.6 and 1.7) are now quantified in the confidence information. The fact that the objective function surface for μ_{max} and K_S changed from a narrow flat valley to a nicely defined minimum can be explained by looking at the parameter estimation errors and the correlation coefficient. The parameter estimation error for K_S , obtained from the batch experiment, is very large compared to the error obtained from the fed-batch experiment causing the confidence interval for K_S to be very large in the batch case. Beside that, μ_{max} and K_S are highly correlated in the batch case (0.93 compared to 0.28) resulting in a large and narrow confidence ellipse for the batch case and a smaller and less narrow ellipse for the fed-batch case (compare Figures 1.8a and 1.9a).

Table 1.3: Covariance and correlation matrix, 95 % confidence intervals and parameter estimation errors for the batch and the fed-batch parameter estimation problem.

Batch				Fed-batch			
Covariance matrix				Covariance matrix			
	μ_{max}	K_S	Y		μ_{max}	K_S	Y
μ_{max}	$5.91E-06$	$4.08E-05$	$1.88E-08$	μ_{max}	$1.12E-06$	$3.22E-07$	$3.26E-06$
K_S	$4.08E-05$	$3.22E-04$	$-1.71E-05$	K_S	$3.22E-07$	$1.15E-06$	$4.65E-07$
Y	$1.88E-08$	$-1.71E-05$	$8.27E-06$	Y	$3.26E-06$	$4.65E-07$	$1.09E-05$
Correlation matrix				Correlation matrix			
	μ_{max}	K_S	Y		μ_{max}	K_S	Y
μ_{max}	1.00	0.93	0.01	μ_{max}	1.00	0.28	0.93
K_S	0.93	1.00	-0.33	K_S	0.28	1.00	0.13
Y	0.01	-0.33	1.00	Y	0.93	0.13	1.00
95 % confidence intervals and estimation errors				95 % confidence intervals and estimation errors			
	Value	Interval (δ)	Error ($e_{\hat{\theta}}$) (%)		Value	Interval (δ)	Error ($e_{\hat{\theta}}$) (%)
μ_{max}	0.391	$\pm 4.92E-03$	1.26	μ_{max}	0.392	$\pm 2.14E-03$	0.54
K_S	0.0759	$\pm 3.63E-02$	47.85	K_S	0.0599	$\pm 2.17E-03$	3.62
Y	0.536	$\pm 5.82E-03$	1.09	Y	0.545	$\pm 6.67E-03$	1.22

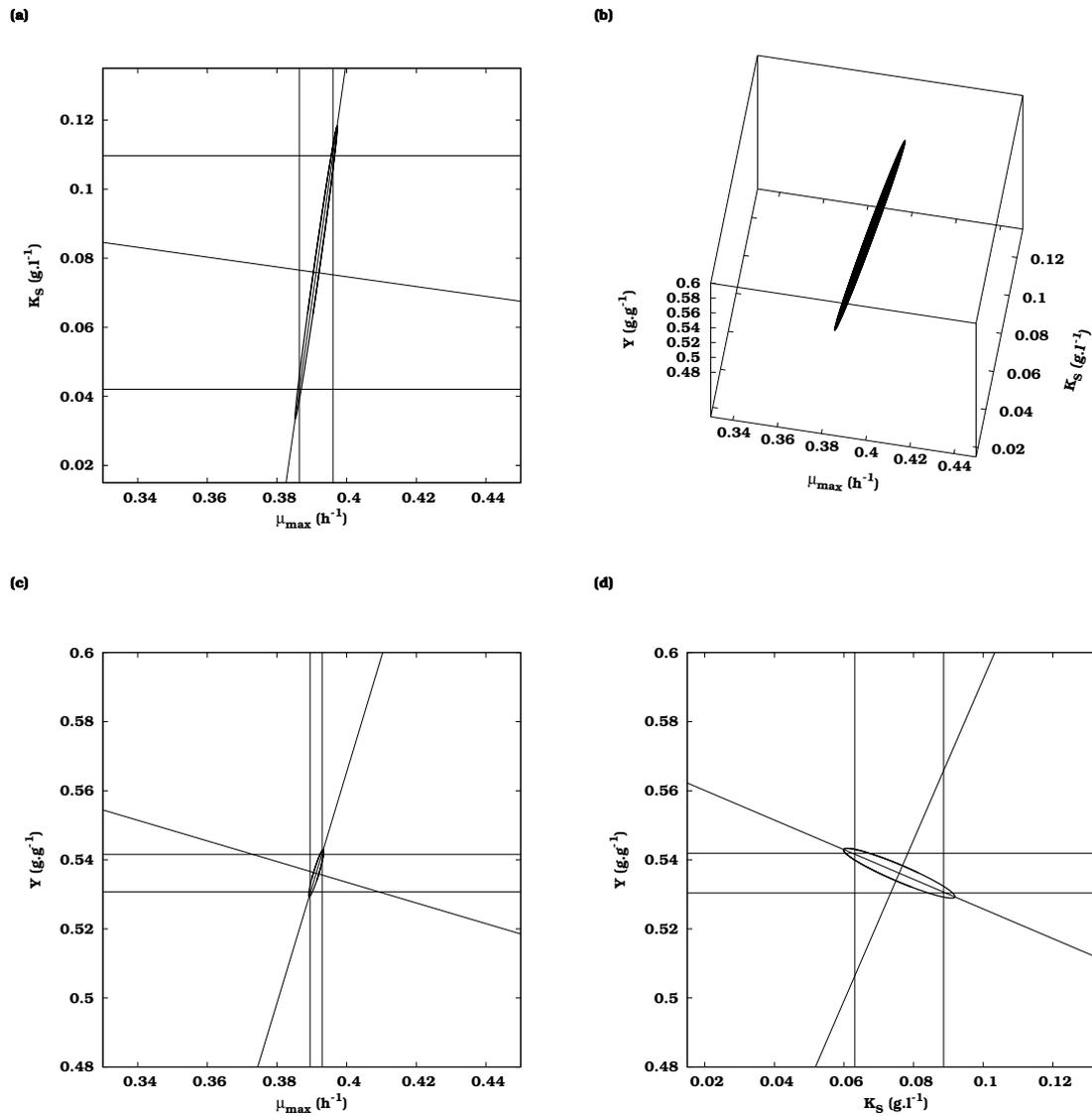


Figure 1.8: Batch parameter estimation problem confidence regions. The 95 % confidence ellipses for parameter combinations $\mu_{max} - K_S$, $\mu_{max} - Y$ and $K_S - Y$ are shown in (a),(c) and (d). Each of these figures also includes the one-dimensional confidence intervals and the ellipse axes. The 95 % confidence ellipsoid for the 3 parameters is shown in (b).

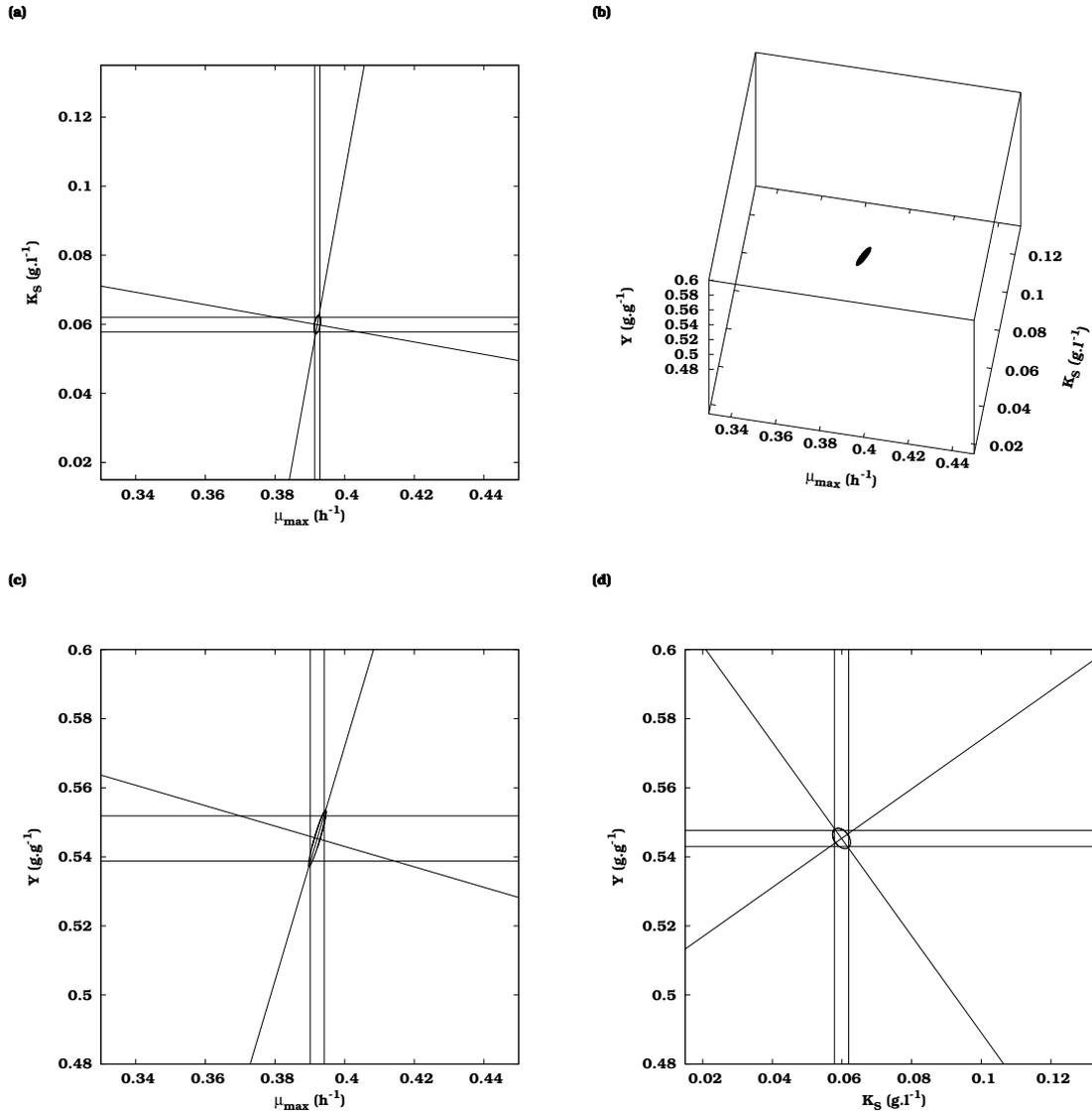


Figure 1.9: Fed-batch parameter estimation problem confidence regions. The 95 % confidence ellipses for parameter combinations $\mu_{max} - K_S$, $\mu_{max} - Y$ and $K_S - Y$ are shown in (a),(c) and (d). Each of these figures also includes the one-dimensional confidence intervals and the ellipse axes. The 95 % confidence ellipsoid for the 3 parameters is shown in (b).

1.2.3 Justifying the linear approximation for nonlinear models

As already discussed in the previous section, most confidence information is provided based on the assumption that any nonlinear model behaves like a linear model sufficiently close to the minimum. From some of the contour plots for the batch and the fed-batch case (especially Figures 1.6c and 1.7c) it is clear that the contours do not show an ideal elliptical shape. However, Figure 1.10a shows the same contours for μ_{max} and Y but closer to the minimum. The contours are drawn starting at an objective function value of 42.88, for the inner contour, which equals the 95 % confidence region according to Equation 1.9. Figure 1.10b shows the 95 % confidence ellipse for μ_{max} and Y according to Equation 1.18. From these figures it is clear that the confidence region and the confidence ellipse coincide and that a linear approximation of the objective function around the optimum is justified in this case.

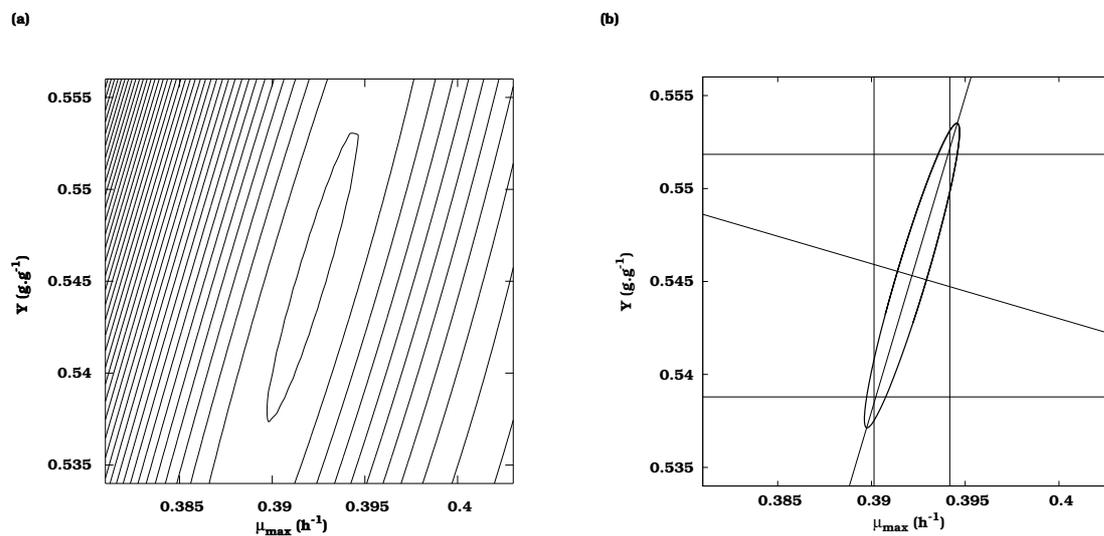


Figure 1.10: Comparison between confidence region and confidence ellipse of the μ_{max} and Y parameter combination for the fed-batch case. Contour lines of equal objective function values are plotted in (a); the inner contour line representing the 95 % confidence region. The 95 % confidence ellipse, based on a linear approximation, is represented in (b).

1.2.4 Drawing confidence ellipses in practice

Confidence ellipses are often employed in order to give a visual interpretation of the relation between several parameters. However, some attention should be paid to how these ellipses are represented. Wrong representation might lead to wrong interpretation of parameter relations. To illustrate this, Figure 1.11 shows the 95 % confidence ellipse for parameter combination K_S

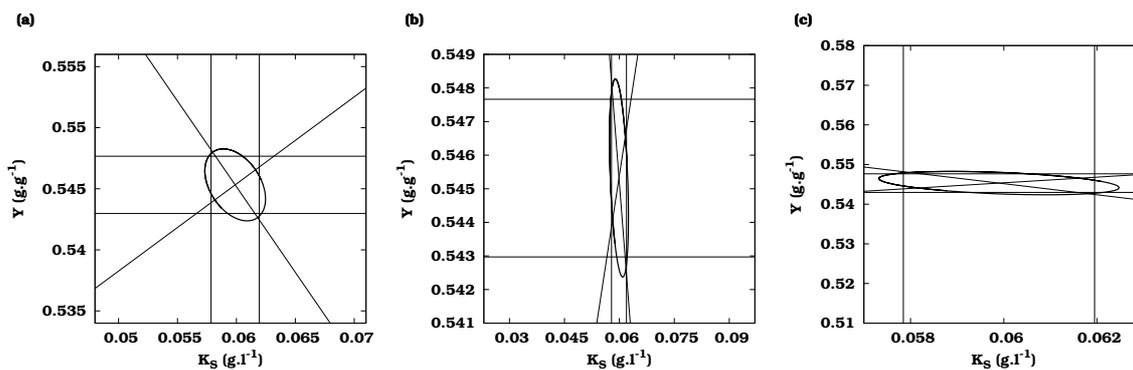


Figure 1.11: The 95 % confidence ellipse for parameter combination K_S and Y for the fed-batch experiment. This ellipse is represented in (a), (b) and (c) for different axes scales.

and Y for the fed-batch experiment. For a correctly represented ellipse (Figure 1.11a), the plot should be square and each axis should have the same scale in order for the axes of the confidence ellipse to be perpendicular. Figures 1.11b and 1.11c show the same confidence ellipse but with different axes scales. The ellipse axes are no longer perpendicular and these ellipses might lead to the conclusion that the parameters are strongly correlated which is not the case in reality. This clearly demonstrates that visual interpretation of parameter confidence information should be done with some caution.

1.3 Optimal experimental design

1.3.1 Background

From the comparison between the batch and the fed-batch experiments described in the previous section it is clear that certain experiments contain more information than others. Therefore, it is desired to design experiments in such a way that the data which will be collected from these experiments are as information-rich as possible. In order to design an experiment, different choices have to be made. It has to be decided whether, where and how the system under study will be manipulated and where, how and when measurements will be performed on this system. Beside these choices, the final purpose of the experiment also needs to be defined. Experimental data can be gathered for several reasons: (1) to get a better understanding of some phenomena of the system under study, (2) to estimate (some of) the parameters of a model or (3) to discriminate between several possible model structures. Experimental design which combines several of these goals is possible. However, the focus of this doctoral thesis will be on optimal experimental design for parameter estimation (OED-PE).

The concept of experimental design is not new. It arose first in the analysis of experiments in agriculture and its first steps can be traced back to the beginning of the 20th century. Especially the work of Fisher (1949) formed the basis for further developments. Initially most attention was given to experimental design for linear regression models. Later, also non-linear models which had an analytical solution were examined. Most of these applications were situated in chemical kinetics (Box and Lucas, 1959; Cochran, 1973; Ford et al., 1989). It was only in the last two decades that more and more optimal experimental design was performed on models which could only be solved numerically. Typical examples of these models can be found in the field of bioprocess engineering, environmental engineering, chemistry and food-technology (Munack, 1991; Vanrolleghem et al., 1995; Cunha et al., 1997; Versyck et al., 1997; Grijspeerdt and Vanrolleghem, 1999; Atkinson and Bogacka, 2002).

1.3.2 Optimal experimental design for parameter estimation (OED-PE)

In order to relate parameter estimation and optimal experimental design we need to return to the weighted least squares objective function (Equation 1.7). The expected value of the objective function for a parameter set slightly different from the optimal one can be written as:

$$E [J(\theta + \delta\theta)] = \sum_{i=1}^N (y_i - y_i(\theta + \delta\theta))^T \times Q_i^{-1} \times (y_i - y_i(\theta + \delta\theta)) \quad (1.21)$$

Introducing a linearisation of the model with respect to the parameters:

$$y_i(\theta + \delta\theta) \approx y_i(\theta) + \left[\frac{\partial y_i}{\partial \theta}(\theta) \right]_{\theta} \times \delta\theta \quad (1.22)$$

leads to the following expression of the expected value:

$$E [J(\theta + \delta\theta)] = E [J(\theta)] + \delta\theta^T \times \left[\sum_{i=1}^N \left(\frac{\partial y_i}{\partial \theta}(\theta) \right)^T \times Q_i^{-1} \times \left(\frac{\partial y_i}{\partial \theta}(\theta) \right) \right] \times \delta\theta \quad (1.23)$$

In order to get a clearly defined minimum for $J(\theta)$ it is necessary that the difference between $J(\theta)$ and $J(\theta + \delta\theta)$ is maximized. If this is the case, the minimum will be located in a deep "pit" rather than a small depression on a flat objective function plane. Maximizing the difference between $J(\theta)$ and $J(\theta + \delta\theta)$ can be achieved by maximizing the last term between square brackets in Equation 1.23. This matrix is known as the Fisher Information Matrix or FIM (Equation 1.24)

$$FIM = \sum_{i=1}^N \left(\frac{\partial y_i}{\partial \theta}(\theta) \right)^T \times Q_i^{-1} \times \left(\frac{\partial y_i}{\partial \theta}(\theta) \right) \quad (1.24)$$

The FIM quantifies the information content of an experiment related to the model parameters. The effect of the FIM maximization on the parameter estimation objective function surface is illustrated in Figure 1.12 for a single parameter estimation problem. This figure shows the objective function surfaces for a parameter estimation based on data of two experiments, a non-optimal experiment (outer curve) and an optimal experiment (inner curve). The parameter estimation based on the data of the optimal experiment (after FIM maximization) results in a much narrower objective function profile and a smaller confidence interval.

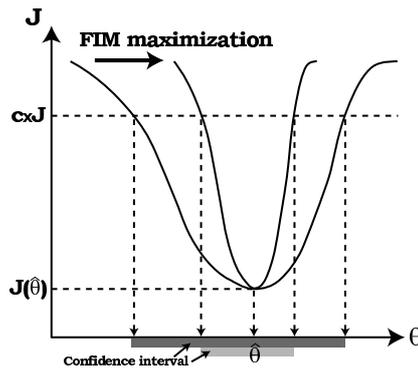


Figure 1.12: Influence of the FIM maximization on the objective function surface $J(\theta)$ and confidence intervals for a single parameter estimation problem. The objective function cuff-off value for constructing the confidence interval is represented by $c \times J$.

The FIM is composed of two components: sensitivity functions ($\partial y / \partial \theta$) and measurement errors (Q). The summation of these two components is made over all measurement points (N).

A first component of the FIM are the sensitivity functions. These are the partial derivatives of the variables to the parameters, evaluated at a certain point in parameter space (local sensitivity), and express how sensitive the variables are to the parameters. In other words: sensitivity functions express how much the variable will change with respect to a change in the parameter value. A variable which is highly sensitive to a certain parameter will contain much information about this parameter, while a variable which is insensitive to the parameter does not contribute to the information content of this parameter.

The second component of the FIM is the measurement error in the form of the inverse measurement error covariance matrix Q^{-1} . It is obvious that a measurement which has a large measurement error will contribute less to the information content than a measurement with a small measurement error. It should also be mentioned that correlations between measurements can also be specified using this covariance matrix. Throughout this thesis it will be assumed that the error characteristics of the measurements can be described in a relatively simple way: by using absolute or relative errors. However, a more thorough investigation of the error characteristics might

be required. Even noise modelling or applying Kalman filtering could be considered.

According to the Cramer-Rao inequality (Ljung, 1999) the inverse FIM is related, under certain conditions, to the lower bound of the parameter estimation error covariance matrix (Equation 1.25). The necessary conditions are that the measurement noise is uncorrelated and white (independent and normally distributes with zero mean). It is also required that the residuals are uncorrelated and white. Something that unfortunately rarely happens for nonlinear systems.

$$C \geq FIM^{-1} \quad (1.25)$$

As an example the two components of the FIM are illustrated in Figures 1.13 to 1.16. Figures 1.13 and 1.14 show the sensitivity functions of the substrate (S) and the biomass (X) to each of the model parameters μ_{max} , K_S and Y for the batch and fed-batch case respectively. The figures show that the sensitivity of the variables to the parameters is dynamic (changing with time). All sensitivity functions show a peak around 5.5 h which means that a change in one of the parameters causes the largest change in the variable at this time instance. This also means that the value of the variable at that time instance contains much information on the parameters and is thus an ideal candidate measurement point. For the batch case most sensitivity functions, except for $\partial X/\partial Y$, have a similar shape. This means that a change in one parameter can be compensated by a change in another parameter. In other words, a change in one parameter can cause the simulated trajectory of a variable to shift in a certain direction while a change in another parameter can cause a nearly exact shift in the opposite direction. This phenomenon indicates that the parameters are heavily correlated. This was also confirmed by the analysis of the parameter estimation quality in Section 1.2.2. For the fed-batch case it can be noticed that the first part of the sensitivity functions is identical to the batch case, but differences appear once the reactor feeding is started at 6 h.

The second component of the FIM are the inverse measurement errors which act as weights for the sensitivity functions. Small measurement error values are indications of reliable measurements and thus correspond to large weights in the FIM. As already mentioned in Section 1.2.1, measurement errors of 5 and 2 % were assumed for substrate and biomass respectively. Since these errors are relative to the value of the variable, the weights will be different at each measurement point. Figure 1.15 shows the log-scaled weights for the batch case. The weights for the substrate increase as the substrate concentration decreases until the substrate concentration reaches the detection limit of 1 mg.l^{-1} resulting in a 0 weight. The weights related to the biomass measurements remain more or less constant because only a slight increase and decrease in the biomass concentration value is observed. For the fed-batch case, the weights are shown in Figure 1.16 and just like the sensitivity functions they are equal to the ones

of the batch case in the first part of the experiment. However, at 6 h new substrate is added to the reactor and the concentration again rises above the detection limit resulting in non-zero weights and thus informative measurements. The biomass concentration weights show a similar profile as the one for the batch case.

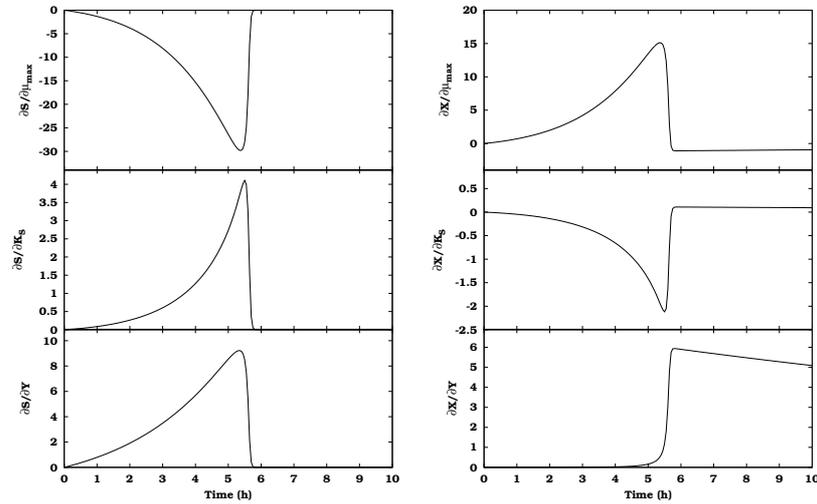


Figure 1.13: Sensitivity functions of the substrate and biomass concentration with respect to the model parameters μ_{max} , K_S and Y for the batch experiment.

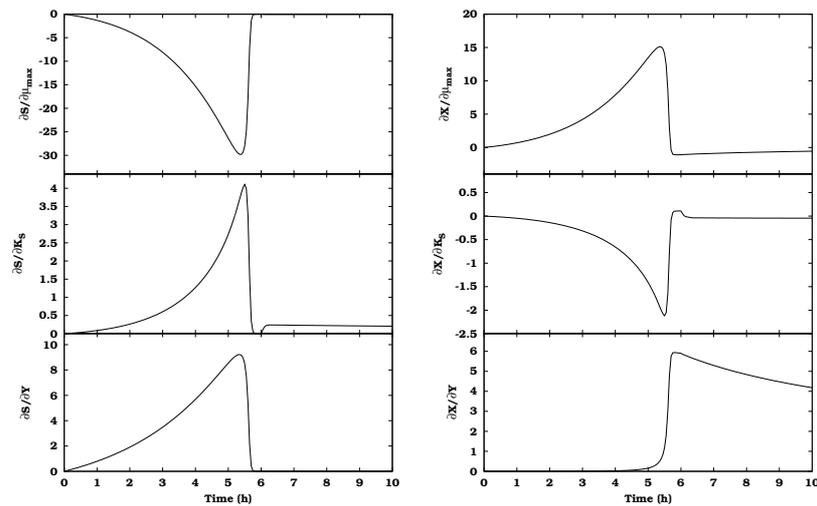


Figure 1.14: Sensitivity functions of the substrate and biomass concentration with respect to the model parameters μ_{max} , K_S and Y for the fed-batch experiment.

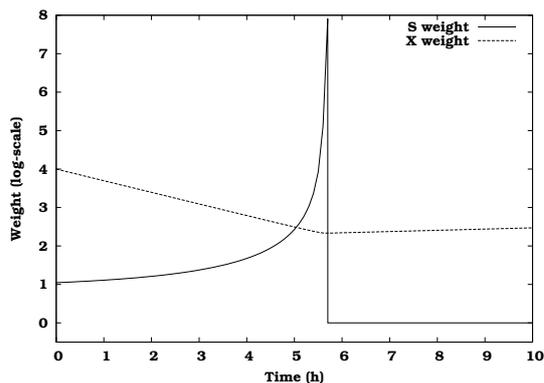


Figure 1.15: Measurement weights for the batch case.

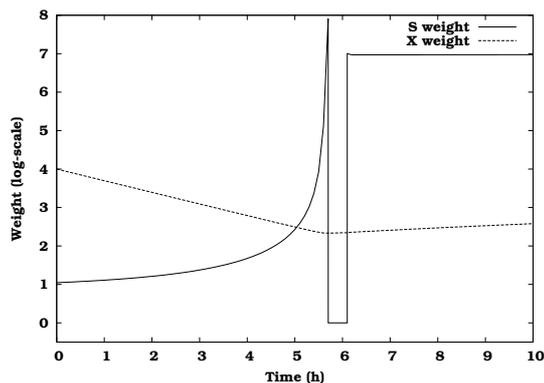


Figure 1.16: Measurement weights for the fed-batch case.

1.3.3 Iterative optimal experimental design procedure

Performing optimal experimental design in practice involves a series of steps summarized in Figure 1.17 (Dochain and Vanrolleghem, 2001). A first and essential requirement of the experimental design procedure is the availability of a preliminary model. This model may have been calibrated on the basis of data of an initial experiment or default or literature parameter values can be assumed. Another important assumption in order for the design procedure to produce useful results is the validity of the model structure. The model structure should be an adequate representation of the system under study. Although methods have been developed to investigate model validity, it will not be the focus of this thesis.

Once a preliminary model is available, the experimental degrees of freedom and constraints which define the boundaries for the experimental design procedure can be defined. The experimental degrees of freedom can be subdivided into two categories: measurements and manipulations. Several questions can be posed concerning the measurements that will be performed during the experiment: (1) what to measure, (2) where to measure and (3) when to measure? The second category of experimental degrees of freedom are the available manipulations. These relate to the excitation signals that will act on the system in order to produce highly qualitative information. In many cases this requires some creative thinking to come up with useful experimental manipulations. Examples of manipulations could be input flow rate profiles, reactor temperature profiles, influent concentrations, pulses of substrate. In order for the experimental degrees of freedom to be practically useful, constraints for each experimental degree of freedom should be defined. This to allow the experimental design procedure to come up with experiments which can be performed in practice.

Before any search for an optimal experiment can be initiated, an objective has to be defined. For optimal experimental design for parameter estimation, the objective will be the information content of the experiment

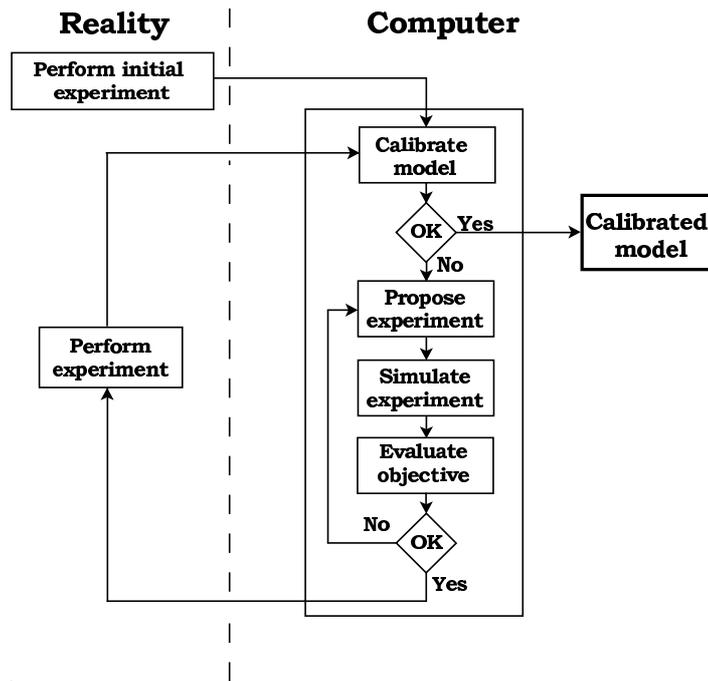


Figure 1.17: Schematic representation of the optimal experimental design procedure (adapted from Dochain and Vanrolleghem, 2001).

expressed as some scalar quantity of the FIM. Since the FIM is inversely proportional to the parameter estimation error covariance matrix, its eigenvalues and eigenvectors directly relate to the eigenvalues of the parameter estimation error covariance matrix and thus to the shape, size and orientation of the confidence region of the parameter estimates (Section 1.2.2). The eigenvalues of the FIM are inversely proportional to the size of the axes of the asymptotic confidence ellipsoids, i.e. large FIM eigenvalues correspond with small confidence ellipsoid axes and thus, with a small confidence region and accurate estimates. On the other hand, a FIM eigenvalue of 0 corresponds to an infinitesimally large confidence ellipsoid axis. In this case the experiment is non-informative and the FIM is said to be singular (it has no inverse). Different scalar properties based on FIM eigenvalues are used in order to quantify the information content of a certain experiment. The most commonly used criteria are listed in Table 1.4.

A-optimal experiments minimize the sum of the eigenvalues of the inverse FIM which corresponds to minimizing the sum of the variances or errors of the estimates of θ . In order for this criterion to be calculated, the inverse FIM is required which might cause numerical problems for nearly non-informative experiments. Therefore, the modified A criterion can be used which maximizes the sum of the FIM eigenvalues. Although this criterion only uses properties of the FIM, it also poses some problems. In the case of a non-informative experiment (an experiment where one or more of the eigenvalues equal 0) the criterion can still be maximized because one of the other eigenvalues might increase.

Table 1.4: Different optimal experimental design criteria based on FIM properties (Walter and Pronzato, 1990).

Name	Criterion
A-optimal design	$\min[\text{tr}(\text{FIM}^{-1})]$
Modified A-optimal design	$\max[\text{tr}(\text{FIM})]$
D-optimal design	$\max[\det(\text{FIM})]$
E-optimal design	$\max[\lambda_{\min}(\text{FIM})]$
Modified E-optimal design	$\min[\lambda_{\max}(\text{FIM})/\lambda_{\min}(\text{FIM})]$

$\text{tr}()$: sum of eigenvalues; $\det()$: product of eigenvalues

The D-optimal design criterion maximizes the product of the eigenvalues of the FIM and thus reduces the overall volume of the confidence region. This is illustrated in Figure 1.18. The relationship between the FIM eigenvalues and the size of the ellipse axes is also shown in this figure; the maximum eigenvalue being proportional to the smallest axis and the minimum eigenvalue to the largest axis. D-optimal experiments possess the attractive property, as opposed to the other design criteria, of being invariant with respect to any rescaling of the parameter units. Although the value of the criterion changes as a function of the parameter units, the optimal experiment remains the same. This will be illustrated later in this section.

E-optimal design corresponds to maximizing the minimum eigenvalue of the FIM and thus minimizing the largest confidence ellipsoid axis or estimation error. The modified E-optimal criterion should be used if a circular confidence region is desired. This criterion minimizes the ratio between the largest and smallest eigenvalue of the FIM, also known as the condition number, which leads to less correlated parameter estimates. Its minimum value is 1 and corresponds with spherical confidence regions. Since this criterion is only related to the shape of the confidence region and not to the volume, it is possible to obtain circular confidence regions with very large volumes. This problem is illustrated in Figure 1.19.

Once the experimental degrees of freedom, constraints and objective have been defined, the iterative search for the optimal experiment can start. This is done by simulating different experiments by varying the degrees of freedom within the applicable constraints and calculating the design criterion (scalar of the FIM). The optimal experiment is found when the selected design criterion is maximized or minimized, depending on the chosen objective. The final experiment is said to be locally optimal because its design was based on a model using certain nominal parameter values. Once the optimal experiment is found, it can be performed in reality resulting in new data. Based on these data the model can be recalibrated and the quality of the parameter estimates evaluated. If required, another iteration of the design loop can be performed potentially leading to an even better experiment.

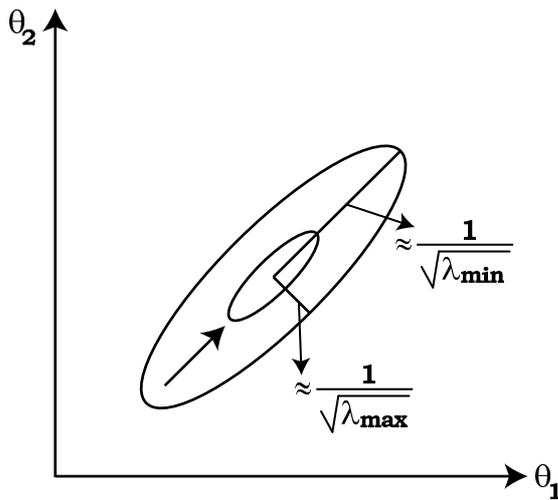


Figure 1.18: Illustration of the D-optimal design criterion which causes the volume of the confidence region to decrease.

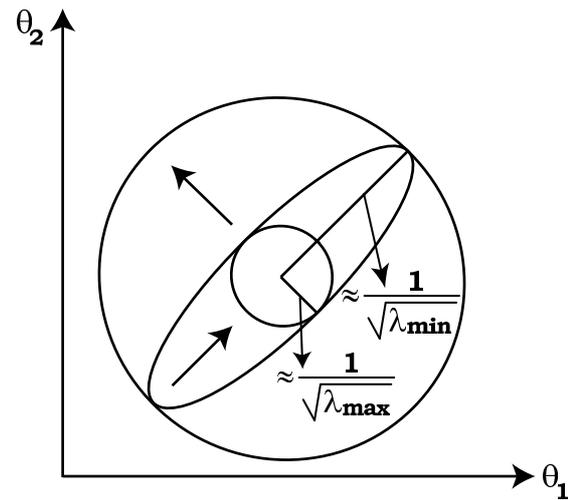


Figure 1.19: Illustration of the modified E-optimal design criterion which causes the shape of the confidence region to become as circular as possible.

A practical illustration of optimal experimental design for the fed-batch growth model is given in Figures 1.20 and 1.21. In these figures the objective function surface as a function of the feed flow rate Q_f and the feeding start time t_f is shown for the D-optimal and modified E-optimal design criteria respectively. These surfaces were constructed by defining a grid of combinations of Q_f and t_f , simulating the experiment with these experimental degrees of freedom and calculating the design criteria for each of the simulated experiments. For these simulated experiments, measurements of the substrate and biomass concentration were taken every 30 minutes and the total time of the experiment was fixed at 10 hours.

The figures indicate that the objective function surfaces show several peaks and valleys. In this case the waves in the objective function surface are caused by the fixed sampling frequency of 30 minutes. Higher measurement frequencies tend to produce more peaks but with a smaller amplitude (Rademaker, 2004). Most conventional optimization algorithms are known to be sensitive to local minima and are unlikely to find the global optimum. This clearly illustrates that the choice of optimization algorithm is a crucial factor in finding the truly optimal experiment.

The optimal value for the D criterion was found to be $4.27E+18$ and the corresponding experimental degrees of freedom are: $Q_f = 0.72 \text{ l.h}^{-1}$ and $t_f = 0.2 \text{ h}$. At this point the modified E-optimal criterion has a value of 772.25. Performing an experiment with a pump flow rate of 0.72 l.h^{-1} and starting the reactor feeding at 0.2 h would thus result in the smallest confidence region for the estimated parameters. For the modified E-optimal criterion the smallest value was found to be 25.88 which corresponds to the following ex-

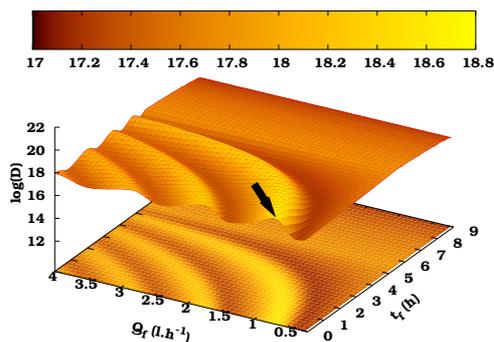


Figure 1.20: Objective function surface of the FIM D-optimal design criterion as a function of the feed flow rate Q_f and the feeding start time t_f for the fed-batch model. The optimal experiment is indicated by the arrow.

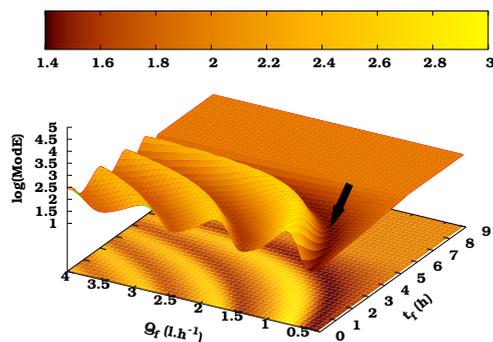


Figure 1.21: Objective function surface of the FIM modified E-optimal design criterion as a function of the feed flow rate Q_f and the feeding start time t_f for the fed-batch model. The optimal experiment is indicated by the arrow.

perimental degrees of freedom: $Q_f = 0.56 \text{ l.h}^{-1}$ and $t_f = 2.0 \text{ h}$. The value of the D-optimal design criterion for these experimental degrees of freedom is $1.55E+17$. These results show that the optimal experiment found, depends on the chosen objective. In order to decrease the volume of the confidence region a different experiment is required than if a circular confidence region is desired.

Figures 1.22 and 1.23 represent the objective function surfaces for the D-optimal and modified E-optimal criteria for the fed-batch model with units expressed as mg in stead of g . For this rescaled model the same grid of experiments as described before was investigated and the design criteria calculated. Comparing Figures 1.20 and 1.22 shows that the shape of the objective function surface has remained the same but that the value of the D-criterion has decreased by a factor 1000^2 ($1 \text{ g} = 1000 \text{ mg}$) caused by the squared contribution of sensitivities in the FIM. The optimal value for the D-criterion equals $4.27E+12$ and this corresponds to the same experimental degrees of freedom: $Q_f = 0.72 \text{ l.h}^{-1}$ and $t_f = 0.2 \text{ h}$. For the modified E-optimal criterion a change in the shape of the objective function surface can be observed, especially noticeable from the flat region that has become slightly curved (compare Figures 1.21 and 1.23). This also has implications for the optimal experiment which is now located at $Q_f = 0.48 \text{ l.h}^{-1}$ and $t_f = 1.2 \text{ h}$ and the corresponding modified E value is $1.98E+06$. This illustrates a drawback of the often used modified E-optimal design criterion. By an adequate rescaling of the model units it is possible to obtain the minimum value ($= 1$) of the modified E-optimal criterion (Petersen, 2000). This result has quite some implications for the experimental design methodology developed around such scale-dependent criteria.

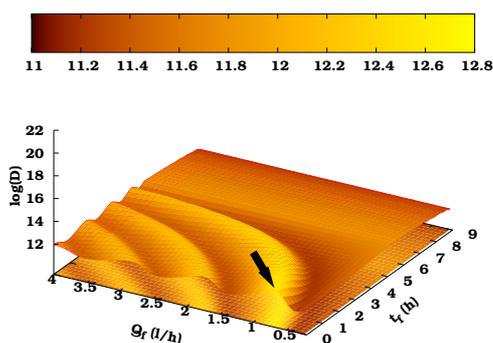


Figure 1.22: Objective function surface of the FIM D-optimal design criterion as a function of the feed flow rate Q_f and the feeding start time t_f for the fed-batch model with rescaled parameters. The optimal experiment is indicated by the arrow.

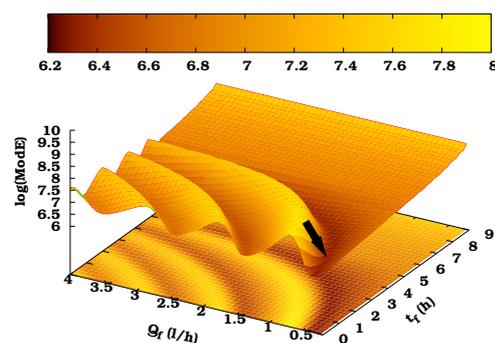


Figure 1.23: Objective function surface of the FIM modified E-optimal design criterion as a function of the feed flow rate Q_f and the feeding start time t_f for the fed-batch model with rescaled parameters. The optimal experiment is indicated by the arrow.

1.4 Problem statement and thesis outline

So far this chapter has introduced the concepts of mathematical modelling, parameter estimation and optimal experimental design. Where possible these concepts were illustrated with a simple dynamic fed-batch reactor model using Monod kinetics. At some points improvements or extensions can be made to the existing concepts. This section describes the problem statement of this thesis and the work performed in order to give answers to these problems.

Optimal experimental design involves a series of complicated steps including parameter estimation, sensitivity analysis, non-linear optimization. Because no existing simulation software package was available which combined all these methods into a single module, an extension was programmed to an existing modelling and simulation package, WEST (Vanhooren et al., 2003). A description of the modified simulation software, called EAST, will be given in Chapter 2. During the development of this software special attention was paid to make sure that the extension was as general as possible enabling it to work with all WEST models. This generality is important because many applications of optimal experimental design up till now have been case specific. Therefore much attention was paid to numerical techniques, as opposed to analytical solutions, which caused the software to be independent of the model used. Chapter 2 also introduces the models that are used to illustrate various aspects throughout this thesis.

Local sensitivity functions are one of the two components of the Fisher Information Matrix. Many methods exist to calculate these local sensitivity functions but only a few are applicable when the model equations are not

directly accessible. In this case the finite difference technique is used most often. In Chapter 3 the practical aspects of this sensitivity analysis technique are investigated and much attention is paid to the numerical aspects of this method. Beside this technique, the complex-step derivative approximation method will also be discussed in detail.

In order to quantify the quality of parameter estimates, the parameter estimation error covariance matrix should be available. This matrix can be calculated based on the Hessian matrix or the FIM. Chapter 4 illustrates the relation between the Hessian matrix and the FIM and emphasises the numerical aspects related to Hessian matrix calculations. Because the FIM is related to the parameter estimation error covariance matrix, it can also be used to decide which parameter subsets can be uniquely estimated from a given data set. The application of the FIM in this sense is investigated and compared to other techniques to select identifiable parameter subsets.

Section 1.3 already showed that the optimization problem related to optimal experimental design can be very complex. Especially when many experimental degrees of freedom and constraints are considered. The FIM criteria objective function surfaces for a simple case of two experimental degrees of freedom show that many local minima are already present (Figures 1.20, 1.21, 1.22 and 1.23). In such conditions, classical optimization algorithms only tend to converge to a local minimum. Therefore, Chapter 5 deals with the application of genetic algorithms for optimal experimental design optimization problems. Genetic algorithms are known as global optimizers and are ideal candidates for these types of optimization problems.

During the illustration of optimal experimental design it also became apparent that the optimal design criteria are often conflicting and that an experiment which is optimal for a certain design criterion is far from optimal for another design criterion. Chapter 6 therefore investigates the use of multi-objective genetic algorithms in order to come up with experiments that are close to optimal for several design criteria. The use of the experimental cost as an additional objective is also investigated.

The complicated steps related to the iterative optimal experimental design procedure require much user interaction and expert knowledge. In Chapter 7 an experimental design methodology is developed which requires only user interaction at the beginning of the procedure and is able to perform the iterative design procedure automatically including performing the experiments, calibrating the model and finding new optimal experiments.

As already mentioned, the optimal experimental design procedure results in experiments which are locally optimal. This is caused by the fact that an initially calibrated model and a corresponding initial parameter set is used as the starting point for the design procedure. It is therefore interesting to investigate what the influence of the values of these initial parameters is on the design of the experiments. In other words, Chapter 8 will deal with the design of robust experiments which account for the uncertainty of the initially specified parameter values.

Finally, Chapter 9 will draw some general conclusions and discusses some future perspectives related to optimal experimental design research.

Chapter 2

Software development and description of test models

2.1 Software development

2.1.1 The WEST modelling and simulation software

2.1.1.1 Introduction

The modelling and simulation package WEST (**W**astewater Treatment Plant **E**ngine for **S**imulation and **T**rainig) provides the modeller with a user-friendly platform to use existing models or to implement and test new models. WEST is a modelling and simulation environment for any kind of process that can be described as a structured collection of Differential and Algebraic Equations (DAE systems). So far, the application of WEST has been mainly devoted to the modelling and simulation of wastewater treatment plants although it has also been applied in other fields such as modelling of rivers (Deksissa et al., 2004), food products (Foubert et al., 2003) and trees (Steppe, 2004).

WEST++ was developed at BIOMATH, the Department of Applied Mathematic, Biometrics and Process Control of Ghent University during the period 1995-1998. It grew from several projects related to the development of modelling and simulation environments. The first of these projects (between IBM, HEMMIS and BIOMATH) resulted in the software package WEST 1.0 which was commercialized in 1993 by HEMMIS. Further developments of the WEST core functions were done at BIOMATH which resulted in WEST++ and WEST 2.0 (commercial) in 1998.

WEST++ was designed as a modular system with a strictly separated modelling and experimentation environment. Its computational back-end and model compiler were developed in C++ (although not entirely object oriented) and its graphical user interface was programmed using the scripting language Tcl and toolkit Tk (Welch, 2000). In this way, the software was platform independent and it was used on Windows, Linux, IRIX, Solaris and AIX. Figure 2.1 shows a screen shot of the WEST++ modelling and simulation environment. The upper left window shows the modelling environment, while the other windows show different parts of the experimentation environment, which is used to perform “virtual” experiments like simulation, optimization, sensitivity analysis, In the modelling environment a graphical representation of a wastewater treatment plant model is shown.

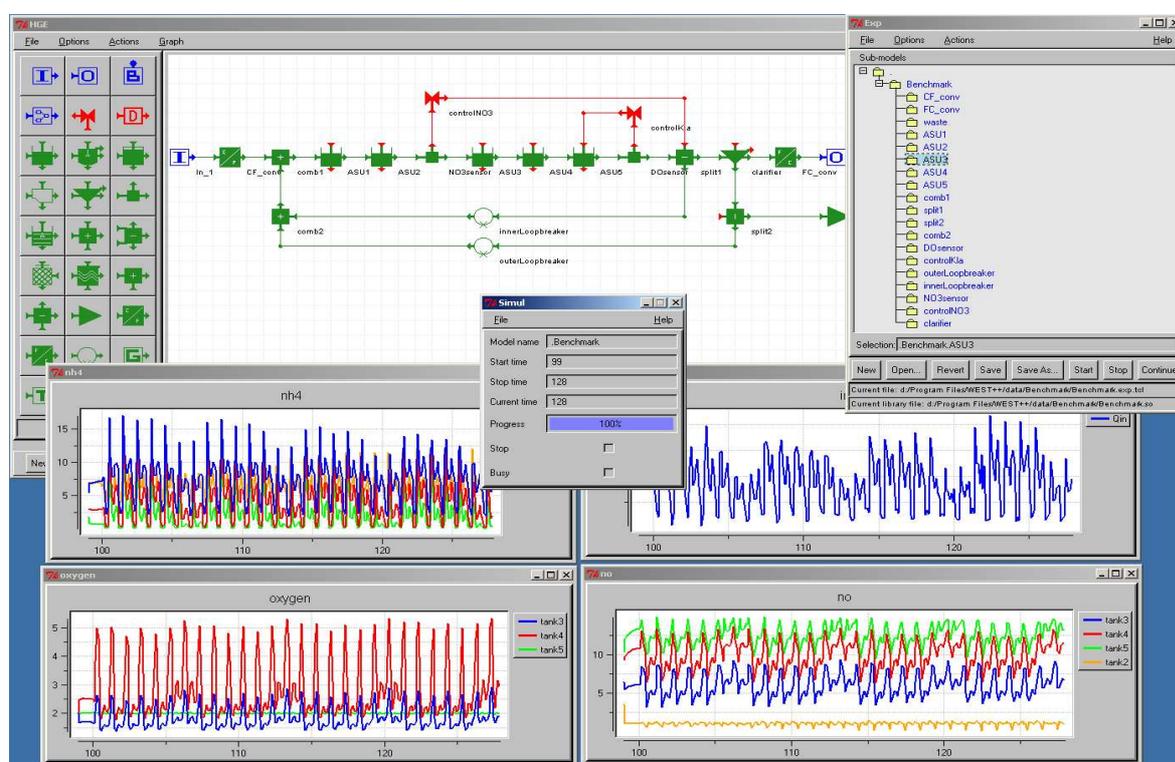


Figure 2.1: Screen shot of the WEST++ modelling and simulation software. The upper left window shows the modelling environment, while the other windows are components of the experimentation environment.

Currently, WEST is one of the most used modelling and simulation packages in Europe and South-East Asia. Because WEST strove for even higher user friendliness its graphical user interface was developed using Borland-Delphi, keeping the original WEST++ back-end and model compiler. These developments caused WEST to become restricted to the Windows platform and was thus no longer platform independent. A screen shot of the WEST experimentation environment is shown in Figure 2.2. Beside the simulation output, a graphical representation of the plant model is also shown.

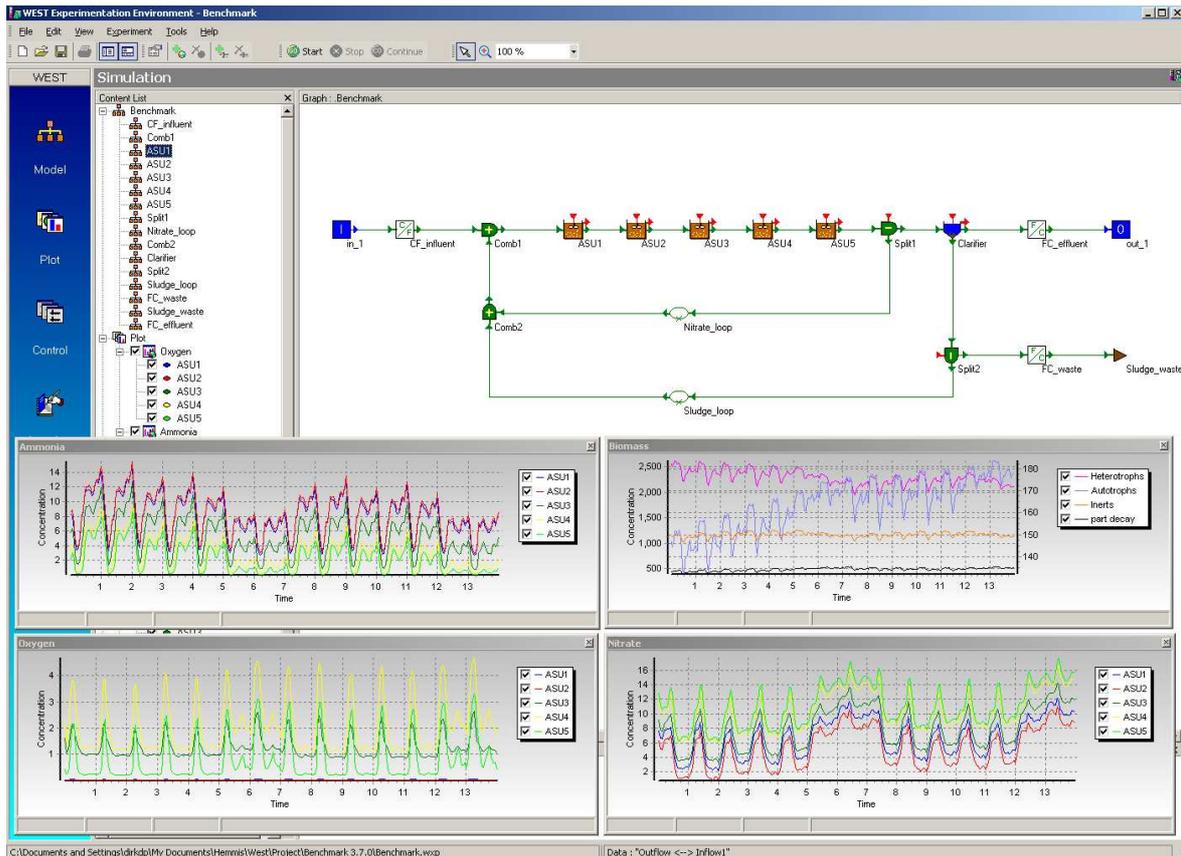


Figure 2.2: Screen shot of the WEST experimentation environment. Beside the simulation output, a graphical representation of the plant model is also shown.

2.1.1.2 WEST(++) software architecture

This section will briefly explain the general architecture of WEST(++) and the different steps that need to be followed in order to build a model and perform experiments with it. This process is graphically represented in Figure 2.3.

One of the central components of WEST(++) is the modelbase. In this modelbase, models are described in MSL-USER (MSL stands for Model Specification Language), a high level object-oriented modelling language (Vanhooren et al., 2003). The modelbase is aimed at maximal re-use of existing knowledge and is therefore structured hierarchically. All re-usable knowledge, such as mass balances, physical units, default parameter values and applicable ranges, is thus defined centrally and can be used by an expert to build new models. In this way, WEST(++) has an open structure which allows the user to change existing models and define new ones as needed.

When the modelling environment is started, the modelbase is loaded and all information relevant for the modeller is extracted from it. Using the graphical annotations in the modelbase, the “atomic” models available in the modelbase are linked to a graphical representation. A hierarchical graph editor (HGE) allows for an interactive composition of complex configurations from

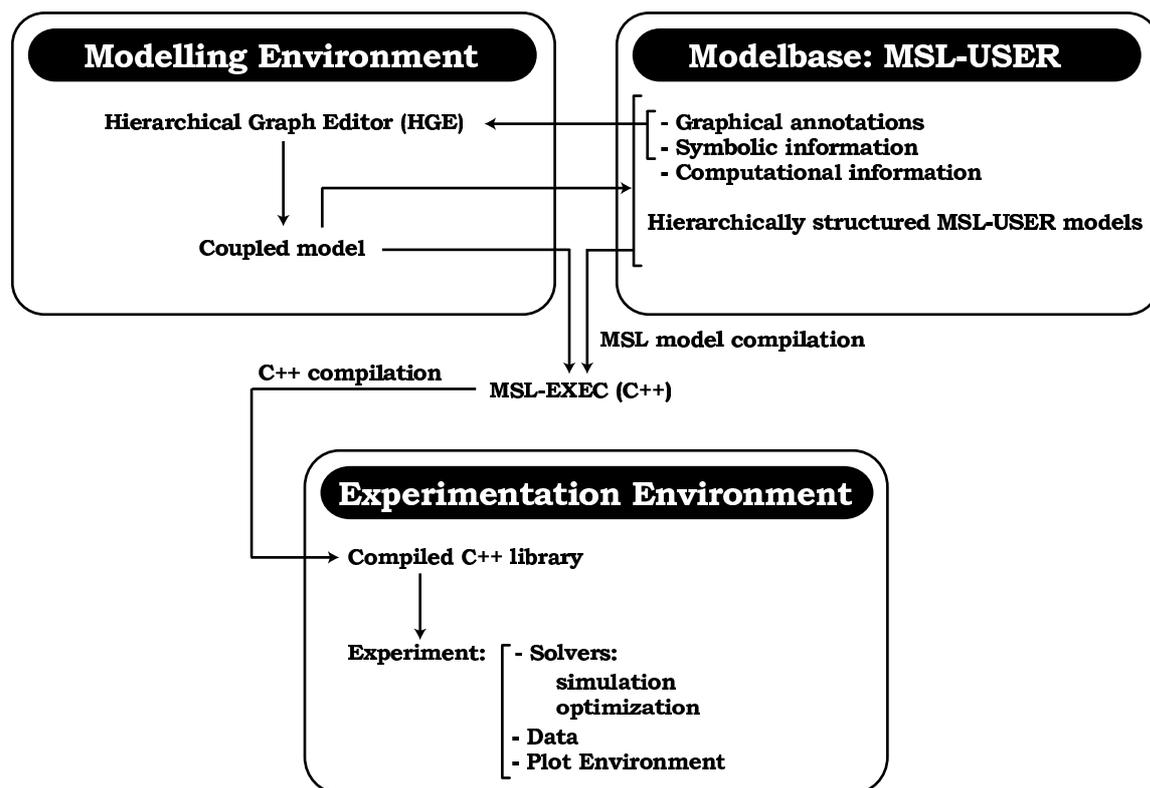


Figure 2.3: Graphical representation of the WEST(++) architecture and the different steps to be followed in order to build a model and perform experiments with it (adapted from Vanhooren et al., 2003).

these basic graphical building blocks. An example of this HGE is shown in Figure 2.1. When a configuration has been built, the HGE starts from the information extracted from the modelbase and creates and outputs a coupled model in MSL-USER, which is automatically added to the modelbase for further use in new modelling exercises. Additional information about the structure of the modelbase and the MSL-USER language can be found in Vanhooren et al. (2003).

In a next step, a model compiler is used to generate low-level MSL-EXEC (C++) code from the MSL-USER model and the atomic model representations in the modelbase. After compilation of this MSL-EXEC code, a model library is formed which can be used for execution in the experimentation environment.

In the experimentation environment the compiled model is loaded and symbolic information is retrieved from the library. Examples of such symbolic information are the model structure and listings of parameters and variables. These listings contain the units, descriptions, default values, initial values and lower and upper bounds for the parameters and variables. Once the model library is loaded, several “virtual” experiments can be performed on the model. In 1998, two types of experiments were available: simulation and optimization (parameter estimation). Since then, several other exper-

iments have been added: sensitivity analysis, scenario analysis, Monte-Carlo experiments, optimal experimental design. Several solvers, like integration and optimization algorithms, form the basis of these experiments. Together with the executable model code, these solvers produce the output data which can be plotted or written to files. More information about the experiments available within WEST(++) can be found in Vanhooren et al. (2003).

2.1.2 The EAST simulation software

2.1.2.1 Modification of the WEST(++) experimentation environment

In order to investigate some of the problems related to sensitivity analysis and optimal experimental design, two modular extensions to WEST(++) had to be programmed. Although the existing WEST(++) experimentation architecture was excellent at performing the tasks it was designed for (simulation and optimization), it became apparent, during the implementation of the sensitivity analysis module, that the back-end was not flexible enough to allow other modules to be added easily. The reason for this was twofold. First of all, the existing user interface of WEST++ was built using the string based scripting language Tcl and toolkit Tk. The user interface scripts acted as glue between the C++ simulation and optimization back-end modules. If communication was required between these modules, data had to be sent from one module through the user interface to the other module. For this communication to work, all data had to be converted into strings which made transfer of large quantities of data slow and cumbersome. Another cause of the lack of flexibility of the existing WEST(++) back-end was the “flat” C API (Application Programming Interface). Although the main programming language of WEST(++) was C++, a C-style (not object oriented) approach was used to write the computational back-end. This was required in order to interface with the Tcl/Tk graphical user interface but did not lend the code to be easily reusable.

As a solution to these problems it was decided to rewrite the WEST(++) experimentation environment and this project led to the EAST (**E**nhanced **A**cademic **S**imulation **T**ool) simulation software. Improvements were made at different levels, including the graphical user interface and the low level computational back-end. One of these improvements was the transformation of the C-style computational back-end into an object oriented form. This caused the code to be more reusable and allowed new modules to be easily created based on existing ones. At the user interface level it was decided to use Trolltech’s Qt framework (Trolltech, 1994). Qt is a complete C++ application development framework, which includes a class library and tools for cross-platform development and internationalization. In this way, EAST could be developed on one platform and can be run on several other platforms (Windows, Linux and other UNIX variants, MAC) without requiring platform-specific code. Other benefits of the Qt framework were the

availability of cross-platform thread support, XML and network support. Figure 2.4 shows the main graphical user interface window of EAST and some plots of model simulation results.

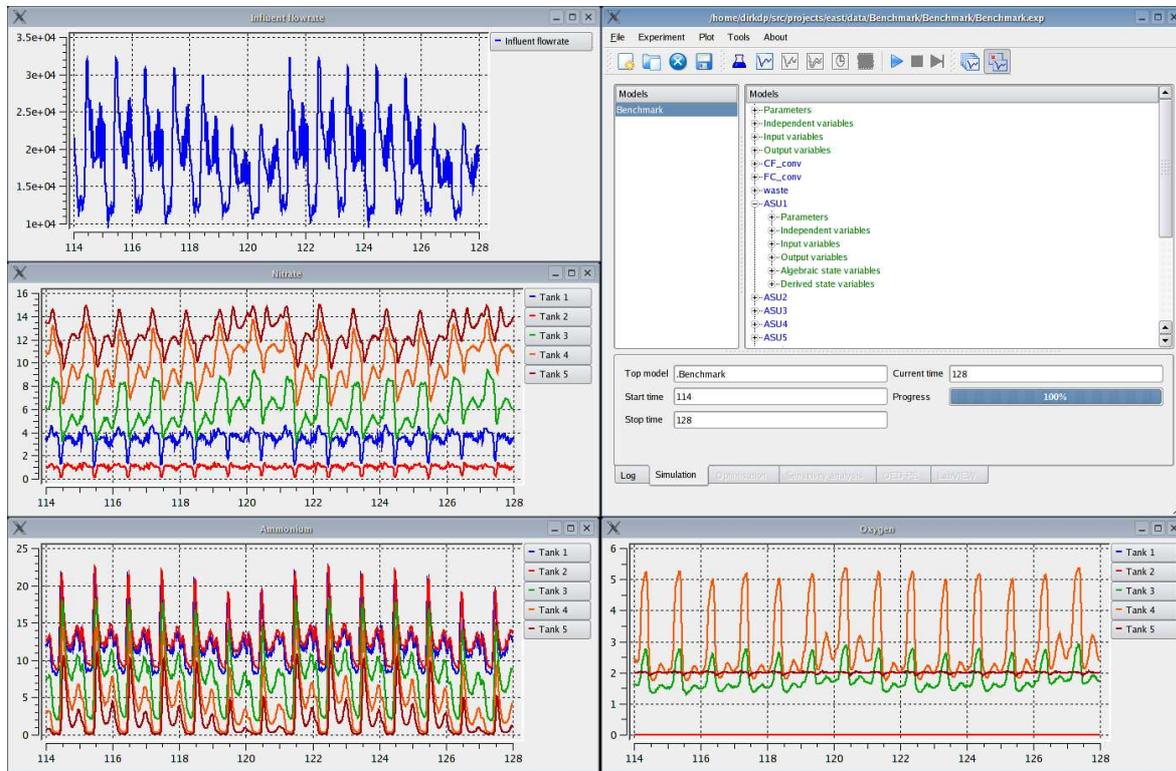


Figure 2.4: Screen shot of the EAST main window showing the model structure and plots of the model simulation results.

The next sections will briefly discuss the components of the EAST back-end and will illustrate some of its features with screen shots of the graphical user interface (front-end).

2.1.2.2 EAST experiment architecture

The top level structural component or class of EAST is the experiment (East-Experiment class) and is schematically represented by Figure 2.5 with its different components organized in boxes. The gray and white boxes respectively represent components with or without sub-components. The East-Experiment class is a container for the different types of virtual experiments available in EAST and keeps track of the currently active experiment. The available experiments are the simulation, optimization, sensitivity analysis (SA), optimal experimental design for parameter estimation (OEDPE), LabVIEW communication optimal experimental design experiments and robust optimal experimental design. Beside these functionalities, the East-Experiment class is also responsible for saving and loading experiments.

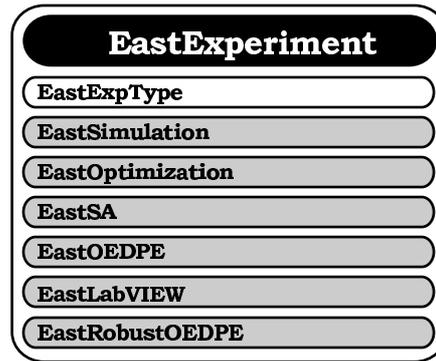


Figure 2.5: Composition of the EastExperiment structure. Gray boxes represent structural elements with sub-structures.

2.1.2.3 EAST simulation architecture

The most essential virtual experiment that can be performed in EAST is the simulation experiment (EastSimulation class), shown in Figure 2.6. This experiment allows the user to simulate the trajectories of the model variables as a function of the independent variable. In order to do so, the differential and algebraic equations have to be solved using a numerical solver or integrator. Different integrators are available within EAST (EastSimulIntegrator class). If the model consists only of algebraic equations there is no need for a complex DAE solver and the algebraic solver can be used which provides the solution of the algebraic equations for any given value of the independent variable. If the model is a DAE system, a forward integrator must be used. EAST provides a 4th order Runge-Kutta fixed step size integrator (Press et al., 1992), a 4th order Runge-Kutta adaptive step size control integrator (Press et al., 1992) and a stiff solver CVODE (Hindmarsh and Petzold, 1995). Beside selecting the integration algorithm, the input of the model also needs to be specified. This input can be supplied using an input file from which the values required at the integration time steps are interpolated. Finally, the method of output needs to be specified: file and/or plot output is available.

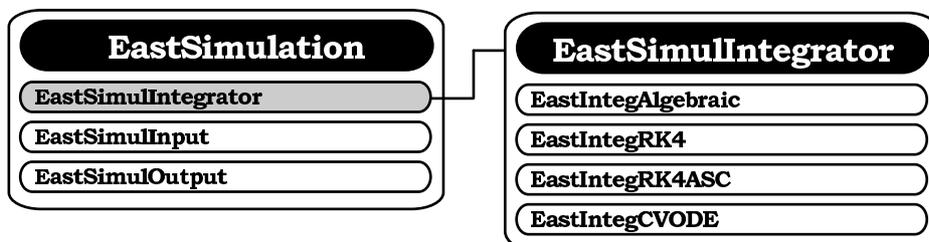


Figure 2.6: EastSimulation and EastSimulIntegrator class components.

2.1.2.4 EAST optimization (parameter estimation) architecture

In this virtual experiment, model parameters can be varied by a number of search algorithms in order to minimise the distance between a simulated trajectory and a given measured trajectory. An overview of the EastOptimization class with its sub-classes is shown in Figure 2.7. The required steps to set up a parameter estimation experiment are reflected in the structure of the EastOptimization module. First of all, the parameters that need to be estimated are selected (EastOptimParameter class). Next, the measure that will be used to quantify the difference between the measured and the simulated trajectory needs to be defined (EastOptimObjective class). This distance measure is typically a sum of squared differences between simulations and measurements, although an absolute difference can also be used. This difference can be weighted to account for measurement accuracy or differences in the order of magnitude of the different values in the objective function. The trajectory simulations, needed to calculate the objective, are provided by the EastSimulation module (discussed earlier). Once the objective is fixed, an optimization algorithm needs to be selected (EastOptimOptimizer class). Three different optimization algorithms are available in EAST: the Simplex method developed by Nelder and Mead (1965), the principal axis (Praxis) method developed by Brent (1973) and genetic algorithms based on the GAlib library (Wall, 1995). Options related to log files and the output directory also need to be specified (EastOptimOptions class). After parameter estimation, covariance information can be computed based on the parameter estimation error covariance matrix (inverse of the Hessian matrix) (EastOptimCovariance class). The Hessian matrix can be calculated using three techniques: one technique based on a quadratic approximation of the objective surface developed by Nelder and Mead (1965), another technique based on Richardson extrapolation (Press et al., 1992) and a third technique based on the combination of Richardson extrapolation and complex-step derivative calculations (Martins et al., 2003). These techniques will be further discussed in detail in Chapter 4.

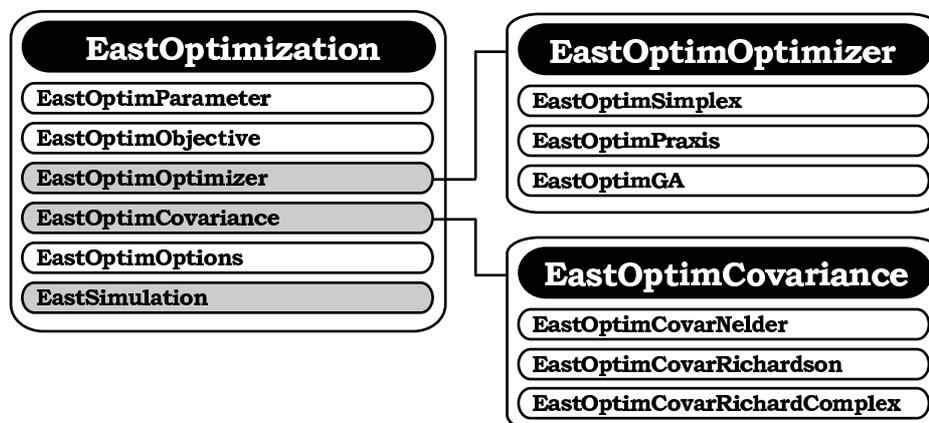


Figure 2.7: EAST optimization experiment architecture.

2.1.2.5 EAST sensitivity analysis architecture

The EAST sensitivity analysis module (EastSA class, Figure 2.8) implements the finite difference technique and the complex step approximation method to calculate trajectory sensitivity functions (further discussed in Chapter 3). These sensitivity functions show the dynamic sensitivity of selected variables with respect to selected parameters. In other words, these sensitivity functions show how much a certain variable changes if a certain parameter is slightly changed. The EastSA module makes use of the EastSimulation module in order to acquire the trajectory simulation of the variables.

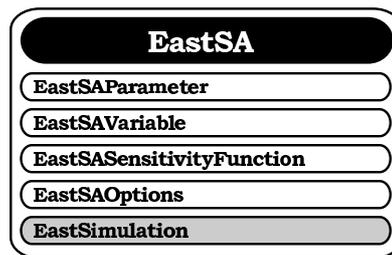


Figure 2.8: EAST sensitivity analysis experiment architecture.

The first step in performing this kind of local sensitivity analysis is the selection of the model parameters for which the sensitivity is to be analysed (EastSAParameter class). For each parameter, the perturbation factor needs to be defined. This factor determines how much the parameter will be changed in the finite difference approximation. A correct choice of this perturbation factor is crucial for obtaining correct results. Therefore the EastSA module calculates different criteria which quantify the correctness of the calculations. More detailed information about this topic can be found in Chapter 3. The graphical user interface for selecting the parameters is shown in Figure 2.9.

Next, the variables need to be selected for which the sensitivity is required (EastSAVariable class). For each variable, it can be specified at which points in time the sensitivity is required. Figure 2.10 shows the variable selection part of the sensitivity analysis graphical user interface.

The selected parameters and variables can then be combined to form sensitivity functions (EastSASensitivityFunction class). In Figure 2.11 the graphical user interface available to compose the sensitivity functions is shown together with graphical output of a selected sensitivity function. The calculated criteria values for assessing the quality of the sensitivity calculation are also shown.

Before starting the sensitivity analysis experiment, some options need to be specified (EastSAOptions). These include the type of finite difference (forward, backward, central difference or complex-step), the type of sensitivity functions (absolute, relative and/or partial relative) and several output options (Figure 2.12).

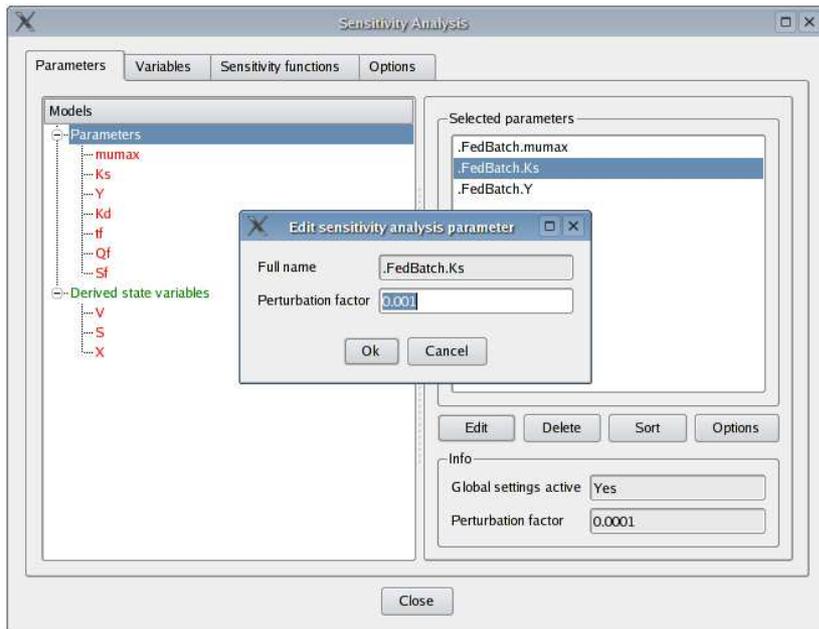


Figure 2.9: EAST sensitivity analysis parameter selection window tab.

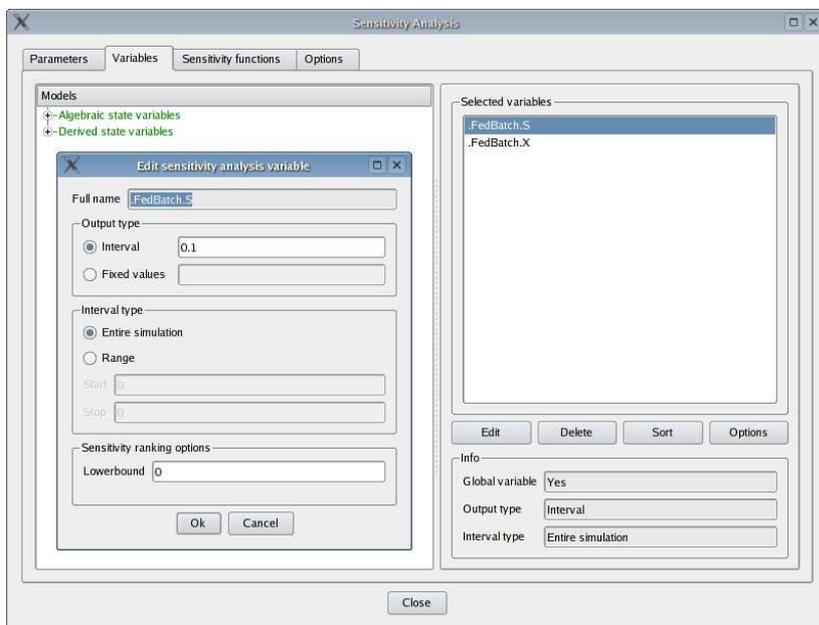


Figure 2.10: EAST sensitivity analysis variable selection window tab.

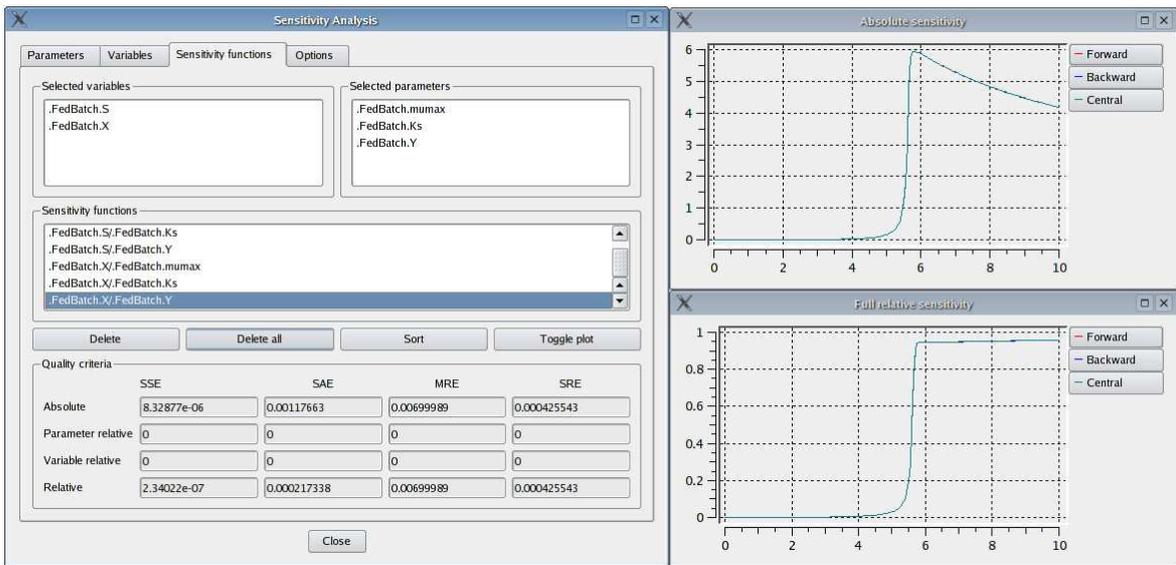


Figure 2.11: EAST sensitivity analysis sensitivity function window tab and plot windows showing an absolute and a relative sensitivity function.

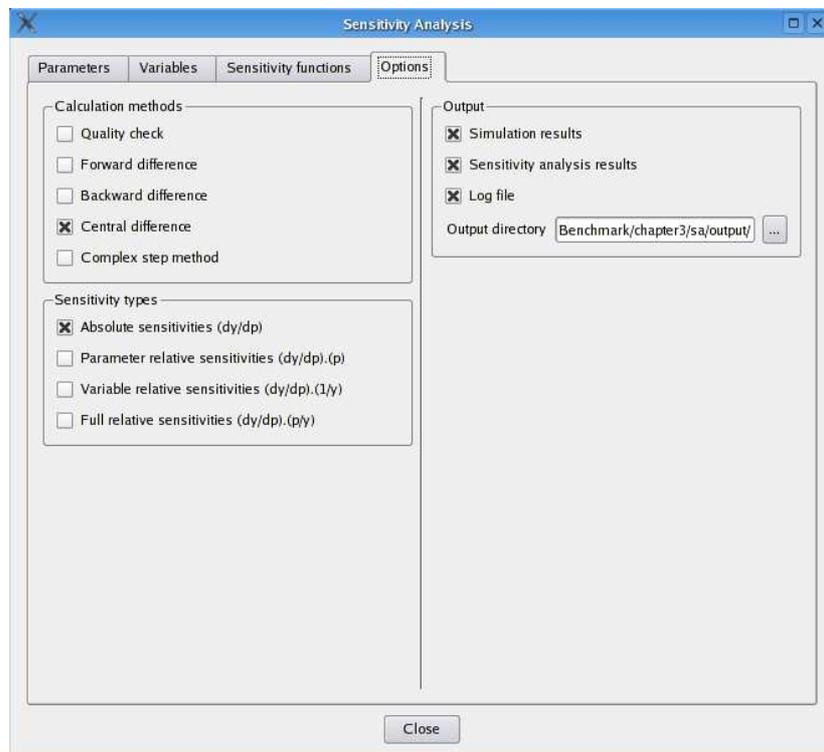


Figure 2.12: EAST sensitivity analysis options window tab.

2.1.2.6 EAST optimal experimental design for parameter estimation (OEDPE) architecture

Optimal experimental design for parameter estimation can be performed using the EastOEDPE module (EastOEDPE class, Figure 2.13). This module allows to find optimal experiments in the sense that they maximize the information content of an experiment related to the parameters expressed as a scalar property of the Fisher Information Matrix (FIM). An important component of the FIM are sensitivity functions which are provided by the EastSA module.

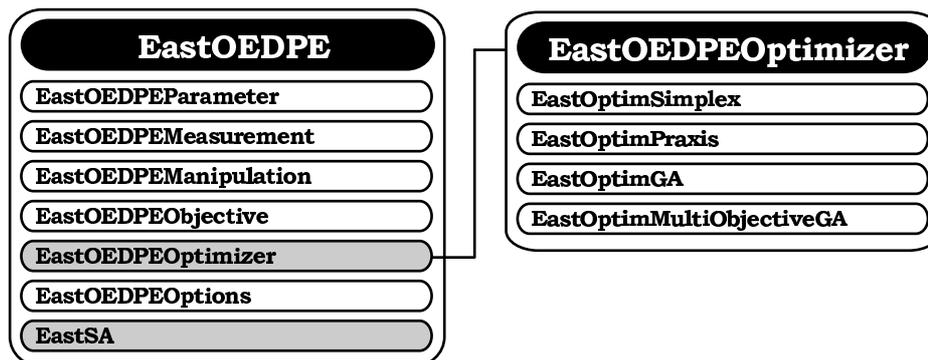


Figure 2.13: EAST optimal experimental design (OEDPE) module architecture.

An optimal experimental design experiment in EAST is started by selecting the parameters for which the experiment will be designed. The perturbation factor, needed by the EastSA module, is also specified at this point.

Next, the available measurements need to be specified using the interface shown in Figure 2.14. For each measurement several properties can be specified:

- **Measurement cost.** This is split up into two components: (1) a fixed cost independent of the number of measurement points and (2) a variable cost for each measurement point. In this way, also measurement costs can be used in the design of experiments.
- **Measurement error and range.** Two types of measurement errors can be specified: (1) relative errors, where the error characteristics are depending on the value of the measurement and (2) absolute errors, independent of the value of the measurement. Lower and upper limits of quantification can also be specified beyond which no measurements are possible.
- **Measurement type.** Three measurement types can be distinguished: (1) a measurement interval, (2) a fixed number of measurement points and (3) a set of pre-specified measurement points. A measurement interval can be specified by the user or used as a variable for optimization. In case of an optimization problem, the measurement interval

can be specified as a (constrained) continuous variable, a (constrained) discrete variable or a variable which can only take certain fixed values. Depending on this choice an appropriate optimization algorithm needs to be selected. If a measurement is specified as a fixed number of measurement points, the total number of measurement points needs to be specified and an optimization algorithm will need to be used to determine the optimal location of these points. Again, this can be formulated as a (constrained) continuous or (constrained) discrete optimization problem.

The (time-varying) correlation structure between several measurements also needs to be specified using the interface also shown in Figure 2.14. The correlations among the measurements are specified as a matrix. If no correlations exist between the measurements, the identity matrix can be used. Several measurements can be grouped into a measurement group. This can be used for optimization problems where a specified number of measurement variables needs to be found based on a group of candidate measurement variables. Examples of this type of optimization problem are finding an optimal sensor location in a case where multiple sensor locations are possible or finding the best sensor in case multiple sensors (with different associated costs or error characteristics) for the same measurement are available.

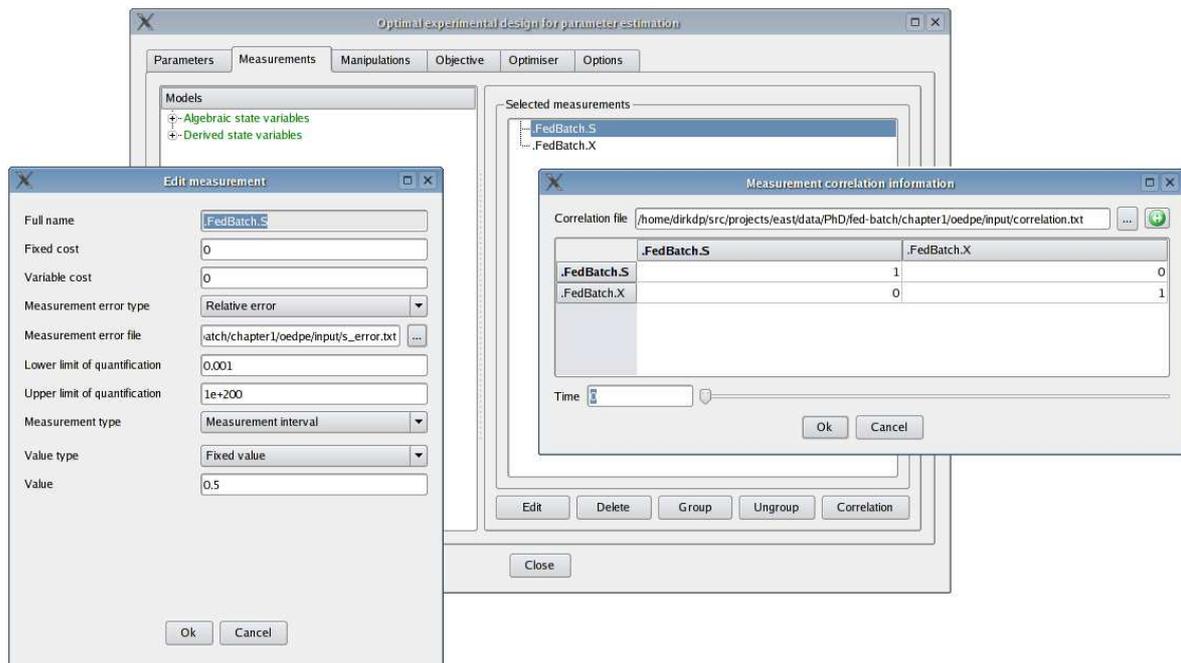


Figure 2.14: Measurement user interface of the EAST OEDPE module. The left window shows the different measurement properties and the right window shows the interface for defining the measurement correlation structure.

Beside measurements, experimental manipulations can also be specified (EastOEDPEManipulation class) using the graphical interface shown in Fig-

Figure 2.15. These manipulations are used to excite the system in order to obtain rich and dynamic data needed for an accurate parameter estimation. The value of the manipulations can be one of two types: fixed value or optimize value. In case the manipulation has a fixed value, the user needs to specify the value that will be used for this manipulation. If the manipulation is specified as an optimization variable, it has to be decided whether the value to be optimized is (constrained) continuous, (constrained) discrete or a value from a predefined list of values.

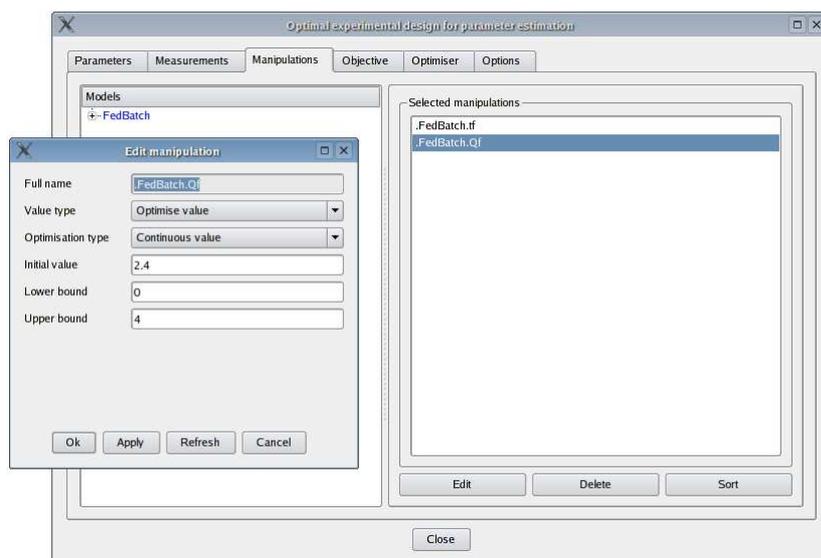


Figure 2.15: Manipulation user interface of the EAST OEDPE module. The left window shows the interface used for specifying different properties of the experimental manipulations.

The objective used to quantify the information content of the experiment is selected in the EastOEDPEObjective module with the user interface shown in Figure 2.16. In case the optimal experimental design experiment is not an optimization problem (all measurement and manipulation properties are fixed values) all criteria are calculated for the experiment and no choices have to be made. If this is not the case, a criterion has to be selected that will be used as objective function for the optimization. If several criteria are selected, a multi-objective optimization problem is created, which requires a special optimization algorithm. Also noteworthy is that the experimental cost, an often neglected experimental quality criterion, can also be selected as optimization objective. An application of multi-objective optimal experimental design, including experimental cost, can be found in Chapter 6.

An important aspect of any optimal experimental design problem is the choice of the optimization algorithm (EastOEDPEOptimizer class). This choice can be made through the user interface shown in Figure 2.17. In EAST, the sometimes complex optimal experimental design optimization problem is split up into two parts, an inner loop optimization and an outer loop optimization, depending on the properties of the experimental degrees

of freedom (see Chapter 5 for more details). All optimisation algorithms available in the parameter estimation module are also available for optimal experimental design. In addition to these, a multi-objective genetic algorithm (NSGA II: Deb, 2001) is also available for solving multi-objective optimal experimental design problems.

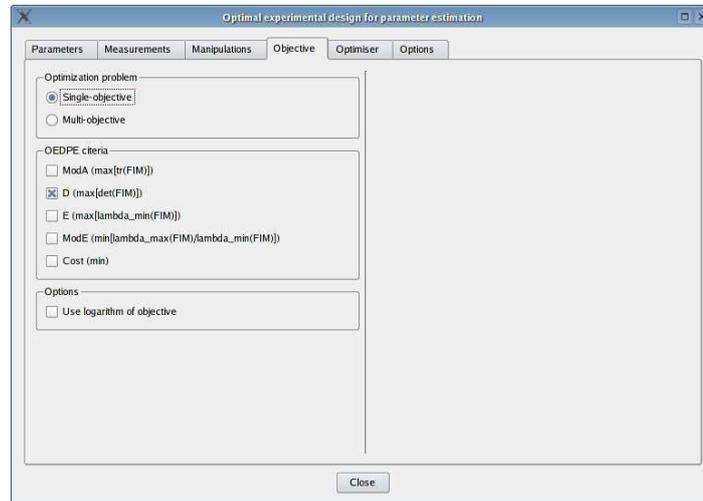


Figure 2.16: Objective selection user interface of the EAST OEDPE module.

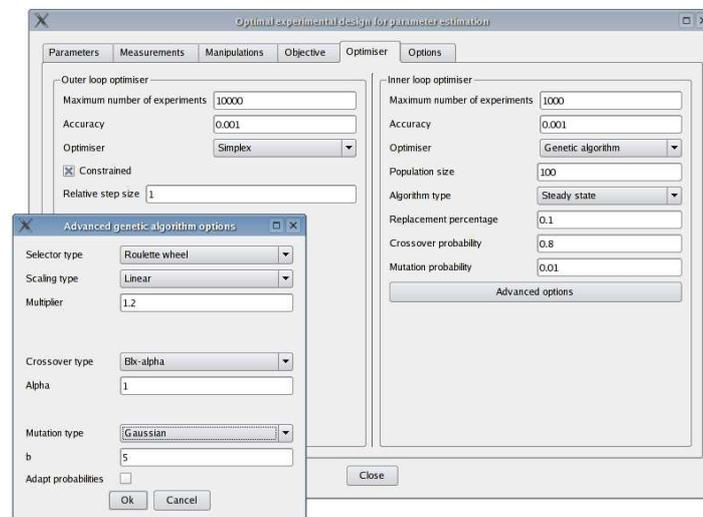


Figure 2.17: User interface for the optimizer selection in the EAST OEDPE module. The left window shows some of the advanced properties of the genetic algorithm optimizer.

Several options related to the optimal experimental design experiment can be specified (EastOEDPEOptions class) using the interface shown in Figure 2.18. These options include which type of finite difference sensitivity calculation method should be used (forward, backward, central difference or complex-step), which sensitivity function type should be used to calculate

the FIM (absolute sensitivity, relative sensitivity or partial-relative sensitivities) and various options related to the output generated by the optimal experimental design module.

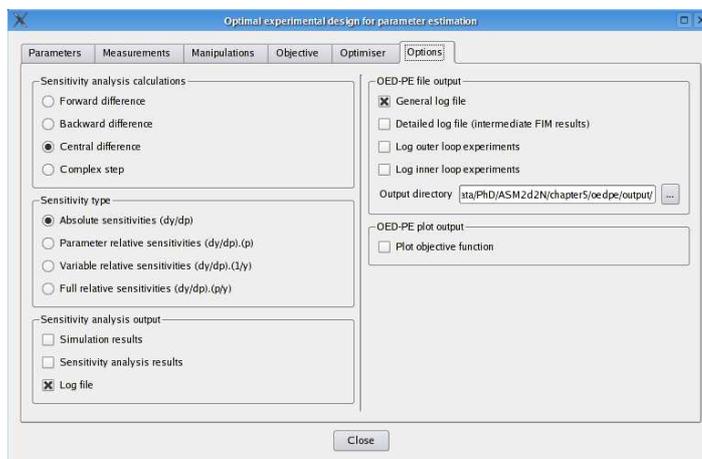


Figure 2.18: Options user interface of the EAST OEDPE module.

2.1.2.7 EAST LabVIEW optimal experimental design architecture

Automatic experimental design for parameter estimation can be performed using the EastLabVIEW module (EastLabVIEW class, Figure 2.19). This experiment combines the parameter estimation experiment (EastOptimization class) and the optimal experimental design experiment (EastOEDPE class) with a TCP-socket connection to an experimental setup controlled by LabVIEW (EastLabVIEWServer class). Beside the settings for the parameter estimation and experimental design modules, the EastLabVIEW module also has some specific settings. These can be accessed through the user interface shown in Figure 2.20. In this interface, various options related to the progress of the automatic optimal experimental design experiment can be changed, including the maximum number of parameter estimation/optimal experimental design iterations and the stopping criterion. Next to this, the type of covariance calculation (optimizer based or FIM based), network and output settings can be specified. An application of the EastLabVIEW module can be found in Chapter 7.

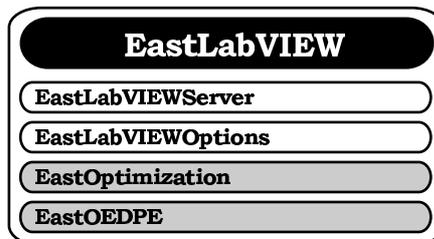


Figure 2.19: EAST LabVIEW architecture.

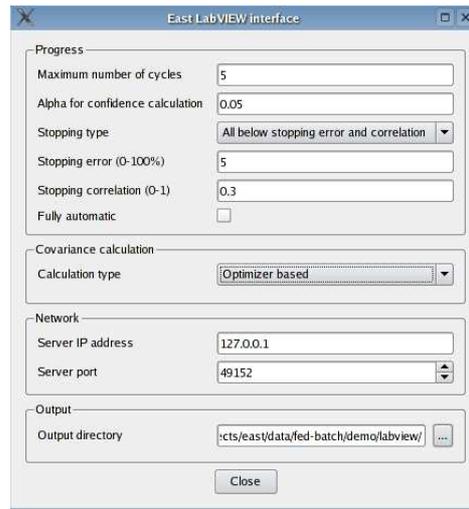


Figure 2.20: Graphical user interface of the EAST automatic optimal experimental design module.

2.1.2.8 EAST robust optimal experimental design architecture

Robust optimal experimental design for parameter estimation can be performed using the EastRobustOEDPE module (EastRobustOEDPE class, Figure 2.21). This module makes use of the optimal experimental design experiment (EastOEDPE class) in order to design experiments which are robust to misspecified parameter values. The selection of the robust experimental design algorithm and criterion can be done using the user interface shown in Figure 2.22. An application of this module can be found in Chapter 8.

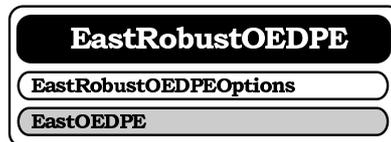


Figure 2.21: EAST robust optimal experimental design architecture.

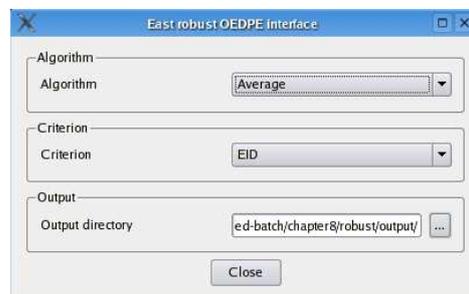


Figure 2.22: Graphical user interface of the EAST robust optimal experimental design module.

2.1.3 Unifying WEST(++) and EAST in Tornado

Since the commercialisation of WEST in 1993, parallel developments of the original WEST back-end continued by BIOMATH and HEMMIS. This led to different versions of the computational back-end and it became increasingly difficult to port functionalities between these versions. Therefore, it was decided in 2004 to start a joint project between HEMMIS and BIOMATH in order to merge all back-end developments made in WEST++, WEST and EAST. This project was named Tornado (also related to the wind and its directions) and is aimed at the development of a general computational back-end that should serve as the basis for all future development projects.

2.2 Short description of test models

Throughout this thesis various aspects of optimal experimental design will be illustrated using several models. As it is not intended to describe these models in detail, only a short outline will be given. For more details and background information, the reader is referred to the publications cited in the description of each model. A list of the models that will be described in the next sections, including the number of parameters and differential equations, is given in Table 2.1.

Table 2.1: List of test models and their associated number of parameters (not including the initial values for the variables) and differential equations.

Model	# parameters	# differential equations
Batch/fed-batch Monod	4	3
AM2	13	6
One-step nitrification	23	13
SBR	63	23
Two-step nitrification SBR	88	24
Benchmark	30	145

2.2.1 Batch and fed-batch Monod kinetics model

This simple bioprocess model describes the growth of microbial biomass on a limiting substrate. Since this model has already been described in detail in Chapter 1, only a summary of the model will be given here.

For this model, the rate at which the bacteria grow, is modelled by the well-known Monod equation (Equation 2.1).

$$\mu(S) = \frac{\mu_{max} \times S}{K_S + S} \quad (2.1)$$

This growth rate μ (h^{-1}) is related to the substrate concentration S ($g.l^{-1}$) and is characterized by a maximum growth rate μ_{max} (h^{-1}) and a saturation constant K_S ($g.l^{-1}$) at which the growth rate is half of the maximum growth rate.

Using the Monod equation to model the growth of microbial biomass, a stirred fed-batch bioreactor in which biomass is grown on a limiting substrate can be described by Equations 2.2 to 2.4. The term fed-batch indicates that the substrate is not only supplied at the start of the process (batch) but is also gradually fed to the reactor during the course of the process (fed-batch).

$$\frac{dX}{dt} = (\mu - K_d) \times X - \frac{Q_f}{V} \times X \quad (2.2)$$

$$\frac{dS}{dt} = -\frac{\mu}{Y} \times X + \frac{Q_f}{V} \times (S_f - S) \quad (2.3)$$

$$\frac{dV}{dt} = Q_f \quad (2.4)$$

In this simple model the process starts off as a batch process with a certain initial volume V_0 (l), substrate concentration S_0 ($g.l^{-1}$) and biomass concentration X_0 ($g.l^{-1}$). At a certain time t_f (h), a pump is switched on and substrate with a concentration S_f ($g.l^{-1}$) is fed to the reactor using a constant pump flow rate Q_f ($l.h^{-1}$). Equation 2.2 describes the change of the biomass concentration X ($g.l^{-1}$) with respect to time and is characterized by growth, decay and dilution. The biomass growth rate μ can be described by Equation 2.1, the decay process is modelled by a constant decay rate K_d (h^{-1}) and dilution is the result of the increasing reactor volume due to the influent feed stream. The change of substrate concentration S ($g.l^{-1}$) is given by Equation 2.3. It decreases due to microbial consumption and increases as a result of the input feeding stream. The amount of substrate which is converted into biomass is determined by the yield coefficient Y (gX/gS). The change in reactor volume (Equation 2.4) is obviously equal to the influent pump flow rate.

Examples of the time evolution of the substrate and the biomass concentration during a batch and a fed-batch experiment are shown in Figures 2.23 and 2.24 respectively.

2.2.2 Anaerobic digestion model AM2

The AM2 anaerobic digestion model was developed within the framework of the EU-TELEMAC project (IST-2000-28156) and is based on the model developed by Bernard et al. (2001). This model was extended in order to be valid over a broader pH range, especially for low pH . AM2 is a two-step (acidogenesis-methanisation) mass-balance model describing the growth of two biomass species: acidogenic bacteria X_1 ($mg.l^{-1}$) and methanogenic bacteria X_2 ($mg.l^{-1}$). In a first step, the acidogenic bacteria consume organic

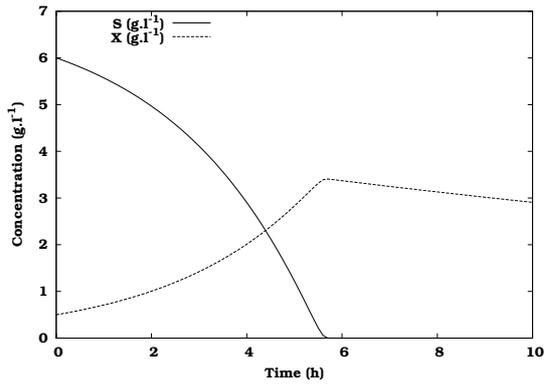


Figure 2.23: Time evolution of the substrate (S) and biomass (X) concentration during a batch experiment.

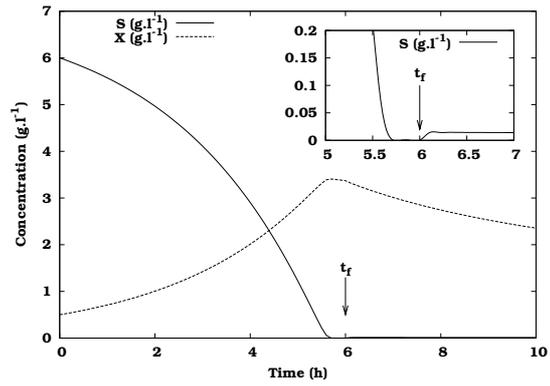


Figure 2.24: Time evolution of the substrate (S) and biomass (X) concentration during a fed-batch experiment. The inset magnifies the substrate concentration profile at the start of substrate feeding (t_f).

substrate S_1 ($mg.l^{-1}$) and produce CO_2 and volatile fatty acids S_2 ($mmol.l^{-1}$). The methanogenic bacteria use the volatile fatty acids in a second step for growth and produce CO_2 and CH_4 . Alkalinity and pH are also modelled because of their importance in anaerobic digestion processes. Equations 2.5 and 2.6 show how the pH ($-$) was modelled in AM2.

$$pH = -\log_{10} \left(\frac{\Lambda + \sqrt{\Lambda^2 + 4K_a K_b Z (C_{tic} - Z + S_2)}}{2Z} \right) \quad (2.5)$$

where Λ is given by:

$$\Lambda = K_a (Z - S_2) - K_b (C_{tic} - Z) \quad (2.6)$$

In these equations, K_a and K_b are two equilibrium constants ($mol.l^{-1}$), Z is the total alkalinity ($mmol.l^{-1}$) and C_{tic} is the total inorganic carbon concentration ($mmol.l^{-1}$).

Overall, this model consists of 6 dynamic state variables and 13 parameters (not including the initial conditions for the variables). Figures 2.25 and 2.26 show the dynamics of the model for the volatile fatty acids concentration S_2 and the total gas production Q_T (sum of CO_2 and CH_4 production rates).

2.2.3 One-step nitrification model

Based on the Activated Sludge Model No. 1 (Henze et al., 1986), a one-step nitrification model was developed where ammonium ($NH_4^+ - N$) is converted into nitrate ($NO_3^- - N$) by autotrophic bacteria, assuming that the $NH_4^+ - N$ to $NO_2^- - N$ (nitrite) conversion is the rate limiting step. The model contains

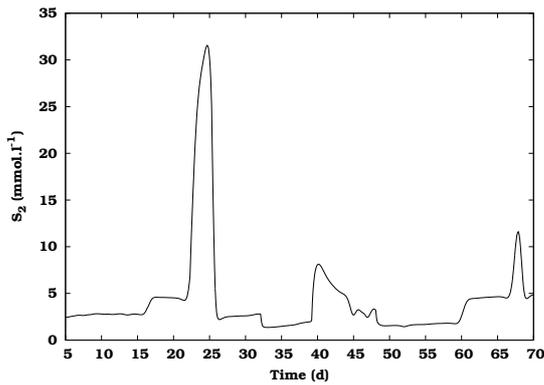


Figure 2.25: Simulated volatile fatty acids concentration S_2 for the anaerobic digestion model.

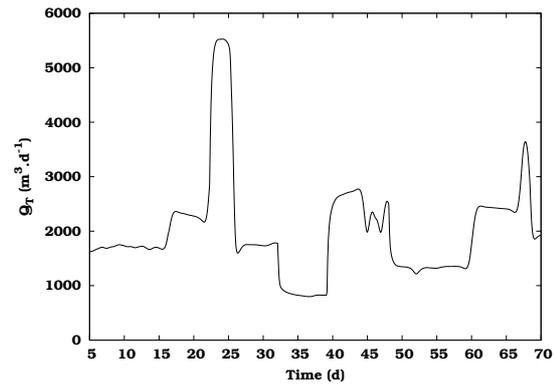


Figure 2.26: Simulated total gas production Q_T for the anaerobic digestion model.

13 dynamic state variables and 23 parameters (not including the initial conditions for the variables) and describes growth and decay of autotrophic biomass, growth and decay of heterotrophic biomass, hydrolysis of slowly biodegradable substrate and hydrolysis of particulate organic nitrogen. This model can be used to describe the oxygen uptake rate (OUR) of bacteria in, for example, a (hybrid) respirometer (Sin et al., 2003). The oxygen uptake rate given by Equation 2.7 for a batch reactor where ammonium is dosed, consists of two components: an endogenous and an exogenous part.

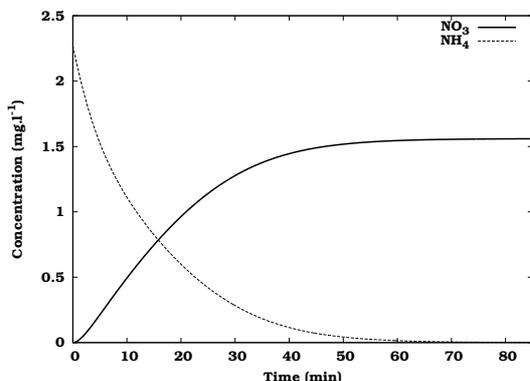
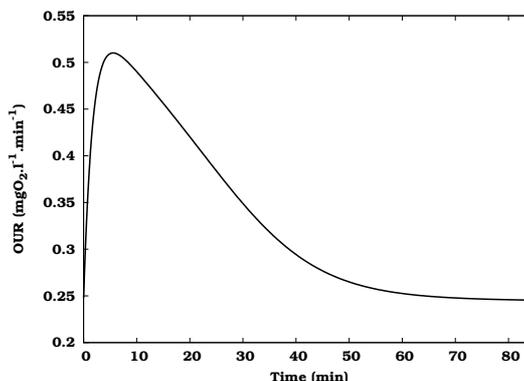
$$\text{OUR} = \underbrace{\frac{4.57 - Y_{NH_4}}{Y_{NH_4}} \times \left(1 - e^{-\frac{t-t_{pulse}}{\tau}}\right)}_{\text{exogenous}} \times \underbrace{\mu_{max,NH_4} \times \frac{S_{NH_4}}{K_{NH_4} + S_{NH_4}} \times X_{NH_4}}_{\text{endogenous}} + \underbrace{(1 - f_P) \times b_H \times X_H}_{\text{endogenous}} \quad (2.7)$$

In this equation Y_{NH_4} ($mgCOD.mg^{-1}N$) is the yield of the autotrophic biomass X_{NH_4} ($mgCOD.l^{-1}$); μ_{max,NH_4} (min^{-1}) is the maximum growth rate of the autotrophic biomass; K_{NH_4} ($mgN.l^{-1}$) is the Monod half saturation constant for ammonium; τ (min) is an OUR transient term time constant (Vanrolleghem et al., 2004); S_{NH_4} ($mgN.l^{-1}$) is the ammonium concentration; f_P ($-$) is the inert particulate fraction of the biomass and b_H (min^{-1}) is the decay coefficient of the heterotrophic biomass X_H ($mgCOD.l^{-1}$).

Parameter values and initial conditions related to the OUR calculation of the model are shown in Table 2.2. Using these values, the conversion of an initial amount of ammonium ($2.27 mgN.l^{-1}$) into nitrate was simulated. The concentration profiles of ammonium and nitrate are shown in Figure 2.27 and the corresponding OUR profile is shown in Figure 2.28.

Table 2.2: Parameter values and initial conditions related to oxygen uptake rate calculation of the one-step nitrification model.

Parameter	Value	Unit
Y_{NH_4}	0.24	$mgCOD.mg^{-1}N$
μ_{max,NH_4}	$6.9E-04$	min^{-1}
K_{NH_4}	1	$mgN.l^{-1}$
τ	2	min
f_P	0.2	—
b_H	$1.39E-04$	min^{-1}
Initial condition	Value	Unit
X_{NH_4}	40	$mgCOD.l^{-1}$
X_H	2000	$mgCOD.l^{-1}$
S_{NH_4}	2.27	$mgN.l^{-1}$
S_{NO_3}	0	$mgN.l^{-1}$

**Figure 2.27:** Simulated ammonium (NH_4^+) and nitrate (NO_3^-) concentrations for the one step nitrification model.**Figure 2.28:** Simulated oxygen uptake rate (OUR) profile for the ammonium to nitrate conversion shown in Figure 2.27.

2.2.4 Sequencing batch reactor (SBR) model

This model describes a sequencing batch reactor (SBR) removing nitrogen and phosphorous. Generally speaking, an SBR is a wastewater treatment unit where the different phases of the treatment process are conducted sequentially in time in the same reactor. This contrasts with the conventional treatment of wastewater where the different phases are performed continuously in separate reactors. More information about SBR's can be found in Wilderer et al. (2001).

The SBR model is built on the basis of Activated Sludge Model No. 1 and 2d (Henze et al., 1986, 1999). Nitrogen transformations were incorporated

following an approach similar to ASM1. The particulate nitrogen is first hydrolysed to soluble organic nitrogen and then ammonified to ammonia by heterotrophic biomass. The model consists of 63 parameters (not including the initial conditions for the variables) and 23 dynamic state variables. The parameters include the composition parameters for biomass and other components, stoichiometric parameters and kinetic parameters. More information about the model can be found in (Sin et al., 2004a).

Model calibration was based on measurements of a pilot-scale SBR reactor of 80 l treating synthetic wastewater (mimicking pre-settled wastewater) installed in the BIOMATH lab (Department of Applied Mathematics, Biometrics and Process Control, Ghent University, Belgium) (Sin et al., 2004a). The daily timing schedule for this SBR reactor consists of 4 SBR cycles (360 min per cycle), each consisting of 5 phases:

1. Anaerobic filling phase (60 min).
During this phase, wastewater is fed to the remaining biomass of the previous SBR cycle. The biomass starts to degrade the organic compounds (COD) using residual oxygen and nitrate as electron acceptor. Hence, denitrification of nitrate occurs. As soon as oxygen and nitrate are removed, phosphate accumulating organisms (PAO) also absorb short chain fatty acids and store them as polyhydroxybutyrate (PHB). Energy requirements for this process are met by the decomposition of internally stored polyphosphate, thus releasing phosphate to the bulk liquid.
2. First aerobic phase (150 min).
Aeration is switched on and COD uptake, nitrification (ammonium is converted to nitrate) and phosphate uptake are achieved. The energy for the phosphate uptake is generated by the oxidation of the internally stored PHB.
3. Anoxic phase (60 min).
Aeration is switched off which causes the conversion of nitrate to nitrogen gas (denitrification) as soon as the oxygen is depleted.
4. Second aerobic phase (30 min).
During this short second aerobic phase, aeration is switched on again in order to remove any excess COD and to strip the produced nitrogen gas. At the end of this phase, an amount of sludge is wasted in order to keep the biomass concentration in the reactor at a constant level.
5. Settling and decanting phase (60 min).
In this last phase, the sludge is allowed to settle and the supernatant clean water is decanted.

The oxygen and phosphate dynamics in the SBR for a SBR cycle with anaerobic/aerobic/anoxic/aerobic phases are illustrated in Figures 2.29 and 2.30.

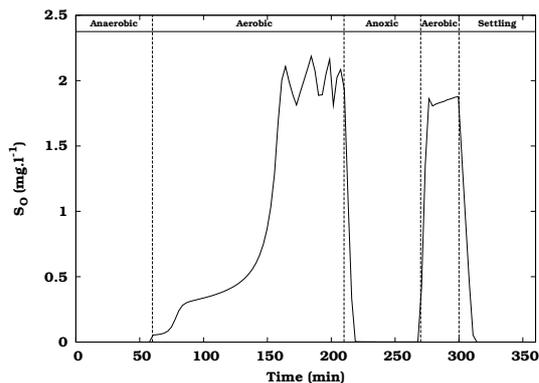


Figure 2.29: Simulated oxygen concentration in the SBR reactor. The different SBR phases are marked by dashed lines.

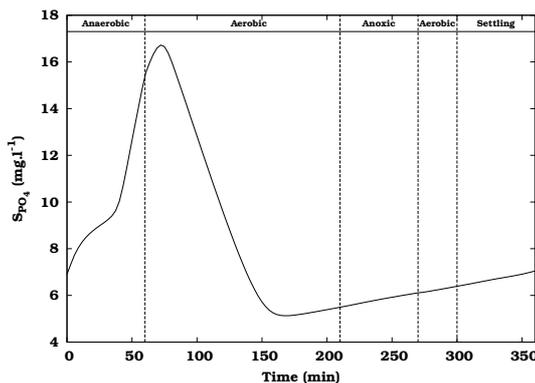


Figure 2.30: Simulated phosphate concentration in the SBR reactor. The different SBR phases are marked by dashed lines.

2.2.5 Two step nitrification/denitrification SBR model

According to a model-based optimization study performed by Sin et al. (2004a) the operational conditions of the SBR described in Section 2.2.4 were adjusted. Phases 2 (first aerobic) and 3 (anoxic) of the previous operation were replaced by 4 intermittent aeration sequences. Each sequence consists of an aeration period of 32.5 minutes followed by an anoxic period of 20 minutes. During each sequence, step feeding of the influent is performed at the beginning of the anoxic phase. The new operational conditions are shown in Figures 2.31 to 2.34. In these figures, the different SBR phases are marked by dashed lines and arrows indicate the time instances of reactor feeding.

After implementation of the new operational conditions, a measurement campaign was performed on 29/01/04. From these measurements, it became clear that nitrite build-up occurred during the aerobic phases of each SBR cycle (Figure 2.34). Therefore, the existing model, which only described one-step nitrification, was no longer valid and had to be extended with two-step nitrification and denitrification. After calibration (based on expert knowledge), this model was able to better describe the dynamics of the SBR system. The model consists of 88 parameters (not including the initial conditions of the variables) and 24 dynamic state variables. The parameters include the composition parameters for biomass and other components, temperature correction factors, stoichiometric parameters and kinetic parameters. More information about this model and its calibration can be found in Sin et al. (2004b).

Measured data and model simulations for dissolved oxygen (S_O), phosphate (S_{PO_4}), ammonium (S_{NH_4}), nitrate (S_{NO_3}) and nitrite (S_{NO_2}) are shown in Figures 2.31 to 2.34 respectively. Oxygen was measured on-line while the other measurements were performed off-line using Dr. Lange cuvette tests.

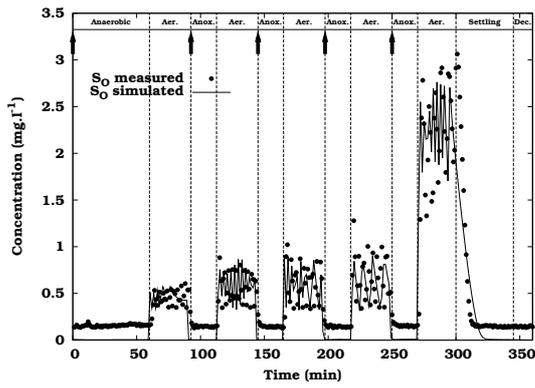


Figure 2.31: Measured and simulated oxygen concentration in the SBR reactor. Model simulations are performed using the two-step nitrification/denitrification SBR model. The different SBR phases are marked by dashed lines and arrows indicate time instances of reactor feeding.

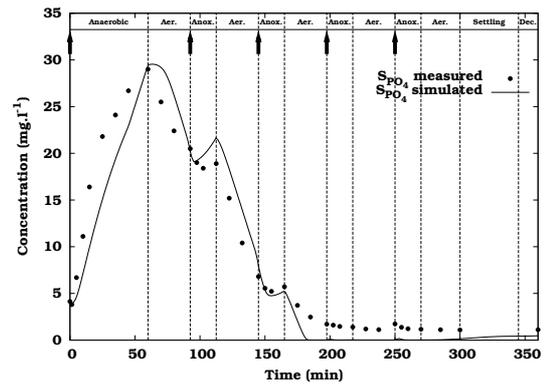


Figure 2.32: Measured and simulated phosphate concentration in the SBR reactor. Model simulations are performed using the two-step nitrification/denitrification SBR model. The different SBR phases are marked by dashed lines and arrows indicate time instances of reactor feeding.

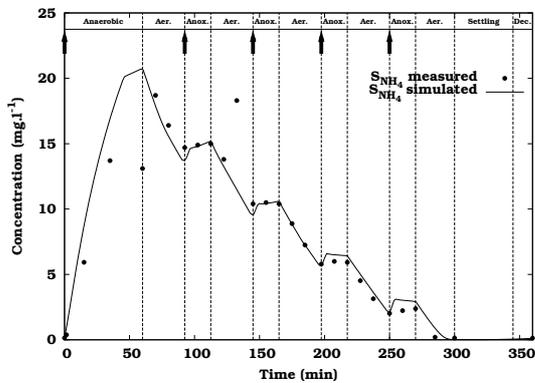


Figure 2.33: Measured and simulated ammonium concentration in the SBR reactor. Model simulations are performed using the two-step nitrification/denitrification SBR model. The different SBR phases are marked by dashed lines and arrows indicate time instances of reactor feeding.

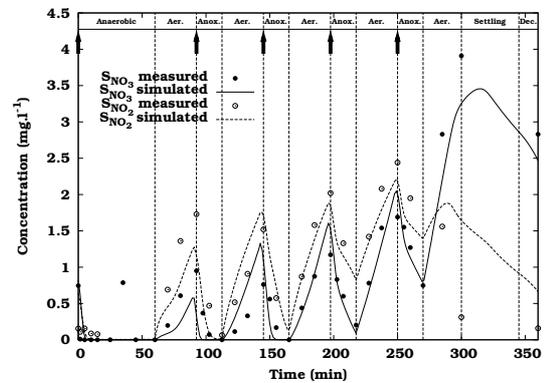


Figure 2.34: Measured and simulated nitrate and nitrite concentration in the SBR reactor. Model simulations are performed using the two-step nitrification/denitrification SBR model. The different SBR phases are marked by dashed lines and arrows indicate time instances of reactor feeding.

2.2.6 Benchmark model

The COST/IWA Simulation Benchmark model (Spanjers et al., 1998; Copp, 2001) is a wastewater treatment model that was designed to provide an unbiased basis for comparison of control strategies without reference to a particular wastewater treatment facility. It was also successfully used for comparing different simulation packages in the wastewater community. Based on the available features of the Simulation Benchmark model almost 100 scientific papers have been published world-wide (Jeppsson and Pons, 2004).

The Simulation Benchmark has five biological tanks in series and a secondary settling tank. The first two tanks are unaerated but completely mixed and the next three tanks are fully aerated. Sludge is recycled in the plant using two internal recycle lines: one nitrate recycle from the 5th to the 1st tank and one RAS (return activated sludge) recycle from the underflow of the settler to the front of the plant. Sludge that is not recycled to the plant is continuously wasted from the underflow of the settling tank. The WEST++ plant layout is shown in Figure 2.35.

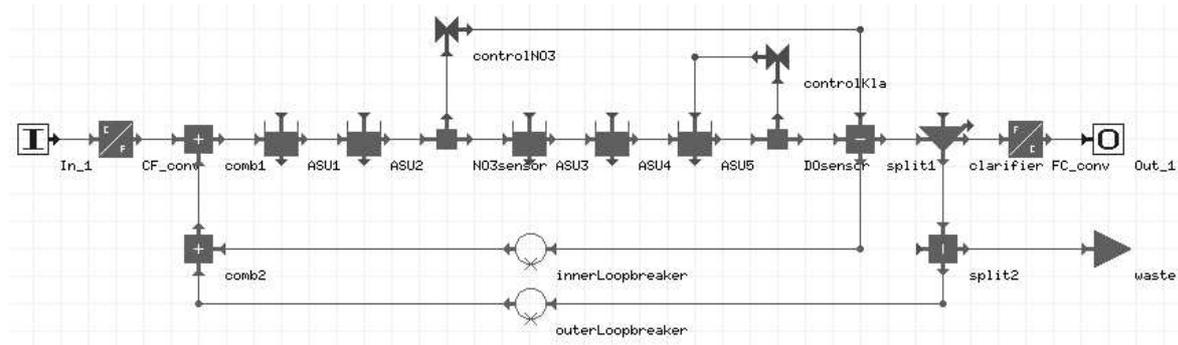


Figure 2.35: The plant layout of the Benchmark model.

The biological tanks are modelled by the Activated Sludge Model No. 1 (ASM1) (Henze et al., 1986). ASM1 has 13 dynamic state variables and 8 processes describing growth and decay of biomass, hydrolysis of organic compounds and ammonification. The secondary settler is modelled using the one-dimensional model of Takacs et al. (1991) which uses a double exponential settling velocity equation. The model consists of 145 ($5 \times 13 + 80$) differential equations. Figures 2.36 and 2.37 show the typical dynamics of the treatment plant for the effluent concentrations of nitrate and readily biodegradable substrate for a simulated period of 3 days using dry weather input conditions.

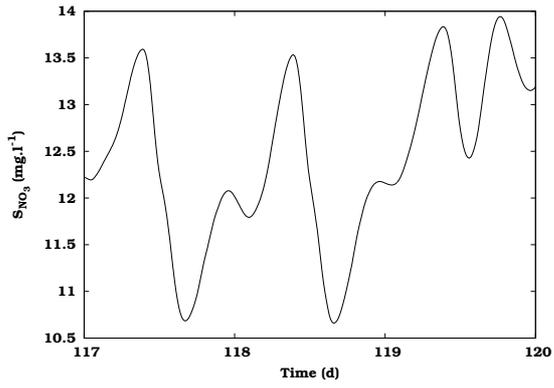


Figure 2.36: Simulated effluent nitrate concentration of the Benchmark treatment plant.

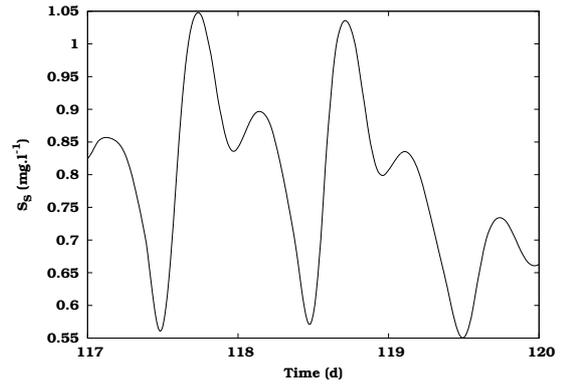


Figure 2.37: Simulated effluent readily biodegradable substrate concentration of the Benchmark treatment plant.

Chapter 3

Local sensitivity analysis

3.1 Introduction

Sensitivity analysis studies the "sensitivity" of the outputs of a system to changes in the parameters, inputs or initial conditions which are often poorly known. Sensitivity analysis can be divided into two large categories: local and global sensitivity analysis. Local sensitivity analysis methods refer to small changes of parameters, while global methods refer to the effect of simultaneous, possibly orders-of-magnitude parameter changes. Both categories will be described in Section 3.2. The main focus however will be on local sensitivity analysis techniques since sensitivity functions are one of the main components of the Fisher Information Matrix needed to perform optimal experimental design. One of these local sensitivity analysis techniques, the finite difference method, will be discussed in Section 3.3 and illustrated for some models described in Chapter 2. The finite difference technique is often used to obtain a sensitivity measure, most of the time without considering the nonlinearity of the model or the round-off error introduced by the output calculations. These influences will be analysed and discussed in detail. Another technique, called the complex-step derivative approximation method, was also implemented and will be compared to the finite difference technique. This technique has the advantage that the round-off error problem is no longer important.

3.2 Sensitivity analysis

As already introduced in Chapter 1, the general equation of the models described in this thesis is given by Equation 3.1.

$$\begin{aligned}\frac{dx}{dt} &= f(x, \theta, u, t), & x(t_0) &= x_0 \\ y &= g(x, \theta, u, t)\end{aligned}\tag{3.1}$$

In this equation x is a vector of state variables, θ a vector of parameters, y a vector of outputs, u a vector of inputs and t the independent variable. The sensitivity of an output variable y to a parameter θ can be expressed as a sensitivity function (Equation 3.2). The output variable y is called sensitive to θ if small changes in θ produce significant changes in y . On the other hand, the variable y is called insensitive to θ if changes in θ produce insignificant changes in y .

$$S(t) = \frac{\partial y(t)}{\partial \theta}\tag{3.2}$$

This partial derivative can be analytically solved if the analytical solution of Equation 3.1 is known. Unfortunately, this is rarely the case and numerical methods have to be used in order to approximate the sensitivity function (Equation 3.2). Local sensitivity analysis techniques evaluate this partial derivative at one specific set of parameter values, also called the nominal parameter set. On the other hand, global techniques evaluate this partial derivative in various points of the parametric domain. Both local and global analysis are useful in studying the behaviour of a system since each has advantages and disadvantages. For detailed reviews of existing sensitivity techniques reference is made to Turanyi (1990) and Rabitz et al. (1983). The main focus of this chapter will be the local sensitivity analysis techniques.

Various techniques for local sensitivity analysis will be described here, which are (a) the finite difference method, (b) the direct differential method, (c) the Green's function method, (d) the polynomial approximation method, (e) automatic differentiation and (f) the complex-step derivative approximation method.

(a) The simplest way of calculating local sensitivities is to use the finite difference approximation. This technique is also called the brute force method or indirect method. It is very easy to implement because it requires no extra code beyond the original model solver. The partial derivative defined in Equation 3.2 can be mathematically formulated by the equation given below (forward difference).

$$\frac{\partial y_i}{\partial \theta_j} = \lim_{\Delta \theta_j \rightarrow 0} \frac{y_i(t, \theta_j + \Delta \theta_j) - y_i(t, \theta_j)}{\Delta \theta_j}\tag{3.3}$$

This equation is only valid if we consider an infinitesimal variation (perturbation) of the parameters, inputs or initial conditions θ ($\Delta \theta_j \rightarrow 0$). Equation 3.3 shows that the application of the finite difference method requires

the solution of the model (Equation 3.1) using the nominal value of the parameters $y_i(t, \theta_j)$ and p solutions of the equations using perturbed parameters $y_i(t, \theta_j + \Delta\theta_j)$, where p is the number of parameters involved in the sensitivity analysis. It should be noted that only one parameter is perturbed at a time while all others are kept at their nominal value. The sensitivities obtained actually belong to the $(\theta + \Delta\theta/2)$ parameter set because Equation 3.3 can also be seen as the average of the sensitivities of the model output y_i at θ_j and $\theta_j + \Delta\theta_j$. If the sensitivities are desired to belong to the nominal values θ_j , Equation 3.3 should be modified into the central difference formula (Equation 3.4) which requires $2p$ solutions.

$$\frac{\partial y_i}{\partial \theta_j} \approx \frac{y_i(t, \theta_j + \Delta\theta_j) - y_i(t, \theta_j - \Delta\theta_j)}{2\Delta\theta_j} \quad (3.4)$$

(b) The finite difference technique was found to be too calculation intensive, especially in cases where sensitivities to many parameters were required. Therefore Atherton et al. (1975) developed the direct method for sensitivity analysis which is applicable to systems of ordinary differential equations. Differentiation of dx/dt of Equation 3.1 with respect to θ_j yields the following set of sensitivity differential equations. The solution of these equations results in the sensitivity functions.

$$\frac{d}{dt} \frac{\partial x}{\partial \theta_j} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial \theta_j} + \frac{\partial f}{\partial \theta_j} \quad (3.5)$$

The term $\partial f/\partial x$ of Equation 3.5 is recognised as the Jacobian J of the original system given by Equation 3.1. The differential equations (dx/dt) of Equations 3.1 and 3.5 are coupled through $\partial f/\partial x$ and $\partial f/\partial \theta$. This means that the solution of Equation 3.5 requires the knowledge of the solution of Equation 3.1 in all points where the ODE solver calculates the right-hand side of Equation 3.5. Connections between these two sets of equations can be made in one of the following ways:

1. Solve Equations 3.1 and 3.5 for every parameter θ simultaneously, which requires the solution of $(p + 1)n$ ODE's. The direct solution of this large system is inefficient. Although a solver has been built for these systems, based on the decomposed direct method (Valko and Vajda, 1984), it is not used very often.
2. Solve the couple of Equations 3.1 and 3.5, which requires the solution of $2n$ ODE's p times (for every parameter) (Dickinson and Gelinias, 1976). This version is the simplest to code, but is the least economical and might cause numerical problems due to ill-conditioned Jacobian matrices (Dougherty et al., 1979; Dougherty and Rabitz, 1980; Kramer et al., 1981).
3. The solution of Equations 3.1 and 3.5 can be decoupled. First, the model equations (Equation 3.1) are solved and the results are stored

in a table. Then Equation 3.5 is solved using the values stored in the table. If values are required at times without tabulated values, they are computed by interpolation (Atherton et al., 1975; Dougherty et al., 1979; Kramer et al., 1981, 1984). An improved version of this decoupled method was introduced by Dunker (Dunker, 1981, 1984). His method made use of the fact that the Jacobian matrix of Equations 3.1 and 3.5 is the same and that it only has to be triangulated once every time step.

(c) The difficulties of solving the large sets of differential equations led to the development of the Green's function method, also called the variational method (Hwang et al., 1978). This method makes use of the fact that the sensitivities (Equation 3.5) can be expressed in integral, rather than differential form.

$$\frac{\partial x(t)}{\partial \theta} = K(t, 0) \frac{\partial x(t)}{\partial \theta}(0) + \int_0^t K(t, \tau) \frac{\partial f(\tau)}{\partial \theta} d\tau \quad (3.6)$$

In this equation, $K(t, \tau)$ is an $n \times n$ Green's function matrix or kernel, given by:

$$\begin{aligned} \frac{d}{dt} K(t, \tau) - J(t) \times K(t, \tau) &= 0 & t > \tau \\ K(\tau, \tau) &= I \end{aligned} \quad (3.7)$$

There are a number of variations of the Green's function method and they differ among each other in the calculation of the matrix K . Probably the most used method is the GFM/AIM method (Analytically Integrated Magnus). This method approximates K by a matrix exponential (Kramer et al., 1981). In all Green's function methods, the numerical effort is proportional to the number of variables and not to the number of parameters. So this method should be preferred when the number of parameters is large compared to the number of variables. When the number of variables is much larger than the number of parameters, direct differential methods should be used.

(d) In 1983, another local sensitivity analysis technique was developed by Hwang (1983, 1985) called the polynomial approximation method. The basic idea behind this method is that the temporal behaviour of the solution of Equation 3.1 is approximated by Lagrange interpolation polynomials. From these polynomials the sensitivities can easily be calculated. However, to our knowledge, this method was never applied to a real problem.

(e) The last two decades, a technique called automatic differentiation gained a lot of attention. Automatic differentiation techniques are based on the fact that every function, no matter how complicated, is executed on a computer as a sequence of elementary functions. By applying the chain rule of differentiation repeatedly to the composition of these elementary operations, one can compute the derivative information exactly and in a completely mechanical fashion. This method produces compilable code (after parsing of

the user specified code) that evaluates derivatives up to machine precision with a minimum of human effort (Hwang et al., 1997). Recently, the applicability of this technique has been improved by using well-known object oriented concepts like derived datatypes, templates and operator overloading (Martins et al., 2003).

(f) Recently, the complex-step derivative approximation method, an unconventional technique for calculating the first derivative of any real function, was introduced to the scientific community (Squire and Trapp, 1998). In order to apply this technique, the variables and parameters of the model need to be transformed into complex variables and all model arithmetic must be replaced by its complex alternative. By applying a small step $\Delta\theta$ to the imaginary part of the parameter θ , the sensitivity function can be obtained from:

$$\frac{\partial y}{\partial \theta} = \lim_{\Delta\theta \rightarrow 0} \frac{\text{Im}[y(\theta + i\Delta\theta)]}{\Delta\theta} \quad (3.8)$$

This equation does not involve a difference operation and is thus not subjected to round-off error or subtractive cancellation, even if extremely small step sizes are used. This constitutes a tremendous advantage over the finite difference approach. A more detailed study of the technique will be given in Section 3.4.

All above techniques except for the complex-step derivative approximation method and the finite difference technique have one thing in common. They all require (complicated) manipulations of the model equations. In many studies this is not practically feasible because the models are too complicated or the model equations are not directly accessible (e.g. because they are compiled in executable commercial code). This is the basic reason why the finite difference method, although inefficient, is still very often used. The practical issues concerning this sensitivity analysis technique are discussed in the next section.

3.3 Practical aspects of the finite difference method

3.3.1 Theoretical background

Equation 3.3 forms the basis of the finite difference method and was implemented as follows:

$$\frac{\partial y}{\partial \theta_j} \approx \frac{y(t, \theta_j + \Delta\theta_j) - y(t, \theta_j)}{\Delta\theta_j} \quad (3.9)$$

where $\Delta\theta_j$ is the change of the parameter value. Practically, $\Delta\theta_j$ was implemented as the nominal parameter value θ_j multiplied by a user defined perturbation factor ξ . As will be shown later, the choice of this perturbation factor will determine the quality of the sensitivity function. Equation 3.9

is only equivalent to Equation 3.3 if the perturbation factor approaches 0. From a theoretical point of view this is correct but numerically this can never be achieved because of the limited precision of the calculations. Taking the perturbation factor too small will result in numerical inaccuracies. On the other hand, $\xi\theta_j$ should not become too large because then the non-linearity of the model will start to play an important role in the sensitivity calculations.

The practical implementation of Equation 3.9 has two major drawbacks. The resulting sensitivity function relates to the $(\theta_j + \xi\theta_j/2)$ parameter set and it does not provide any information on the quality of the sensitivity function. As already mentioned in Section 3.2, if sensitivities are required around the nominal values of the parameters then the central difference formula should be used (Equation 3.4). Although this method requires $2p$ model evaluations, it also provides additional information concerning the quality of the sensitivity function. Rather than just making two evaluations of $y(t, \theta)$ and applying Equation 3.4, the central difference is obtained by calculating (and averaging) two sensitivity functions. The first sensitivity function is calculated by increasing the nominal parameter value by $\xi\theta_j$, the second sensitivity function is calculated by decreasing the nominal parameter value by $\xi\theta_j$.

$$\frac{\partial y}{\partial \theta_{j+}} = \frac{y(t, \theta_j + \xi\theta_j) - y(t, \theta_j)}{\xi\theta_j} \quad (3.10)$$

$$\frac{\partial y}{\partial \theta_{j-}} = \frac{y(t, \theta_j) - y(t, \theta_j - \xi\theta_j)}{\xi\theta_j} \quad (3.11)$$

To calculate the centralised sensitivity function the average of both sensitivity functions is taken.

3.3.2 Selection of test models

In order to illustrate some practical issues concerning the finite difference method, four different models described in Chapter 2 will be used ranging from a very simple to more complicated models.

The first and most simple model used is the batch/fed-batch Monod kinetics model (Section 2.2.1). This model was run in batch mode ($t_f = \infty$) and the other parameters used in the model are listed in Table 3.1. Simulations were performed from 0 to 150 hours. For this model the sensitivity of both variables (substrate concentration S and biomass concentration X) to four parameters (μ_{max} , K_S , Y and K_d) is investigated (8 sensitivity functions). An example sensitivity function of the substrate concentration (S) to the maximum growth rate (μ_{max}) calculated for a perturbation factor of $1E-07$ is shown in Figure 3.1.

A more complex anaerobic digestion model (AM2) was also considered (Section 2.2.2). For this model, simulations were run from 5 to 70 days using the

Table 3.1: Nominal parameters and initial conditions for the batch Monod model.

Name	Value	Unit
μ_{max}	0.5	h^{-1}
K_S	0.02	$g.l^{-1}$
Y	0.5	–
K_d	0.03	h^{-1}
Initial condition	Value	Unit
X_0	0.0001	$g.l^{-1}$
S_0	0.01	$g.l^{-1}$
V_0	15	l

input described in Bernard et al. (2001). For the sensitivity analysis study, 3 variables were selected: the volatile fatty acids concentration S_2 , the total gas production Q_T and the pH . The sensitivity of these variables to 6 parameters was investigated: three kinetic parameters related to the volatile fatty acids (k_1, k_2 and k_3), the maximum growth rate of the methanogenic bacteria (μ_{2max}), the half saturation constant of the volatile fatty acids (K_{S_2}) and the fraction of biomass in the liquid phase (α). Combining the 3 variables and the 6 parameters resulted in 18 sensitivity functions. The sensitivity of the volatile fatty acids concentration (S_2) to the half saturation constant of the volatile fatty acids (K_{S_2}) calculated for a perturbation factor of $1E-03$ is shown in Figure 3.2.

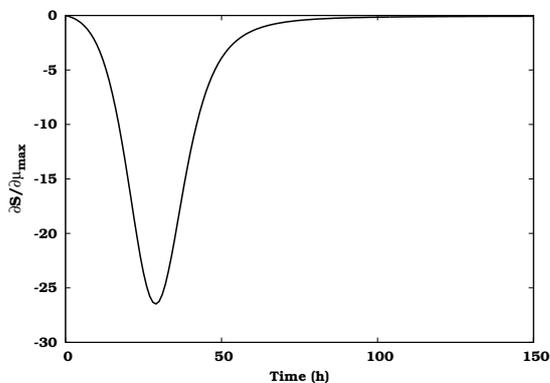


Figure 3.1: Sensitivity of the substrate concentration (S) of the Monod model to the maximum growth rate (μ_{max}) calculated for a perturbation factor of $1E-07$.

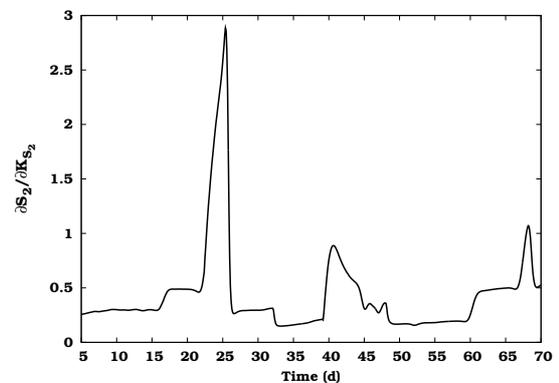


Figure 3.2: Sensitivity of the volatile fatty acids concentration (S_2) of the anaerobic digestion model to the half saturation constant of the volatile fatty acids (K_{S_2}) calculated for a perturbation factor of $1E-03$.

The third model used was the sequencing batch reactor (SBR) model (Section 2.2.4). The model was run for 0.2 days, which corresponds with one SBR cycle without settling phase. From this model, 4 variables and 6 pa-

parameters were selected to form 24 sensitivity functions. The selected variables were important nutrients measured in the reactor: ammonia (S_{NH}), nitrate (S_{NO_3}), phosphate (S_{PO_4}) and oxygen (S_O). The parameters were: the yield for heterotrophic biomass (Y_H), the hydrolysis rate constant (k_h), the anoxic yield coefficient for heterotrophs (Y_{HNO_3}), the autotrophic yield coefficient (Y_{NH}), the rate constant of polyphosphate storage (Q_{PP}) and the yield coefficient for phosphate release (Y_{PO_4}). As an example, the sensitivity of the phosphate concentration (S_{PO_4}) to the heterotrophic yield (Y_H) calculated for a perturbation factor of $1E-04$ is shown in Figure 3.3.

Finally, the more complex COST/IWA Simulation Benchmark model (Section 2.2.6) was also used and simulations were performed for days 117 to 120. From this model, 5 variables and 8 parameters were selected for the sensitivity analysis study. Three effluent variables were selected: ammonia (S_{NH}), nitrate (S_{NO_3}) and readily biodegradable substrate (S_S). In reactor 3 of the system, two biomass components were also selected: active heterotrophic biomass ($X_{B,H}$) and active autotrophic biomass ($X_{B,A}$). Four biological parameters were selected: the maximum heterotrophic growth rate (μ_{mH}), maximum autotrophic growth rate (μ_{mA}), heterotrophic yield (Y_H) and autotrophic yield (Y_A). Beside these, four physical parameters of the clarifier model were also evaluated: the clarifier surface (A_{clar}), the clarifier height (H_{clar}), the maximum settling velocity (v_0) and the non-settleable fraction (f_{ns}). The sensitivity functions of each of these variables to each of these parameter were calculated, resulting in a total of 40 sensitivity functions. Figure 3.4 shows the sensitivity function of the nitrate concentration (S_{NO_3}) to the maximum autotrophic growth rate (μ_{mA}) calculated for a perturbation factor of $1E-03$.

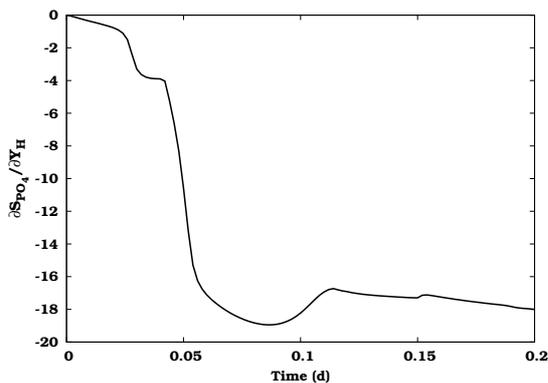


Figure 3.3: Sensitivity of the phosphate concentration (S_{PO_4}) of the SBR model to the heterotrophic yield (Y_H) calculated for a perturbation factor of $1E-04$.

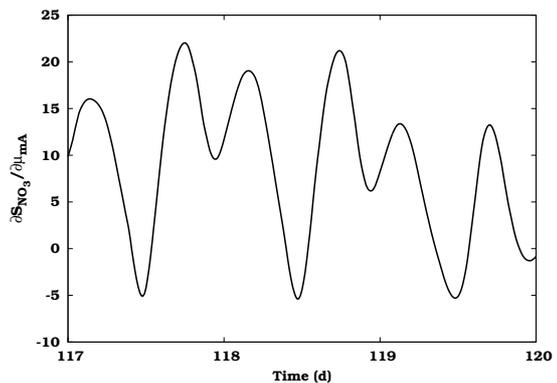


Figure 3.4: Sensitivity of the nitrate concentration (S_{NO_3}) of the Benchmark model to the maximum autotrophic growth rate (μ_{mA}) calculated for a perturbation factor of $1E-03$.

3.3.3 Quantifying the sensitivity calculations quality

To make the numerical error and the error introduced by the nonlinearity of the model as small as possible, the difference between the two sensitivity functions (Equations 3.10 and 3.11) should be minimal. This difference can be used to select the proper perturbation factor. Several criteria can be used to quantify this difference:

1. Sum of squared errors (SSE).

$$\frac{\sum \left(\frac{\partial y}{\partial \theta_+} - \frac{\partial y}{\partial \theta_-} \right)^2}{N} \quad (3.12)$$

For this criterion, the squared error between both sensitivity functions is calculated and summed over all times where the sensitivity is desired (N).

2. Sum of absolute errors (SAE).

$$\frac{\sum \left| \frac{\partial y}{\partial \theta_+} - \frac{\partial y}{\partial \theta_-} \right|}{N} \quad (3.13)$$

For this criterion, the absolute error between both sensitivity functions is calculated and summed over all times where the sensitivity is desired (N).

3. Maximum relative error (MRE).

$$\left| \frac{\frac{\partial y}{\partial \theta_+} - \frac{\partial y}{\partial \theta_-}}{\frac{\partial y}{\partial \theta_+}} \right|_{MAX} \quad (3.14)$$

This criterion returns the maximum value of the relative difference between both sensitivity functions. One should be careful with this criterion because $\partial y / \partial \theta_+$ may become 0. In these special cases the criterion returns 0.

4. Sum of relative errors (SRE).

$$\frac{\sum \left| 1 - \frac{\frac{\partial y}{\partial \theta_-}}{\frac{\partial y}{\partial \theta_+}} \right|}{N} \quad (3.15)$$

This criterion is also based on the ratio of the sensitivity functions. The ideal case is when this ratio equals 1, because then both sensitivity functions are equal. The criterion returns the sum of deviations from this ideal situation over all times where the sensitivity is desired (N). Like the MRE criterion one should be careful if $\partial y / \partial \theta_+$ becomes 0. In this special case no contribution is made to the total sum.

Figure 3.5 illustrates the use of these criteria for the sensitivity of the Benchmark autotrophic biomass ($X_{B,A}$) to the maximum autotrophic growth rate (μ_{mA}) for perturbation factors (ξ) ranging from $1E-01$ to $1E-07$. The optimal perturbation factor can be found when the criteria reach their minimum, in this case $1E-04$. For larger perturbation factors ($> 1E-04$) the nonlinearity effect of the model influences the sensitivity calculations. This is illustrated in Figure 3.6 for sensitivities calculated with a perturbation factor $1E-02$ (1% parameter change). For comparison, the sensitivity function calculated with the optimal perturbation factor $1E-04$ is also given. It is clear that $\partial y/\partial\theta_-$ differs significantly from $\partial y/\partial\theta_+$. Figure 3.5 also illustrates the effect of numerical errors when perturbation factors lower than $1E-04$ are selected. An example of this is given in Figure 3.7 for a perturbation factor of $1E-06$.

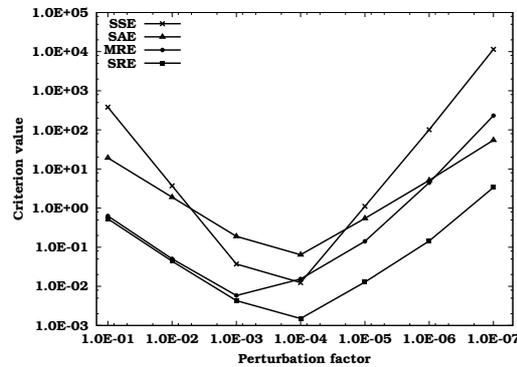


Figure 3.5: Criterion values for the sensitivity of the Benchmark autotrophic biomass ($X_{B,A}$) to the maximum autotrophic growth rate (μ_{mA}) calculated with perturbation factors ranging from $1E-01$ to $1E-07$.

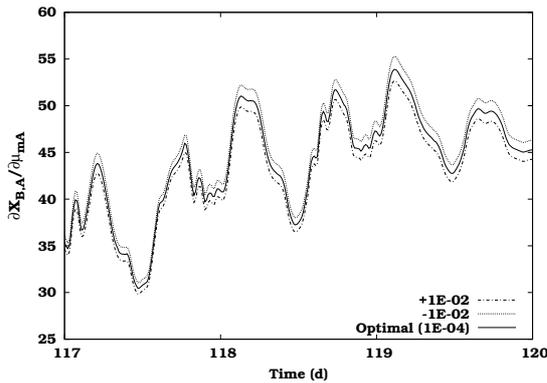


Figure 3.6: Nonlinear effect: sensitivity of the Benchmark autotrophic biomass ($X_{B,A}$) to the maximum autotrophic growth rate (μ_{mA}) calculated with perturbation factors $\pm 1E-02$ and $1E-04$.

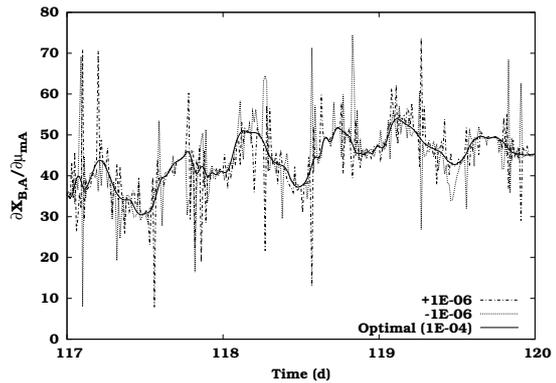


Figure 3.7: Numerical error effect: sensitivity of the Benchmark autotrophic biomass ($X_{B,A}$) to the maximum autotrophic growth rate (μ_{mA}) calculated with perturbation factors $\pm 1E-06$ and $1E-04$.

Sensitivity calculations would be quite easy if a fixed perturbation factor could be used. Therefore it was investigated whether one optimal perturbation factor exists that is adequate for all combinations of variables and parameters. For each model, Table 3.2 gives the obtained optimal perturbation factor ranges for all variables for a particular parameter. This table clearly shows that optimal perturbation factors are parameter dependent: some parameters have an optimal perturbation factor of $1E-01$, others $1E-08$. Consequently, applying a fixed perturbation factor to all parameters is not advisable. In literature, however, this is frequently done (Hoffmann et al., 2001; Marsili-Libelli et al., 2001; Reichert, 1994; Saltelli et al., 2000). We can also see that the optimal perturbation factor depends on the variable considered. For example, the optimal perturbation factors for the Benchmark μ_{mH} are within the range of $1E-02$ to $1E-04$ for the SSE criterion. This means that the sensitivity of different variables to the same parameter (μ_{mH}) resulted in different optimal perturbation factors for that parameter. However, taking the average of the indicated range as the perturbation factor results in acceptable sensitivity functions for all variables.

From Table 3.2, it can also be concluded that the different criteria indicate roughly the same optimal perturbation factors (also shown in Figure 3.5). Note that the ranges for the MRE criterion are often different from the other criteria because this criterion is based on a maximum value while all other criteria are based on averaging.

3.3.4 Comparing different quality criteria

In order to find out whether the value of a quality criterion obtained for a sensitivity calculation can be directly used to predict whether the sensitivity function calculation is valid, a closer look was taken at the criterion values obtained from the four case studies. Table 3.3 lists the minimum and maximum values of the different criteria for all calculated sensitivity functions and the minimum and maximum values of the different criteria at the optimal perturbation factors. In total, the criterion values of 634 sensitivity functions were calculated: for the Monod model, 8 sensitivity functions at 15 different perturbation factors ($1E-01$ to $1E-15$); for the anaerobic digestion model, 18 sensitivity functions at 5 different perturbation factors ($1E-01$ to $1E-05$); for the SBR model, 24 sensitivity functions at 6 different perturbation factor ($1E-01$ to $1E-06$); for the Benchmark model, 40 sensitivity functions at 7 different perturbation factors ($1E-01$ to $1E-07$).

From the table we can see that the SSE and SAE criteria have a very broad range. This is caused by the use of the absolute values of the sensitivity functions. Sensitivity functions which are small in value result in small SSE and SAE values, while sensitivities which are large result in large SSE and SAE values. For the MRE and SRE criteria we observe a much narrower range. This is because these criteria are based on relative differences between the sensitivity functions.

Table 3.2: Ranges of optimal perturbation factors for all variables to the given parameters.

	SSE	SAE	MRE	SRE
Monod:				
μ_{max}	$1E-08$	$1E-08$	$1E-07 - 1E-08$	$1E-08$
K_S	$1E-08$	$1E-07 - 1E-08$	$1E-07 - 1E-08$	$1E-08$
Y	$1E-07$	$1E-07$	$1E-06 - 1E-07$	$1E-07$
K_d	$1E-07$	$1E-07$	$1E-06 - 1E-07$	$1E-07$
Anaerobic:				
k_1	$1E-01 - 1E-02$	$1E-02 - 1E-03$	$1E-01 - 1E-02$	$1E-02 - 1E-03$
k_2	$1E-01 - 1E-02$	$1E-02 - 1E-03$	$1E-01 - 1E-02$	$1E-01 - 1E-02$
k_3	$1E-02 - 1E-03$	$1E-03$	$1E-01 - 1E-02$	$1E-01 - 1E-03$
$\mu_{2_{max}}$	$1E-03$	$1E-03$	$1E-02 - 1E-03$	$1E-03$
K_{S_2}	$1E-02 - 1E-03$	$1E-03$	$1E-02 - 1E-03$	$1E-03$
α	$1E-03$	$1E-03$	$1E-03$	$1E-03$
SBR:				
Y_H	$1E-04$	$1E-04$	$1E-03 - 1E-04$	$1E-04$
k_h	$1E-04$	$1E-04$	$1E-03 - 1E-04$	$1E-04$
Y_{HNO_3}	$1E-04$	$1E-04$	$1E-03 - 1E-04$	$1E-03 - 1E-04$
μ_{NH}	$1E-04$	$1E-04$	$1E-02 - 1E-05$	$1E-04$
Q_{PP}	$1E-02 - 1E-03$	$1E-02 - 1E-04$	$1E-02 - 1E-03$	$1E-02 - 1E-04$
Y_{PO_4}	$1E-02 - 1E-03$	$1E-02 - 1E-03$	$1E-01 - 1E-02$	$1E-01 - 1E-03$
Benchmark:				
μ_{mH}	$1E-02 - 1E-04$	$1E-03 - 1E-04$	$1E-02 - 1E-04$	$1E-02 - 1E-04$
μ_{mA}	$1E-03 - 1E-04$	$1E-03 - 1E-04$	$1E-03 - 1E-04$	$1E-03 - 1E-04$
Y_H	$1E-04$	$1E-04$	$1E-03 - 1E-04$	$1E-04$
Y_A	$1E-03 - 1E-04$	$1E-03 - 1E-04$	$1E-02 - 1E-04$	$1E-03 - 1E-04$
A_{clar}	$1E-03 - 1E-04$	$1E-03 - 1E-04$	$1E-03 - 1E-04$	$1E-03 - 1E-04$
H_{clar}	$1E-02 - 1E-04$	$1E-03 - 1E-04$	$1E-02 - 1E-04$	$1E-02 - 1E-04$
v_0	$1E-03$	$1E-03$	$1E-02 - 1E-04$	$1E-03$
f_{ns}	$1E-02$	$1E-02 - 1E-03$	$1E-02$	$1E-02 - 1E-03$

Table 3.3: Ranges of the quality criterion values for all calculated sensitivity functions and ranges of the quality criterion values at the optimal perturbation factors.

Criterion	Values for all sensitivity functions	Values at optimal perturbation factors
SSE	$1.93E-16 - 1.72E+11$	$1.93E-16 - 7.96E+02$
SAE	$8.19E-09 - 2.21E+05$	$8.19E-09 - 2.02E+01$
MRE	$3.41E-07 - 7.20E+03$	$3.41E-07 - 1.94E+01$
SRE	$1.57E-07 - 5.23E+01$	$1.57E-07 - 3.99E-01$

It can also be seen that the values of the criteria at the optimal perturbation factors show significant fluctuations. Using the upper bound of these ranges as the threshold value to decide whether a sensitivity calculation is correct or not might seem a good choice, but the next section will show that this is not the optimal one.

3.3.5 Selection of a useful criterion threshold value

As a next step, expert knowledge was used to classify all 634 sensitivity functions in two categories: "good" and "bad". This was done based on visual inspection of both $\partial y/\partial\theta_-$ and $\partial y/\partial\theta_+$ (Equations 3.10 and 3.11). When $\partial y/\partial\theta_-$ and $\partial y/\partial\theta_+$ looked identical, the sensitivity function was classified as "good" and when clear differences were noticed between $\partial y/\partial\theta_-$ and $\partial y/\partial\theta_+$ it was classified as "bad". From this classification it became clear that not only sensitivity functions calculated with an optimal perturbation factor were "good", but also some sensitivity functions which were calculated with a perturbation factor close to the optimal one.

Next, it was investigated whether a certain criterion value could be used as a threshold for automatic detection of "good" and "bad" sensitivity functions. Sensitivity functions with a criterion value lower than the threshold value are regarded as "good", sensitivity functions with a criterion value higher than the threshold value are regarded as "bad". Figures 3.8 to 3.11 show the percentages of "good" and "bad" sensitivity functions that were detected when imposing a certain threshold value for the different criteria. These percentages thus show which fraction of the "good" and "bad" sensitivity functions judged by expert knowledge was also classified as such by using the threshold value.

As an example one can consider a threshold value of $1E-01$ for the SRE criterion (Figure 3.11). From the figure we can see that with this threshold value 95 % of the "good" sensitivity functions can be detected and only 65 % of the "bad" sensitivity functions.

The intersection of both curves on the graph corresponds with the threshold value at which a maximum amount of "good" and "bad" sensitivity functions are detected correctly, e.g. for the SRE criterion a threshold value of $2.35E-02$ with 85 % detection was found. In other words, there is a 85 % chance that a new sensitivity function with a SRE criterion value smaller than $2.35E-02$ is "good" and a 85 % chance that a new sensitivity function with a SRE criterion value larger than $2.35E-02$ is "bad". This threshold value could therefore be used to judge criterion values of future sensitivity functions.

From the figures it is also clear that the different criteria have different threshold values that correspond with the maximum detection level. The maximum detection levels are also different for each criterion: 72, 71, 77 and 85 percent for the SSE, SAE, MRE and SRE criterion respectively. Hence, the SRE criterion should be preferred since its maximum detection level is the highest.

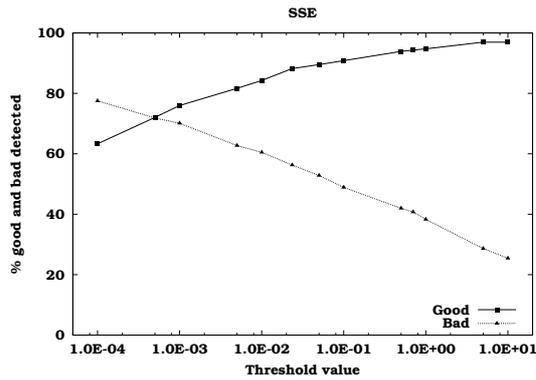


Figure 3.8: Percentages of "good" and "bad" sensitivity functions that were detected as a function of the threshold value for the SSE criterion.

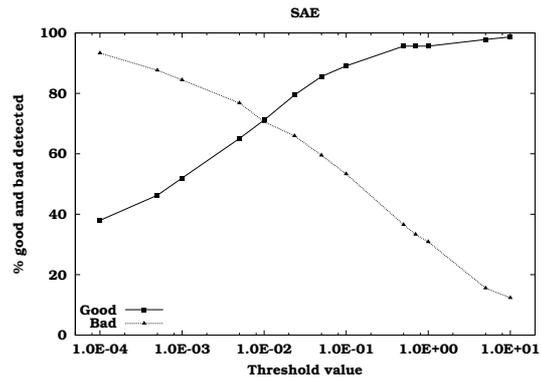


Figure 3.9: Percentages of "good" and "bad" sensitivity functions that were detected as a function of the threshold value for the SAE criterion.

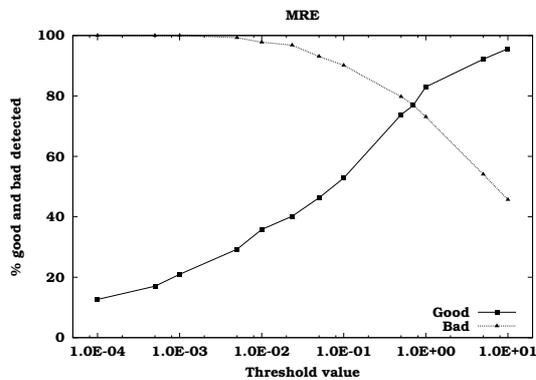


Figure 3.10: Percentages of "good" and "bad" sensitivity functions that were detected as a function of the threshold value for the MRE criterion.

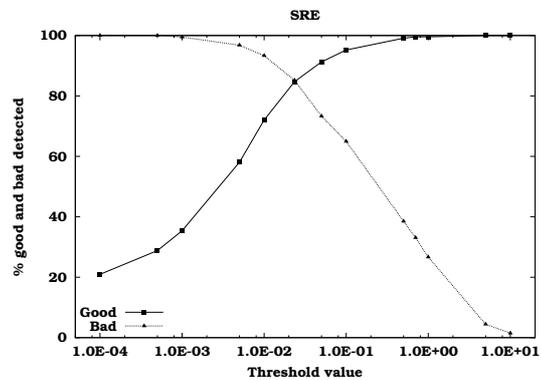


Figure 3.11: Percentages of "good" and "bad" sensitivity functions that were detected as a function of the threshold value for the SRE criterion.

The data shown in Figures 3.8 to 3.11 can also be presented in another way. At a certain threshold value, it can be analysed how many of the sensitivity functions that were classified as “good” by the expert are also detected as good using the threshold value and how many are not detected correctly. The same analysis can be done for the sensitivity functions that were classified as “bad” by the expert. This is shown in Table 3.4 for the SRE criterion using a threshold value of $2.35E-02$. From all sensitivity functions analysed (634), 229 were classified by the expert as “good” and 345 classified as “bad”. From these 229 expert “good” sensitivity functions, 194 were also detected as “good” by the automatic method (using a threshold value of $2.35E-02$), while 35 were detected as “bad” (false negatives). For the expert “bad” sensitivity functions, 345 were detected as “bad” and 60 as “good” (false positives). A useful threshold value should therefore result in a simultaneous minimiza-

tion of the number of false positives and false negatives. This is achieved at the intersection of the curves on Figures 3.8 to 3.11.

Table 3.4: Number of sensitivity functions that are classified by the expert as “good” or “bad” which are also detected (or not) based on a threshold value of $2.35E-02$ for the SRE criterion.

	Detected as “good”	Detected as “bad”
Expert “good”	194	35
Expert “bad”	60	345

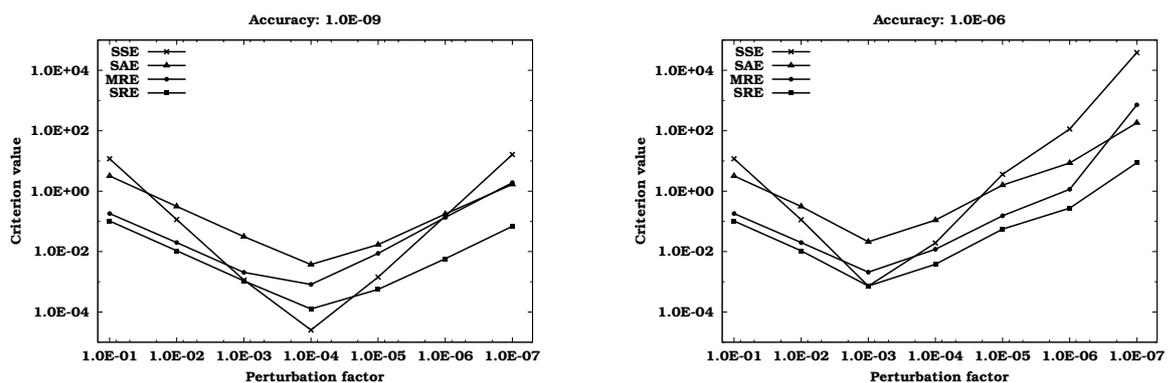
Figures 3.8 to 3.11 are based on the analysis of the combined sensitivity functions of all models (634 sensitivity functions). In order to know whether the threshold value can be used for other models we need to consider the models separately. In Table 3.5 the threshold value and the corresponding maximum detection level for each model and each criterion are shown. The threshold values for the case in which all models are combined are also listed. Looking at the different criteria we can conclude that only the SRE criterion has an “optimal” threshold value that is more or less constant for the different models. The maximum detection limit for this criterion is also larger and more constant compared to the other criteria. The only model that shows deviations is the Monod model, probably due to the limited number of sensitivity functions that were considered for this model. Based on these results we can conclude that the threshold value of $2.35E-02$ for the SRE criterion can be used to judge with a relatively high probability whether a sensitivity function calculation is “good” or “bad”.

3.3.6 Influence of integrator accuracy

All model evaluations used to calculate the sensitivity functions were performed using a 4th order Runge Kutta Adaptive Step size Control integrator (RK4ASC) (Press et al., 1992). The adaptive step size option allows the integrator to decrease or increase the step size during the integration in order to maintain a user defined accuracy. The influence of this integrator accuracy on the calculated sensitivity functions and optimal perturbation factors was investigated. Figure 3.12 shows the optimal perturbation factor for the sensitivity of the Benchmark nitrate concentration (S_{NO_3}) to the heterotrophic yield coefficient (Y_H) for different integrator settings. The left figure is the result of calculations with an integrator accuracy of $1E-09$ and the right with an accuracy of $1E-06$. Note that an accuracy of $1E-09$ has been used to produce all results shown before.

Table 3.5: Threshold values and corresponding maximum detection level for each model and each quality criterion.

	Threshold value	% good and bad detected
SSE:		
Monod	$6.20E-06$	85
Anaerobic digestion	$1.66E-03$	57
SBR	$3.00E-03$	80
Benchmark	$7.10E-04$	67
All models	$5.00E-04$	72
SAE:		
Monod	$1.10E-03$	85
Anaerobic digestion	$4.33E-03$	56
SBR	$2.10E-02$	78
Benchmark	$1.80E-02$	66
All models	$1.00E-02$	71
MRE:		
Monod	$3.00E-02$	83
Anaerobic digestion	$1.60E+00$	79
SBR	$1.20E+00$	80
Benchmark	$9.60E-01$	74
All models	$7.00E-01$	77
SRE:		
Monod	$3.00E-03$	92
Anaerobic digestion	$2.40E-02$	82
SBR	$4.70E-02$	80
Benchmark	$2.80E-02$	84
All models	$2.35E-02$	85

**Figure 3.12:** Optimal perturbation factor for the sensitivity of nitrate (S_{NO_3}) to the heterotrophic yield (Y_H) for different integrator accuracies: $1E-09$ (left) and $1E-06$ (right).

From both figures we can conclude that the accuracy has an influence on the optimal perturbation factor, $1E-04$ for an accuracy of $1E-09$ and $1E-03$ for an accuracy of $1E-06$. The accuracy also has an influence on the criterion values. All criterion values are lower for the $1E-09$ accuracy and thus result in better sensitivity functions. For other models, parameters and variables this influence has also been demonstrated. It should also be noted that higher integrator accuracies result in longer calculation times.

3.4 The complex-step derivative approximation method

3.4.1 Theoretical background

Computing sensitivity derivatives using complex variables was first suggested by Lyness and Moler (1967). For some reasons, such as the inability of compilers to deal with complex arithmetic, this technique was not exploited much until Squire and Trapp (1998) reintroduced it. Since then, the technique has been applied several times. Some fields of application are: aeronautics and astronautics (Martins et al., 2000, 2001, 2003), engineering (Perez-Foguet et al., 2000; Vatsa, 2000) and chemical kinetics (Butuk and Pemba, 2003). The specific application of this technique in this chapter deals with the calculation of dynamic local sensitivity functions of models consisting of algebraic and differential equations.

The basic equation for the complex-step derivative approximation method (Equation 3.8) can be derived using the Taylor series expansion of a function in terms of complex variables:

$$y(\theta + i\Delta\theta) = y(\theta) + i\Delta\theta \frac{dy}{d\theta} - \frac{\Delta\theta^2}{2} \frac{d^2y}{d\theta^2} - \frac{i\Delta\theta^3}{6} \frac{d^3y}{d\theta^3} + hot \quad (3.16)$$

where *hot* are the higher order terms of the expansion. Isolating the imaginary part of this equation leads to:

$$Im[y(\theta + i\Delta\theta)] = \Delta\theta \frac{dy}{d\theta} - \frac{\Delta\theta^3}{6} \frac{d^3y}{d\theta^3} + hot \quad (3.17)$$

For very small imaginary steps $i\Delta\theta$, the third and higher order terms of the expansion become negligible and the derivative can be written as:

$$\frac{dy}{d\theta} \approx \frac{Im[y(\theta + i\Delta\theta)]}{\Delta\theta} \quad (3.18)$$

Notice that this equation avoids the subtraction of nearly equal numbers, as is the case for the standard finite differencing methods, and therefore does not exhibit the accuracy problems associated with small step sizes.

In order for Equation 3.18 to be valid, the complex function y needs to be analytic which means that the function must be complex differentiable. Some

complex functions may fail to be analytic due to the presence of singularities (e.g. an infinite derivative) and discontinuities (e.g. “if” statements). Martins et al. (2001) studied this problem and found that the complex-step derivative approximation method still produced accurate derivatives in the neighbourhood of the singularities and discontinuities.

3.4.2 Practical implementation issues

Implementation of the complex-step derivative approximation method involves the conversion of the source code floating point-value functions into their complex equivalent, i.e. the functions need to be modified in such a way that they can accept complex arguments. To be practically usable, this process should be as automatic as possible because manually changing the source code not only is a tedious task, but may also introduce coding errors. The most elegant way of automatic implementation is the use of derived data types and operator overloading. With this technique the data type of the variables in the functions is replaced by a new complex data type and all operators and arithmetic functions are redefined for this new data type. Fortunately, FORTRAN and C++ natively support these functionalities, making this technique easily implementable. Martins et al. (2003) provide a single C++ file (`complexify.h`) that needs to be included with the source code. This file contains the definition of a new complex C++ class in which all standard operators (e.g. `+`, `-`, `*`, `/`, ...) and mathematical functions (`sin`, `cos`, `fabs`, ...) are overloaded, mostly calling the standard C complex arithmetic functions (defined in `complex.h`). Because no further changes to the source code need to be made, this technique is almost as straightforward to implement as the finite difference method.

The data type of the variables in WEST(++) MSL-EXEC models is the *RealType* which corresponds to the C++ *double* type (double precision floating point number). In order to “complexify” any WEST(++) MSL-EXEC model, the *RealType* data type should be redefined as a *ComplexType* and the `complexify.h` header file included in the model code. Next, the model is recompiled and is ready to be used. Beside these minor changes, the solvers which integrate the differential equations of the model also need to use the new complex data type. This requires adapting and recompiling some of the back-end code but this is of no concern to the user. From a practical point of view, this method could easily be made transparent to the user by switching on the fly between the original and the complex equivalent of the model and the solvers.

Performing a sensitivity analysis using this technique in EAST can be summarized as follows:

1. Select variables (y) and parameters (θ) and compose the sensitivity functions.
2. Apply a small perturbation $\Delta\theta$ (e.g. $1E-20$) to the imaginary part of one parameter.

3. Run the simulation and retain the trajectories of the imaginary part of all required variables.
4. The sensitivity functions are then calculated by dividing each point of these trajectories by the perturbation $\Delta\theta$.
5. Repeat steps 2-4 for all parameters.

Although this technique only requires p simulations compared to $p + 1$ simulations required for a forward difference and $2p$ simulations for a central difference, its major drawback is an increased simulation time due to the complex arithmetic. A comparison of the computational cost between the finite difference and the complex-step derivative approximation technique will be given in Section 3.4.4. An additional drawback of the complex-step derivative approximation method is that the amount of memory used is approximately doubled due to the introduction of the new complex data type. For the bioprocess models discussed in this thesis, this poses no problems since most large models never grow beyond a few MB.

3.4.3 Numerical example

In this section the influence of different perturbation factors on the accuracy of the sensitivity calculations will be studied for the complex-step derivative approximation method and compared to the forward and central difference technique. The sensitivity function that will be used for this purpose is the sensitivity of the volatile fatty acids concentration (S_2) of the anaerobic digestion model (AM2) to the half saturation constant of the volatile fatty acids (K_{S_2}). This sensitivity function was calculated for different perturbation factors, ranging from 1 to $1E-17$, using the complex-step derivative approximation method, the forward difference and the central difference technique. The model equations were solved using the Runge Kutta 4 Adaptive Step size Control integration algorithm (RK4ASC) (Press et al., 1992) using an accuracy of $1E-06$. In order to quantify the calculation accuracy of each calculated sensitivity function, the sum of relative errors with respect to a reference sensitivity function was calculated. The reference used for this purpose was the sensitivity function obtained from the complex-step derivative approximation method using a perturbation factor of $1E-30$.

In Figure 3.13 the calculated sum of relative errors as a function of the different perturbation factors is shown for the complex-step derivative approximation method, forward difference and central difference sensitivity calculations. As expected, the error of the finite difference methods initially decreases (for perturbation factors 1 to $1E-03$) to reach a minimum value for a perturbation factor of $1E-03$. For smaller perturbation factors, the error increases again due to the subtractive cancellation errors. These results agree with the general findings of Section 3.3. Also notice that, as expected, the central difference reaches a lower error than the forward difference.

For the complex-step derivative approximation method it is clear that the errors decrease with decreasing perturbation factor. At a perturbation factor of $1E-07$ the error stabilizes at a value of $\pm 1E-15$ which corresponds to the accuracy of a double precision floating point number. This value is maintained for perturbation factors as low as $1E-307$ ($1E-308$ being the smallest possible non-zero double precision floating point number). Figure 3.14 shows the calculated sensitivity function based on the complex-step derivative approximation method for a perturbation factor of $1E-15$. Notice that the slope between the perturbation factor and the sum of relative errors for the complex step approximation method is 2 since the complex step derivative approximation method is a second order approximation to the derivative. This can be derived by dividing Equation 3.17 by the perturbation factor $\Delta\theta$. For a central difference the slope is also 2 which is, however, not clear from Figure 3.13. In contrast to that, the expected slope of 1 for the forward difference is shown in Equation 3.17.

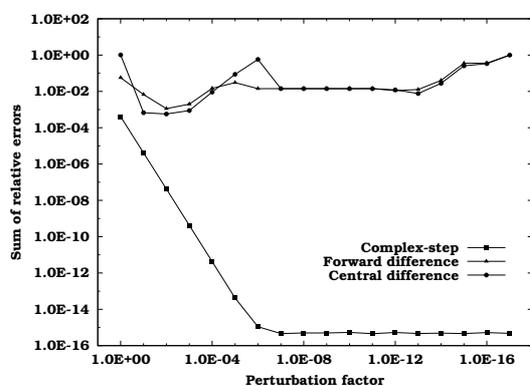


Figure 3.13: Sum of relative errors of the sensitivity function $\partial S_2/\partial K_{S_2}$ of the anaerobic digestion model for the complex-step derivative approximation method, forward difference and central difference.

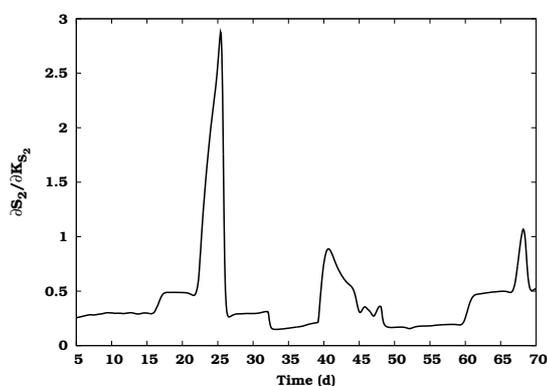


Figure 3.14: Sensitivity of the volatile fatty acids concentration (S_2) of the anaerobic digestion model to the half saturation constant of the volatile fatty acids (K_{S_2}) calculated using the complex-step derivative approximation method with a perturbation factor of $1E-15$.

3.4.4 Computational cost comparison

As already mentioned in Section 3.4.2, the major drawback of the complex-step derivative approximation method is an increase in the computational cost due to the complex arithmetic of the model. To gain more insight into this problem, the four models used in this chapter were compared for the time needed to perform a complete sensitivity analysis. The comparison was made between the complex step derivative approximation method and the (classical) central difference technique. All simulations were performed

on an Intel PIII 1GHz system running Linux (kernel 2.4.22). Again, the integration algorithm used was RK4ASC with an accuracy of $1E-06$.

Table 3.6 lists the ratios between the execution times of a single sensitivity analysis for the complex model (with complex arithmetic) and the normal model (without complex arithmetic). For the simple Monod model, no difference in execution speed was detected since the model simulations were too fast to produce a noticeable effect. For the slightly more complex AM2 model, an effect was already observed. For even more complex models (SBR and Benchmark) large differences in calculation time exist. On average, the execution of the sensitivity analysis of the complex model is 10 times slower than the model without complex arithmetic which is much more than the ratios between 1 and 3 reported in literature (Martins et al., 2000; Vatsa, 2000; Burg and Newman III, 2003). These reported ratios, however, were all related to implementations of the complex-step derivative approximation method on FORTRAN codes as opposed to C++ in this study. Therefore, inefficient C++ complex arithmetics might explain part of the speed difference, but more research to pinpoint the exact cause is certainly needed.

Table 3.6: Ratios between the execution times for a sensitivity analysis for the complex model (with complex arithmetic) and the normal model (without complex arithmetic).

Model	Ratio
Monod	1
Anaerobic	2
SBR	9
Benchmark	11

It has to be mentioned that in a real finite difference application typically much more simulations are required than the $2p$ used here because of the search for the optimal perturbation factors. This can be very cumbersome and time consuming (especially if many parameters are involved), and requires the execution of many finite difference sensitivity analyses. Even then, it is not guaranteed that an adequate perturbation factor can always be found. This requirement of additional simulations for the finite difference technique makes the complex-step derivative approximation method certainly attractive even with the associated simulation time increase for large models.

3.5 Conclusions

In this chapter the finite difference sensitivity analysis technique was discussed in detail. Practically this technique was implemented using the central difference method. This method produces sensitivity functions around the nominal parameter values θ and at the same time it allows to assess the

quality of the sensitivity function calculations. Different criteria were used as a measure for this quality.

The perturbation factor used in the finite difference method was found to be parameter dependent and to a lesser extent variable dependent. Among the proposed criteria, the SRE criterion was found useful to assess the quality of sensitivity function calculations. A threshold value of $2.35E-02$ for this criterion can be used to distinguish with high probability between "good" and "bad" sensitivity functions. It was also found that the integrator accuracy has a large influence on the value of the optimal perturbation factors and on the quality of the calculated sensitivity functions. Using a fixed perturbation factor without considering the integrator settings is thus not advisable.

Beside the finite difference sensitivity analysis technique another recently developed technique was investigated: the complex-step derivative approximation method. The main advantage of this technique is that it is not susceptible to errors introduced by small perturbation factors, ruling out the entire search for optimal perturbation factors. However, the main disadvantage is a drastic execution time increase for large models.

Chapter 4

Use of the Hessian and the FIM in parameter estimation

4.1 Introduction

As was mentioned in Chapter 1, the cornerstone for assessing parameter estimation quality is the parameter estimation error covariance matrix. This matrix can be calculated on the basis of the second derivatives of the parameter estimation objective function to the parameters, also called the Hessian matrix. In this chapter, several numerical techniques to calculate the Hessian matrix will be compared and discussed.

It was also shown in Chapter 1 that the Fisher Information Matrix (FIM) is related to the parameter estimation error covariance matrix. This also implies a relationship between the Hessian and the FIM. A second part of this chapter will therefore be devoted to the comparison of both matrices. The Hessian matrix is calculated based on sensitivity values of the parameter estimation objective function whereas the FIM is calculated based on sensitivity functions of the model outputs. It will also be evaluated whether the knowledge of this relationship would be useful in a parameter estimation exercise.

One of the goals of optimal experimental design is to design experiments that would result in data with a maximal information content. However, many experiments performed today do not result in such data. This has important consequences for parameter estimation. Low quality data might not allow all model parameters to be estimated. Therefore, a subset of parameters has to be selected that can be estimated reliably from the available data, often called an identifiable parameter set. Since the FIM properties are directly related to the parameter estimation confidence information, its

properties should be ideal candidates to be used in this selection procedure. This will be the focus of the final part of this chapter.

4.2 Numerical Hessian matrix calculation techniques

4.2.1 Introduction

The Hessian matrix is composed of the second derivatives of the parameter estimation objective function to the parameters and its elements are given by:

$$H_{ij} = \frac{\partial^2 J}{\partial \theta_i \partial \theta_j} \quad (4.1)$$

where J can be defined as the weighted least squares objective function:

$$J = \sum_{k=1}^N (y_k - y_k(\theta))^T \times Q_k^{-1} \times (y_k - y_k(\theta)) \quad (4.2)$$

This objective function is the sum over all data points (N) of the squared difference between the model prediction $y_k(\theta)$ and the measurement y_k weighted by the inverse of the measurement error covariance matrix (Q_k). Calculating the second derivatives is only analytically feasible if the objective function can be written explicitly. If this is not the case, numerical techniques have to be used.

In literature, only few methods are available which describe techniques to numerically calculate an exact Hessian matrix. One of these techniques, introduced by Spendley et al. (1962) and further developed by Nelder and Mead (1965), is based on the quadratic approximation of the objective surface around the minimum. This technique was implemented in EAST and will be discussed in detail in Section 4.2.2. Another technique was proposed by Marsili-Libelli et al. (2003) and uses a series of central differences to calculate the Hessian elements. A search algorithm was also presented in order to detect the optimal perturbation factor for these central differences requiring, however, a very large number of objective function evaluations.

Before continuing the discussion on numerical techniques, it has to be mentioned that the Hessian matrix is often used in non-linear parameter estimation. Many optimization algorithms, based on Newton's method, require the evaluation of the Hessian in their search for the minimum objective function (Fletcher, 1987). Since the calculation of the second derivatives is rather complex, almost all algorithms use some kind of approximation of the Hessian matrix. These algorithms are often referred to as quasi-Newton and build up, iteratively, a good approximation of the Hessian matrix which should converge to the true Hessian close to the minimum. However, since

only an approximation is used, it cannot be guaranteed that the final Hessian will be the true Hessian. Van Vooren (2000) investigated whether the internally calculated Hessian of the Praxis optimization algorithm (Brent, 1973) could be used as a basis for the calculation of the parameter estimation covariance matrix but found that the approximation yielded unsatisfactory result. Thus, using these Hessian matrices in parameter estimation covariance calculations should be done with some caution. Probably the most used approximation of the Hessian matrix is the Fisher Information Matrix. Section 4.3 will go more in detail about the relation between these two matrices.

4.2.2 The quadratic approximation technique

Spendley et al. (1962) proposed a method to find the minimum of an objective function using simplexes, and included a technique to calculate the final minimum using a quadratic approximation of that objective function. Their method was further developed by Nelder and Mead (1965) who also proposed to use this quadratic approximation to estimate the Hessian matrix of second derivatives at the minimum.

For a parameter estimation problem with p parameters, a simplex consists of $p + 1$ points (Y_0, Y_1, \dots, Y_p) . Figure 4.1 illustrates this for two parameters θ_1 and θ_2 . With each point, a value of the objective function J is associated. Next, "half-way points" $(Y_{ij} = (Y_i + Y_j)/2)$ are formed (also shown in Figure 4.1). If the starting point of the Hessian matrix calculations is not a simplex but just the optimum (Y_0) then the other points can easily be constructed by perturbing each parameter with a certain perturbation factor (Figure 4.2).

Once the objective value at each point is evaluated a quadratic surface can be fitted to these $(p + 1)(p + 2)/2$ points. In order to simplify the calculations involved in this fit, the coordinates of the points are rescaled (Equation 4.3).

$$\begin{aligned}
 &(0, 0, 0, \dots, 0) \\
 &(1, 0, 0, \dots, 0) \\
 &(0, 1, 0, \dots, 0) \\
 &\dots \\
 &(0, 0, 0, \dots, 1)
 \end{aligned}
 \tag{4.3}$$

The half-way points are rescaled as in Equation 4.4.

$$\begin{aligned}
 &(\frac{1}{2}, 0, 0, \dots, 0) \\
 &(\frac{1}{2}, \frac{1}{2}, 0, \dots, 0) \\
 &\dots
 \end{aligned}
 \tag{4.4}$$

A quadratic approximation of the objective function J using the points and half-way points can be written as:

$$y = A_0 + 2A^T x + x^T Bx
 \tag{4.5}$$

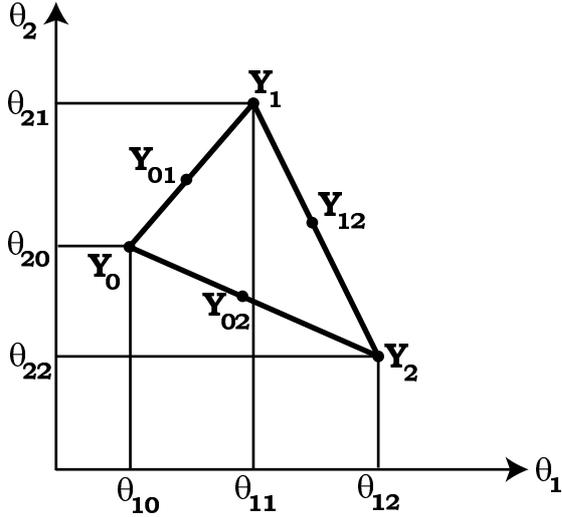


Figure 4.1: Layout of points (Y_0, Y_1 and Y_2) and half-way points (Y_{01}, Y_{02} and Y_{12}) starting from a simplex for a two parameter estimation problem.

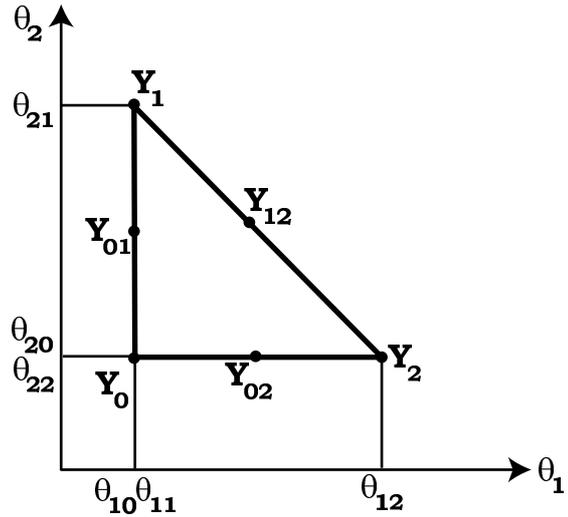


Figure 4.2: Layout of points (Y_0, Y_1 and Y_2) and half-way points (Y_{01}, Y_{02} and Y_{12}) starting from the minimum (Y_0) for a two parameter estimation problem.

where y is the quadratic approximation of the objective value J corresponding to the point with coordinates x . The constant A_0 and the elements of vector A and matrix B are given by:

$$\begin{aligned}
 A_0 &= y_0 \\
 A_i &= 2y_{0i} - (y_i + 3y_0)/2 & i = 1, \dots, p \\
 B_{ii} &= 2(y_i + y_0 - 2y_{0i}) & i = 1, \dots, p \\
 B_{ij} &= 2(y_{ij} + y_0 - y_{0i} - y_{0j}) & i \neq j
 \end{aligned}
 \tag{4.6}$$

where y_i are the values of the objective function at the points Y_i and y_{ij} are the objective function values at the half-way points Y_{ij} . The Hessian matrix in the rescaled coordinate system corresponds with B . The Hessian matrix H in the original coordinate system is given by Equation 4.7.

$$H = 2(D^{-1})^T B D^{-1}
 \tag{4.7}$$

where D is the direction matrix whose i th column is the distance between Y_0 and Y_i (i.e. $Y_i - Y_0$).

The final quality of the Hessian matrix depends on the distance between point Y_0 and points $Y_1 \dots Y_p$. When this distance is too large, the quadratic approximation of the objective function surface J will be poor and when it is too small, numerical error will influence the calculations of y .

In order to use this method in EAST, the user is required to specify a perturbation factor that will be used to construct the points $Y_1 \dots Y_p$ based on the optimum Y_0 . This can be a global perturbation factor (valid for all parameters) or it can be specified separately for each parameter.

4.2.3 Richardson's extrapolation

Calculation of the second derivative of any function can be regarded as the derivative of the derivative of the function of interest. If this would be performed using finite differences, the errors introduced by selecting non-optimal perturbation factors would certainly result in wrong results, especially in the case where a finite difference of a finite difference approximation would be considered. However, if a technique could be found that can automatically select an appropriate perturbation factor, one would be able to perform two finite difference approximations in a row to obtain the second derivative. Fortunately, such a technique exists in the field of polynomial approximation.

An interpolating polynomial of degree $N - 1$ through N points $y_1 = f(x_1)$, $y_2 = f(x_2)$, \dots , $y_N = f(x_N)$ is given explicitly by Lagrange's classical formula (Equation 4.8) (Press et al., 1992).

$$y(x) = \frac{(x - x_2)(x - x_3) \dots (x - x_N)}{(x_1 - x_2)(x_1 - x_3) \dots (x_1 - x_N)} y_1 + \frac{(x - x_1)(x - x_3) \dots (x - x_N)}{(x_2 - x_1)(x_2 - x_3) \dots (x_2 - x_N)} y_2 + \dots + \frac{(x - x_1)(x - x_2) \dots (x - x_{N-1})}{(x_N - x_1)(x_N - x_2) \dots (x_N - x_{N-1})} y_N \quad (4.8)$$

This function represents the polynomial passing through each of the N points and can be used to calculate the function value y at any value of x . Implementing this formula directly would be correct but it is awkward to program and does not provide an error estimate. Therefore a better algorithm is Neville's algorithm (Press et al., 1992). This algorithm is a recursive way to calculate the value of $y(x)$ for the polynomial of degree $N - 1$ based on lower order polynomials passing through subsets of points. Let Y_1 be the value at x of the unique polynomial of degree zero (i.e. a constant) passing through the point (x_1, y_1) ; so $Y_1 = y_1$. Likewise, define Y_2, Y_3, \dots, Y_N . Now let Y_{12} be the value at x of the unique polynomial of degree one passing through both (x_1, y_1) and (x_2, y_2) . Again, also define $Y_{23}, Y_{34}, \dots, Y_{(N-1)N}$. Similarly, continue until $Y_{12\dots N}$ is reached, which can be used to calculate the value at x of the unique polynomial of degree $N - 1$ passing through all N points. This schema is illustrated in Figure 4.3 for a polynomial passing through 4 points. The actual polynomials passing through the different subsets of points are illustrated in Figure 4.4.

Neville's algorithm now allows to calculate the values in the table one column (m) at a time using following recurrence equation:

$$Y_{i(i+1)\dots(i+m)} = \frac{(x - x_{i+m})Y_{i(i+1)\dots(i+m-1)} + (x_i - x)Y_{(i+1)(i+2)\dots(i+m)}}{x_i - x_{i+m}} \quad (4.9)$$

Beside this, an error estimate on the true value of $y(x)$ can also be obtained by keeping track of the small differences between the values of the polynomials of lower order in the schema (Press et al., 1992).

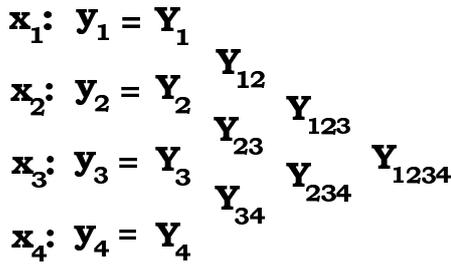


Figure 4.3: Neville’s schema to construct the interpolating polynomial passing through 4 points (Press et al., 1992).

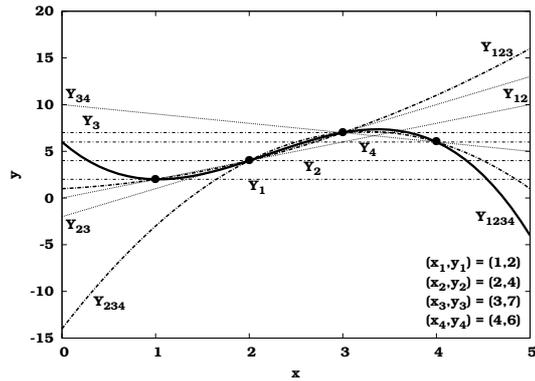


Figure 4.4: Interpolating polynomials passing through different subsets of 4 points.

The algorithm described in the previous paragraph can be applied to derivative calculations by performing finite difference calculations with smaller and smaller values of the perturbation factor ξ . The influence of the perturbation factor on the calculation of the first derivative of the parameter estimation objective function is illustrated in Figure 4.5 for a case study discussed later in this chapter.

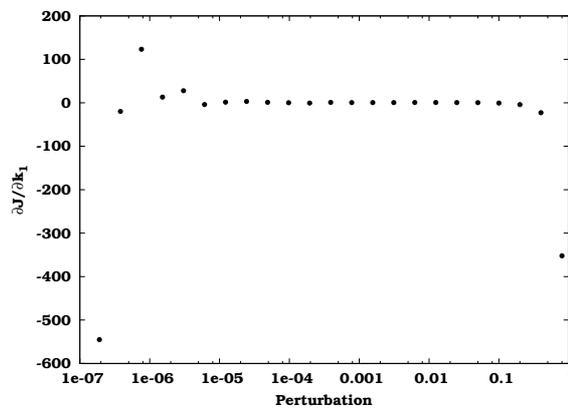


Figure 4.5: Central difference calculation of the first derivative of the parameter estimation objective function for different perturbation factors.

The conclusions that were made in Chapter 3 for the calculation of sensitivity functions are also valid here. For large perturbation factors (larger than 0.1), the non-linearity effect of the parameter on the objective function can be observed. For very small perturbation factors (smaller than $1E-5$), subtractive cancellation errors occur. Therefore a good choice of the perturbation factor needs to be made. This can be achieved by using Neville’s algorithm to construct polynomials, using finite difference derivatives calculated with increasingly smaller perturbation factors, which are able to describe the derivative at any value of the perturbation factor. This general

idea is also known as “Richardson’s deferred approach to the limit”. Rewriting Equation 4.9 in terms of derivatives ($\partial J/\partial\theta$) and perturbation factors (ξ) gives:

$$\partial J/\partial\theta_{i(i+1)\dots(i+m)} = \frac{(\xi - \xi_{i+m})\partial J/\partial\theta_{i(i+1)\dots(i+m-1)} + (\xi_i - \xi)\partial J/\partial\theta_{(i+1)(i+2)\dots(i+m)}}{\xi_i - \xi_{i+m}} \quad (4.10)$$

One then needs to extrapolate this polynomials to $\xi \rightarrow 0$ to get the derivative corresponding to an infinitesimally small perturbation factor. This extrapolation can be calculated by substituting $\xi = 0$ in Equation 4.10 and assuming that the ratio between two successive perturbation factors $\Delta\xi$ is constant:

$$\left[\partial J/\partial\theta_{i(i+1)\dots(i+m)}\right]_{\xi=0} = \frac{\partial J/\partial\theta_{(i+1)(i+2)\dots(i+m)}\Delta\xi^m - \partial J/\partial\theta_{i(i+1)\dots(i+m-1)}}{\Delta\xi^m - 1} \quad (4.11)$$

This equation thus gives an estimate of the derivative for an infinitesimally small perturbation factor based on the calculated finite difference derivative values for a number of perturbation factors. For each of these extrapolations an extrapolation error can also be calculated, based on the difference between the current extrapolation and extrapolations calculated using polynomials of lower order. After each perturbation factor decrease, the extrapolation error is evaluated and the algorithm is stopped when a certain threshold minimal error is reached.

For the case presented in Figure 4.5, the algorithm is started at a perturbation factor of 0.1 (chosen arbitrarily, but large enough). Next, finite difference derivatives are calculated by successively halving the perturbation factor. For each of these perturbation factors, extrapolations and their associated extrapolation errors are calculated. Figure 4.6 shows the derivatives calculated for perturbation factors ranging from 0.1 to $1.9E-4$ and is a magnification of Figure 4.5. It was found that for a perturbation factor of $6.25E-3$ the extrapolation error reached its minimum for the polynomial Y_{45} (passing through points (x_4, y_4) and (x_5, y_5)). This polynomial is shown in Figure 4.6 (thick line) together with the other polynomials passing through these two points (dotted lines). These other polynomials were all associated with higher extrapolation errors.

The practical implementation of this method in EAST consists of calculating the elements of the Hessian matrix using the finite difference approximation (central difference) to the second derivative given by Equation 4.12.

$$\frac{\partial^2 J}{\partial\theta_i\partial\theta_j} = \frac{\frac{\partial J}{\partial\theta_i}(\theta_j + \xi\theta_j) - \frac{\partial J}{\partial\theta_i}(\theta_j - \xi\theta_j)}{2\xi\theta_j} \quad (4.12)$$

For each Hessian element, the Richardson’s extrapolation technique is used to determine the optimal perturbation factor for the finite difference approximation of the first derivatives. Each of these first derivatives is also calculated from a finite difference approximation given by Equation 4.13.

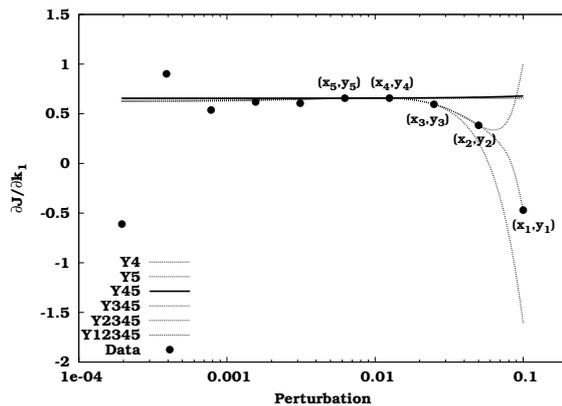


Figure 4.6: Magnification of Figure 4.5 showing the polynomial Y_{45} passing through points (x_4, y_4) and (x_5, y_5) (thick line) for which a minimal extrapolation error was found. Other polynomials passing through these points are also shown (dotted lines).

$$\frac{\partial J}{\partial \theta_i} = \frac{J(\theta_i + \xi\theta_i) - J(\theta_i - \xi\theta_i)}{2\xi\theta_i} \quad (4.13)$$

One might be tempted to calculate these first derivatives using a normal finite difference approximation (with fixed perturbation factor) but the error that would result from this would be magnified in the second derivative calculations. Therefore, Richardson's extrapolation technique is used here again to determine the optimal perturbation factor for the finite difference approximation of the parameter estimation objective function J .

4.2.4 Combining the Richardson's extrapolation method and the complex-step derivative approximation technique

The main drawback of applying the Richardson's extrapolation technique twice is that the number of objective function calculations increases very rapidly when a large number of parameters is involved (1269 simulations for the 4 parameter estimation problem presented in the next section). In order to limit the number of objective function evaluations, the first derivative calculations can be replaced by the complex-step derivative approximation technique. This technique has already been described in detail in Section 3.4 and only the basic idea will be repeated here.

In order to apply this technique, the variables and parameters of the model and the objective function calculation code need to be transformed into complex variables and all arithmetic replaced by its complex alternative. Next, the parameter estimation objective function J is calculated while applying a small step $\xi\theta$ to the imaginary part of the parameters θ (which indirectly influences J through the model simulations). The first deriva-

tive of the parameter estimation objective function can then be obtained by dividing the imaginary part of J by the parameter change:

$$\frac{\partial J}{\partial \theta} = \lim_{\xi \theta \rightarrow 0} \frac{\text{Im}[J(\theta + i\xi\theta)]}{\xi\theta} \quad (4.14)$$

This equation does not involve a difference operation and is thus not subjected to round-off error or subtractive cancellation, even if extremely small step sizes are used. Typically a step size of $1E-20$ or lower is used. Equation 4.14 can then be used to replace Equation 4.13 and requires only one objective function evaluation. This contrasts strongly with the typical 10 to 20 objective function evaluations needed if the Richardson's extrapolation technique would be used. However, it has been shown in Section 3.4 that the transformation of the model calculations into the complex form resulted in an increased simulation time for some (complicated) models. This might counteract the gain in calculation time compared to applying the Richardson's extrapolation technique twice.

4.2.5 Comparison of the numerical Hessian calculation techniques

In this section, model AM2 (Section 2.2.2) will be used to illustrate and compare the three numerical techniques to calculate the Hessian matrix described in Sections 4.2.2, 4.2.3 and 4.2.4. Using the default parameter values and input specified in Bernard et al. (2001), the volatile fatty acids (VFA) concentration S_2 ($mmol.l^{-1}$) was simulated for 30 days using an output interval of 0.1 day. A realistic error with standard deviation $1 mmol.l^{-1}$ was used to add normally distributed noise to the simulation results which then could be used as virtual measurements for the parameter estimation objective calculation. The Hessian matrix was calculated for 4 parameters: the yield coefficient for VFA production, k_2 ($mmol.g^{-1}$); the maximum methanogenic bacteria growth rate, $\mu_{2,max}$ (d^{-1}); the VFA half saturation constant, K_{S_2} ($mmol.l^{-1}$); and the fraction of bacteria in the liquid phase α (-). The parameter values for which the Hessian is calculated are taken to be the default model values: 116.5, 0.74, 9.28 and 0.5 respectively.

The results of the Hessian calculations using the three techniques are listed in Table 4.1, together with the number of required objective function evaluations for each technique. Among the three techniques, both the Richardson's extrapolation and the Richardson's extrapolation combined with the complex-step derivative approximation method yield identical results. The quadratic approximation technique results in slightly different Hessian elements, but they are still very close to those of the other techniques. This clearly illustrates that the three techniques are capable of producing a correct Hessian matrix.

When looking at the number of objective function evaluations that were required, it is clear that the Richardson's extrapolation technique needed

Table 4.1: Comparison of the Hessian matrices calculated using the quadratic approximation technique (QUAD), the Richardson's extrapolation technique (RICH) and the combined Richardson's extrapolation and complex-step derivative approximation technique (RICH+COMP). For each calculation method the number of objective function evaluations is also indicated.

Method						# evaluations
QUAD	k_2	$\mu_{2,max}$	K_{S2}	α		15
	k_2	$6.82345E-04$				
	$\mu_{2,max}$	$6.32544E+00$	$6.62539E+05$			
	K_{S2}	$-7.80334E-02$	$-1.44883E+04$	$3.44691E+02$		
	α	$-1.10545E+01$	$-9.79492E+05$	$2.13604E+04$	$1.45240E+06$	
RICH	k_2	$\mu_{2,max}$	K_{S2}	α		1269
	k_2	$6.82410E-04$				
	$\mu_{2,max}$	$6.32521E+00$	$6.62569E+05$			
	K_{S2}	$-7.80634E-02$	$-1.44885E+04$	$3.44713E+02$		
	α	$-1.10544E+01$	$-9.79495E+05$	$2.13601E+04$	$1.45237E+06$	
RICH+COMP	k_2	$\mu_{2,max}$	K_{S2}	α		105
	k_2	$6.82410E-04$				
	$\mu_{2,max}$	$6.32521E+00$	$6.62569E+05$			
	K_{S2}	$-7.80634E-02$	$-1.44885E+04$	$3.44713E+02$		
	α	$-1.10544E+01$	$-9.79495E+05$	$2.13601E+04$	$1.45237E+06$	

most objective function evaluations. It can also be seen that replacing the calculation of the first derivative in the Richardson's extrapolation technique by the complex-step derivative approximation method resulted in an approximately 10 times lower number of required objective function calculations. The fastest technique is the quadratic approximation method which only required 15 objective function evaluations. However, this number only reflects the amount of evaluations needed to calculate the Hessian with given perturbation factors. Since this technique is not automatic, it is still up to the user to find the appropriate perturbation factors. This search obviously will require a large number of executions of the quadratic approximation method. Another disadvantage of this technique is that no error estimate of the result is available which means that the user is unable to verify whether the calculated Hessian is correct, unless the results can be compared with other techniques, as is done here.

The Hessian matrix calculated using the quadratic approximation technique listed in Table 4.1 was obtained using the following specific perturbation factors for the parameters: $1E-03$ for k_2 , $1E-05$ for $\mu_{2,max}$, $1E-04$ for K_{S2} and $1E-05$ for α . Only by using these specific perturbation factors could the Hessian matrix approach the matrices calculated by the other techniques. Table 4.2 shows the Hessian matrix calculated using a fixed perturbation

Table 4.2: Hessian matrix calculated using the quadratic approximation technique (QUAD) with a fixed perturbation factor of $1E-05$. The ratios (RATIO) between the elements of this matrix and the correct Hessian matrix elements (Table 4.1) are also shown.

Method				
QUAD	k_2	$\mu_{2,max}$	K_{S_2}	α
k_2	$-7.01944E-03$			
$\mu_{2,max}$	$5.09553E+00$	$6.62539E+05$		
K_{S_2}	$-8.03396-02$	$-1.45047E+04$	$3.44607E+02$	
α	$-1.24753E+01$	$-9.79492E+05$	$2.13349E+04$	$1.45240E+06$
RATIO	k_2	$\mu_{2,max}$	K_{S_2}	α
k_2	-10.28625			
$\mu_{2,max}$	0.80559	0.99995		
K_{S_2}	1.02916	1.00112	0.99969	
α	1.12854	1.00000	0.99882	1.00003

factor of $1E-05$ together with the ratios between the elements of this Hessian matrix and the elements of the Hessian matrix calculated using the Richardson's extrapolation method (Table 4.1). As can be seen from this matrix, the ratios of the Hessian elements corresponding to parameters $\mu_{2,max}$ and α are almost 1. On the other hand, the ratios of the Hessian elements corresponding to parameter k_2 deviate significantly from unity. For K_{S_2} this is less pronounced. These results suggest that the perturbation factor significantly influences the Hessian matrix calculations, contradicting the results of Van Vooren (2000) who found little influence of the perturbation factors on the Hessian calculation for his specific application. These findings suggest that the influence of the perturbation factors is parameter specific and thus also case specific.

In order to explain the different required perturbation factors, it is interesting to investigate the sensitivities of the measured variables to the parameters. Based on the sensitivity functions of S_2 to the four selected parameters it could be concluded that S_2 is the least sensitive to k_2 , more sensitive to K_{S_2} and the most sensitive to $\mu_{2,max}$ and α . This correlates with the magnitude of the different perturbation factors. If the variable is very sensitive to a certain parameter, this parameter only has to be perturbed slightly (low perturbation factor) in order to produce a significant change in the objective function value. On the other hand, if the variable is not very sensitive to a parameter, this parameter needs to be highly perturbed (high perturbation factor) in order to produce a significant change in the objective value. These results suggest that a global perturbation factor is not a good choice, unless the sensitivity of the measured variables to the parameters is more or less equal.

4.3 Relationship between the Hessian and the FIM

4.3.1 Theoretical background

One of the most used approximations of the Hessian matrix is the Fisher Information Matrix. A relationship between them can be established starting from the elements of the Hessian matrix, being the second derivatives of the parameter estimation objective function to the model parameters (Equation 4.2) (Press et al., 1992; Dochain and Vanrolleghem, 2001; Marsili-Libelli et al., 2003):

$$\begin{aligned}
 H_{ij} &= \frac{\partial^2 J}{\partial \theta_i \partial \theta_j} \\
 &= 2 \sum_{k=1}^N \left[\left(\frac{\partial y(x_k, \theta)}{\partial \theta_i} \right)^T Q_k^{-1} \left(\frac{\partial y(x_k, \theta)}{\partial \theta_j} \right) - (y_k - y(x_k, \theta))^T Q_k^{-1} \left(\frac{\partial^2 y(x_k, \theta)}{\partial \theta_i \partial \theta_j} \right) \right] \\
 &= 2 \sum_{k=1}^N \left[\left(\frac{\partial y(x_k, \theta)}{\partial \theta_i} \right)^T Q_k^{-1} \left(\frac{\partial y(x_k, \theta)}{\partial \theta_j} \right) \right] \\
 &\quad - 2 \sum_{k=1}^N \left[(y_k - y(x_k, \theta))^T Q_k^{-1} \left(\frac{\partial^2 y(x_k, \theta)}{\partial \theta_i \partial \theta_j} \right) \right] \\
 &= 2 \times FIM_{ij} - 2 \sum_{k=1}^N \left[(y_k - y(x_k, \theta))^T Q_k^{-1} \left(\frac{\partial^2 y(x_k, \theta)}{\partial \theta_i \partial \theta_j} \right) \right] \tag{4.15}
 \end{aligned}$$

From this equation it can be seen that the Hessian and two times the FIM differ by the term:

$$2 \sum_{k=1}^N \left[(y_k - y(x_k, \theta))^T Q_k^{-1} \left(\frac{\partial^2 y(x_k, \theta)}{\partial \theta_i \partial \theta_j} \right) \right] \tag{4.16}$$

This term contains three components: (1) the estimation error ($y_k - y(x_k, \theta)$), (2) a term of second derivatives of y to the parameters, also known as “curvature” and (3) the inverse measurement error covariance matrix. The estimation error can be seen as an amplifier for the curvature (which is 0 for linear models only), and for a successful model, this term should be the random (normally distributed with mean 0 and variance σ^2) measurement error of each point. This error can have either sign and, therefore, the second derivative terms tend to cancel out when summed over all points k . However, this is only true when the model fits the experimental data perfectly, i.e. the model structure is correct and the global minimum is found. If this is not the case, Equation 4.16 will cause H to be significantly different from $2 \times FIM$.

Marsili-Libelli et al. (2003) proposed to use the conceptual difference between the Hessian matrix and the FIM to detect inaccurate parameter estimation results due to the optimization algorithm terminating far from the

optimum. This method was illustrated successfully for two rather simple ecological models using “virtually” generated data. However, in a real parameter estimation problem, other factors beside the premature convergence of the optimization algorithm might also influence the difference between the Hessian matrix and the FIM. One obvious factor is the fact that the model is inadequate. Closely related to this is the fact that in a typical parameter estimation problem not all model parameters are estimated. Fixing some parameters at specified values might result in an unsatisfactory model fit (although being the global optimum for the selected parameters), again, causing Equation 4.16 to contribute significantly to the difference between the Hessian matrix and the FIM. As a consequence of these issues, the difference between the Hessian matrix and the FIM can conveniently be used as an indicator for model inadequacies, local minima and/or non-normally distributed residuals.

4.3.2 Numerical illustration

In order to study the differences between the Hessian matrix and the FIM, model AM2 was again used to generate virtual data. Using the default parameter values and input specified in Bernard et al. (2001), 6 variables were simulated for 30 days using an output interval of 0.1 day. Realistic error characteristics were used to add normally distributed noise to the simulation results. The variables used are listed in Table 4.3 together with their standard deviation. Since a weighted least squares objective function is used, the contribution of each variable to the total objective function is approximately 301, equal to the number of data points. This results in an approximate minimum objective function of 1806.

Table 4.3: AM2 model variables that are used to generate virtual data. Based on the listed standard deviations, normally distributed noise was added to the model simulations.

Name	Unit	Description	Std. deviation σ
S_1	$g.l^{-1}$	organic substrate	0.2
S_2	$mmol.l^{-1}$	volatile fatty acids	1
pH	–	pH	0.02
C_{tic}	$mmol.l^{-1}$	total inorganic carbon	3
Q_C	$l.d^{-1}$	carbon dioxide flow rate	50
Q_M	$l.d^{-1}$	methane flow rate	50

Using these virtual data, all model parameters and initial values were estimated (19 in total), starting from the default values. The minimum objective value reached, was 1803.66 (lower than the theoretical value of 1806 due to this particular noise sequence). Using the optimal parameter values found, both the Hessian matrix and the FIM were calculated. The calculation of the 19×19 Hessian matrix required 2061 objective function evaluations and was

performed using the method which combines the Richardson's extrapolation method and the complex-step derivative approximation method (Section 4.2.4). The sensitivity functions of the FIM were calculated using the complex-step approximation method and required 19 model simulations. Next, the ratios between the elements of the Hessian matrix and the FIM were calculated. The average of all ratio's between the elements was calculated to be 1.9997 (standard deviation 0.0349) which is very close to the theoretical ratio of 2. This illustrates that indeed, $H = 2 \times FIM$ at the minimum and also gives confirmation of the correctness of the calculation of the Hessian matrix and the FIM.

The fact that H only equals $2 \times FIM$ when a good model fit is achieved will now be visually illustrated using 95% confidence ellipses for 2 of the 19 estimated parameters: k_1 and K_{S_1} . The ellipses are constructed for two different parameter sets: set 1 representing a situation far from the optimum ($k_1 = 60$ and $K_{S_1} = 5$) and set 2 at the optimum found after calibration of the 19 parameters and initial conditions based on the virtual data ($k_1 = 42.77$ and $K_{S_1} = 7.03$). At each of these points in parameter space, H and $2 \times FIM$ are calculated and used to construct the confidence ellipses. The confidence ellipses for parameter set 1 are shown in Figure 4.7. This figure clearly shows that the confidence ellipses do not coincide. Indeed, the confidence ellipse calculated based on $2 \times FIM$ (dotted line) extends beyond the borders of the figure. On the other hand, Figure 4.8 shows that the confidence ellipses at the optimum do coincide due to the fact that at this point in parameter space indeed $H = 2 \times FIM$.

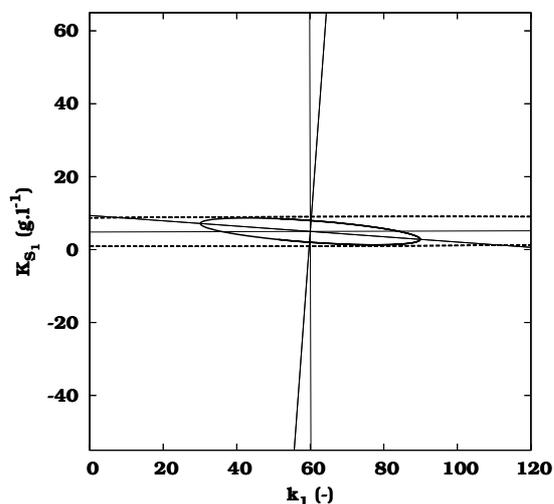


Figure 4.7: 95% confidence ellipses and axes for parameters k_1 and K_{S_1} constructed using H (full line) and $2 \times FIM$ (dashed line) at a point in parameter space far from the optimum.

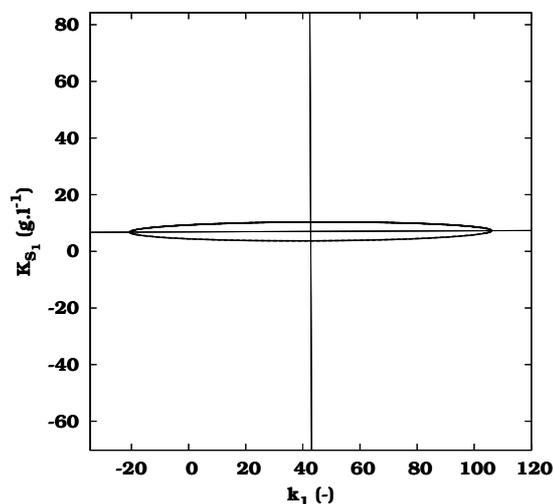


Figure 4.8: 95% confidence ellipses and axes for parameters k_1 and K_{S_1} constructed using H (full line) and $2 \times FIM$ (dashed line) at the optimum found after calibration.

4.4 Selecting identifiable parameters

4.4.1 Overview of existing methods

4.4.1.1 Identifiability

The question of identifiability is a very fundamental one in (bioprocess) modelling. Loosely speaking, the problem is whether an identification procedure can yield unique values of the model parameters given certain data. In the field of bioprocess engineering, the complexity of the mathematical models and the scarcity of good experimental data makes the identification of model parameters a rather difficult task. A lack of identifiability can have several causes: the problem could be related to the model structure itself (structural identifiability problems) or the quantity and/or quality of the experimental data (practical identifiability problems).

Structural identifiability, as a first step, addresses whether it is possible to find unique parameter values assuming the measured input/output variables are available in the absence of noise. Structural identifiability can be performed in the absence of any prior information on the value of the parameters and even before collecting any data on the system to be studied. Many techniques exist to assess structural identifiability such as the Laplace transform method, the similarity transform method, the Taylor series method and the local state isomorphism method (Walter and Pronzato, 1997; Dochain and Vanrolleghem, 2001). These methods are only feasible for relatively simple non-linear models. For Monod-based activated sludge models, Petersen et al. (2003) showed that structurally identifiable parameters could be directly derived from the Petersen matrix, thus reducing the time-consuming task of finding the structurally identifiable parameters using classical techniques.

Practical identifiability is not only related to the model structure but also to the quality and the availability of the data. In other words, is the available data informative enough to identify the model parameters and give them a unique and accurate value? Most methods for the evaluation of practical identifiability are based on a sensitivity analysis of the model parameters in combination with information on the measurement uncertainty (Vanrolleghem et al., 1995; Versyck et al., 1997; Brun et al., 2001; Petersen et al., 2001).

Practical identifiability is an important complement to structural identifiability. In the case of noise corrupted data the structural identifiability of certain parameters might no longer be sufficient to allow practical identification because a change in one parameter can be compensated almost completely by a proportional shift in another, while still producing a satisfying fit between the model predictions and the data. Structural identifiability can be regarded as the upper bound for model parameter identifiability which will probably not be reached in a practical situation. Therefore, it is valid to investigate the practical identifiability of the model parameters even if the

structural identifiability was not studied, since structurally unidentifiable parameters will also be practically unidentifiable.

4.4.1.2 Selecting practically identifiable parameter subsets

Knowing that most bioprocess models are complex and highly non-linear it can be expected that not all model parameters are identifiable. The question that then arises is: which parameter subset can be identified given a measured data set? Rather than trying to identify the true parameters, the task could also be to find physically reasonable parameter values that describe the data adequately. Several approaches to answer this question have been developed in the past and will now be discussed.

Model reduction

An obvious strategy to overcome identifiability issues is to reduce or simplify the model structure in such a way that the model only describes that what is observed and only identifiable parameters remain. However, simplifying the model structure might not be an easy task and might eventually result in a pure input-output model without any (desired) mechanistic properties. In addition to that, Omlin and Reichert (1999) illustrated the danger of performing extrapolations or predictions with simplified identifiable models.

Monte Carlo techniques

A general technique that does not require the parameters to be uniquely identifiable is the use of prior information on the parameters and uncertainty on the measurement data to obtain sets of parameters which yield almost the same level of agreement between measurements and model predictions. This idea forms the basis of many identification methods including Bayesian identification (Omlin and Reichert, 1999), Regionalized Sensitivity Analysis (RSA) (Spear and Hornberger, 1980) also known as the Hornberger-Spear-Young-algorithm (HSY) (Beck, 1987), Generalised Likelihood Uncertainty Estimation (GLUE) (Beven and Binley, 1992) and the Monte Carlo Set-Membership algorithm (Keesman, 1990). Although these techniques were not developed to identify a single set of parameter values, they can be adjusted for this purpose. The parameter set resulting in the closest agreement between the measurements and the model predictions can be used as the “best” parameter set (Vanrolleghem and Keesman, 1996; von Sperling, 1993). The major drawback of these methods is the use of Monte Carlo sampling techniques which are computationally demanding when a large number of parameters is involved. Beside this, prior information on the uncertainty of the model parameters might also be difficult to obtain (Omlin et al., 2001).

Regression methods

The method of Abusam et al. (2001) to select the best identifiable parameters is based on the response surface method (RSM) and was applied to the calibration of ASM1 on a full-scale oxidation ditch. This method requires the simulation of the model at specific points in parameter space around the nominal parameter vector (based on a two-level factorial design). Based on the simulated values of the measured variables for the different parameter sets, a second order regression model can be fitted. From the coefficients of this regression model, it then can be concluded which parameters are identifiable. With this method, the number of required simulations increases exponentially with the number of parameters. Therefore it was suggested by Abusam et al. (2001) to restrict the number of parameters to no more than 10 to 12 and to use a sensitivity analysis to perform a pre-selection of the most sensitive model parameters.

Beside a second order regression model, this technique can also be generalized by performing a Monte Carlo sampling of the parameter domain and calculating the model output for each of these parameter sets. Based on the simulated values (of the measured variables) at these points in parameter space, a general regression model can be fitted. It can then be verified using a statistical test whether the coefficients of this regression model are significantly different from 0 which gives an indication of the parameter identifiability (Saltelli et al., 2000).

Visual interpretation of local sensitivity functions

Probably the most used class of techniques to find practically identifiable parameter subsets based on a given data set is by using local sensitivity functions. These sensitivity functions express how sensitive certain variables (which are measured) are to the model parameters and thus are a useful tool to determine the most influential parameters. Important parameter correlations can also be detected to a certain extent by studying the linear dependency between the sensitivity functions. Parameters which are correlated will show a similar sensitivity function shape. However, visual interpretation of these sensitivity functions is only possible for relatively simple models (Reichert et al., 1995; Marsili-Libelli et al., 2001; Seagren et al., 2003; Steppe, 2004).

Method based on scalar properties of the FIM

Visual inspection of sensitivity functions becomes increasingly difficult if the number of parameters and (measured) variables increases. Therefore, the FIM, which is in essence a condensation of the sensitivity information and measurement uncertainty in one matrix, can be used to select identifiable parameter combinations.

Weijers et al. (1996) applied this idea for the calibration of a full scale wastewater treatment plant model (ASM1). Parameter subset selection was based on analysing two scalar properties of the FIM: the modified E-optimal cri-

terion (ratio between the largest and smallest FIM eigenvalue), which is a measure for the shape of the parameter estimation confidence region and the C criterion which is a measure of the mean parameter estimation error (sum of FIM eigenvalues). These two criteria were calculated for all possible parameter combinations and used to determine sets of identifiable parameter combinations.

This idea was further developed by Weijers and Vanrolleghem (1997) who proposed a general procedure for calibrating an activated sludge plant based on selecting identifiable parameter subsets. This procedure was applied to the calibration of ASM1 and involves several steps:

1. Define the plant model, input data and initial conditions. This step includes analyzing the plant and gathering all available data.
2. Define the set of outputs and a priori parameter values. During this stage, it has to be decided which outputs will be used for calibration and initial parameter values should be selected. For example, default model parameter values or values from literature can be taken.
3. Select a reduced set of most sensitive parameters. In order to avoid a combinatorial explosion and reduce the computational time of the parameter subset selection method, it is suggested to reduce the number of parameters to approximately 10 to 15. This is accomplished by calculating sensitivity coefficients for each parameter (θ_j) related to each output (y_i) using Equation 4.17.

$$S_{\theta_j}^{y_i} = \frac{\frac{1}{N} \sum_{k=1}^N \left| \frac{\partial y_i(k)}{\partial \theta_j} \right|}{\frac{1}{N} \sum_{k=1}^N y_i(k, \theta_0)} \times \theta_{j0} \quad (4.17)$$

where N is the number of measurement points, $y_i(k, \theta_0)$ is the model output using the nominal parameter values θ_0 , $\partial y_i / \partial \theta_j$ is the sensitivity of the output y_i to the parameter θ_j and θ_{j0} is the nominal value of parameter θ_j . The parameter set can then be reduced using two techniques. Either by ranking the averaged sensitivity coefficients, obtained by averaging over all n outputs ($\sum_{i=1}^n S_{\theta_j}^{y_i} / n = S_{\theta_j}$) or by selecting those parameters which show a scaled (with respect to the maximum sensitivity coefficient for that output) sensitivity coefficient larger than 0.2 in at least one of the outputs. The last technique ensures that at least one parameter will be selected for each output.

4. Select those parameter subsets, from the reduced set, which show the highest identifiability. Identifiability is assessed in this step by evaluating the FIM modified E-optimal and D-optimal criterion (determinant of the FIM) for each possible combination of parameters selected from the reduced parameter set. For each parameter subset size, the subsets were ranked according to the D-optimal criterion and among the

highest ranked subsets the subset was chosen which had a significantly lower modified E-optimal criterion value than the surrounding sets. This results in one “best” parameter subset for each subset size.

5. Estimate the selected parameters from real data. Non-linear parameter estimation is now used to find the values of the parameters for the chosen “best” parameter subset. All other parameters are kept at their initial values.

Since no critical values for the modified E-optimal and D-optimal criteria could be supplied to select the best identifiable parameter subset, it was decided to use virtually generated data based on known parameter values to estimate the values of the best parameter subsets of each subset size in order to determine the subset size at which convergence problems occurred. It was found that for subset sizes of 6 to 8 parameters substantial deviations from the true values occurred. However, this result only holds for the specific case study performed by the authors.

Weijers and Vanrolleghem (1997) also investigated the influence of the a priori assumed parameter values on the parameter subset selection procedure. Using realistic ranges for all parameters (determined from literature), Latin hypercube sampling was used to generate different sets of initial parameters. For each of these sets, steps 3 and 4 of the proposed procedure were repeated. From the analysis of these results it could be concluded that the best identifiable parameter set selection was not influenced much by the a priori assumed parameter values. Beside the influence of the a priori assumed parameter values, attention was also paid to the influence of the values of the parameters that were fixed during parameter estimation, also known as the bias problem. Errors in the assumed values of the fixed parameters will be compensated by the values of the estimated parameters, sometimes leading to unrealistic estimated values. This problem also leads to rather optimistic confidence regions for the estimated parameters because the parameters which are not estimated are fixed at their “true” values, which are generally unknown.

The FIM was also used by Ioslovich et al. (2004) as the basis of a parameter selection procedure to calibrate a lettuce growth model. In their method, correlations between parameters are assessed by analysing the off-diagonal elements of the FIM calculated from normalized relative sensitivity functions (the sensitivity vectors have unit length). Normalization is performed by dividing the relative sensitivity vectors by their Euclidean norm. Beside this, parameter importance was determined from the diagonal elements of the non-normalized FIM.

Parameter subset selection based on the collinearity index

Another technique which is centered around the analysis of local sensitivity functions was developed by Brun et al. (2001) and was illustrated with two rather simple environmental models. This technique was also applied to

other, more complex cases: the calibration of a biochemical model of Lake Zurich (Omlin et al., 2001), the calibration of River Water Quality Model No. 1 (RWQM1) (Reichert and Vanrolleghem, 2001), the calibration of an ASM2d model of a full scale wastewater treatment plant (Brun et al., 2002) and the calibration of a population balance model of an industrial leaching process (Diez et al., 2004).

In order to obtain a parameter importance ranking, non-dimensional sensitivity functions are used:

$$S_{i,j} = \frac{\Delta\theta_j}{sc_i} \times \frac{\partial y_i}{\partial \theta_j}, \quad \tilde{S}_{i,j} = \frac{S_{i,j}}{\sqrt{\sum_{k=1}^N S_{k,j}^2}} \quad (4.18)$$

where $\Delta\theta_j$ is the uncertainty range of the parameter θ_j , sc_i represents either the typical magnitude or measurement accuracy of y_i , $\partial y_i / \partial \theta_j$ is the sensitivity of y_i with respect to the parameter θ_j and N is the total number of points. A measure for the importance of a single parameter can then be formulated as:

$$\delta_j^{msqr} = \sqrt{\frac{1}{N} \sum_{i=1}^N S_{i,j}^2} \quad (4.19)$$

Parameters with high δ^{msqr} values have a significant influence on the variables y . Note that the value of δ^{msqr} can be very sensitive to the choice of $\Delta\theta_j$ and sc_i . A suitable choice for these values is therefore very important. For $\Delta\theta_j$ it is suggested by Brun et al. (2001) to use the standard deviation of the range of reasonable values for θ_j based on literature or expert knowledge. In cases where little prior information about the parameter is available, a reasonable choice can be the nominal parameter value θ_{j0} . This at least accounts for the different scales of the parameters. For sc_i , values are usually chosen to be constant for each output and should account for the different scales of the different outputs (Brun et al., 2001). Additionally, it should be noted that the sensitivity importance index is sensitive to the nominal parameter values θ_0 since it relies on local sensitivity functions.

In order to determine the linear dependencies of the sensitivity function for different parameters in a subset, a collinearity index $\gamma(\theta)$ can be calculated according to Equation 4.20.

$$\gamma(\theta) = \frac{1}{\sqrt{\lambda_{min} [\tilde{S}^T \tilde{S}]}} \quad (4.20)$$

where λ_{min} represents the smallest eigenvalue of the matrix of squared normalized sensitivity functions \tilde{S} (Equation 4.18). Values for $\gamma(\theta)$ range from unity for linearly independent sensitivity functions to infinity for increasing degrees of dependency between the sensitivity functions.

A set of identifiable parameters can then be constructed from the sensitivity importance index starting from the most sensitive parameter and adding

less sensitive parameters unless the collinearity index becomes too large (larger than a threshold of the order of 10 – 15). Parameters that lead to a too severe increase in the collinearity index are omitted and the procedure continues until either no additional parameter is found that leads to a moderate value of the collinearity index or the sensitivity importance measure becomes too small (Reichert and Vanrolleghem, 2001).

Since this procedure of selecting an identifiable parameter subset relies on the nominal parameter values θ_0 , Brun et al. (2002) suggest to use an iterative procedure to calibrate a model. Starting from an initial parameter set θ_0^{ini} , the best identifiable parameter subset is determined and used to calibrate the model. Based on this calibrated model, a new iteration of finding the best identifiable parameter subset and calibrating the model is performed. The iterative process is continued until convergence is obtained which results in the final parameter values θ_0^{end} . At the end of this procedure it is also suggested to investigate potential bias problems with parameters which were not estimated in order to account for unrealistic values of estimated parameters.

Parameter subset selection using principal component analysis (PCA)

Recently, Li et al. (2004) also proposed a parameter subset selection technique based on local sensitivity analysis and applied it to the problem of the kinetic parameter estimation of an industrial model for a polymerization reactor. The magnitude of each parameter effect on the measured variables is quantified by applying principal component analysis to a dimensionless steady-state parameter-output sensitivity matrix given by:

$$\bar{S}_{ij} = \frac{\theta_j}{\bar{y}_i} \times \frac{\partial \bar{y}_i}{\partial \theta_j} \quad (4.21)$$

where \bar{y}_i is the steady state output value corresponding with a certain operating point of the reactor and $\partial \bar{y}_i / \partial \theta_j$ is the sensitivity of \bar{y}_i to model parameter θ_j . Principal component analysis is applied to the matrix $X = \bar{S}^T \bar{S}$. The absolute value of principal component element C_{ij} reflects the contribution of the j th parameter to the variance of the i th output. The weighted sum of m (the number of outputs) nonzero eigenvalues λ_i of X and their corresponding principal component elements is used as a measure of the overall effect of the j th parameter:

$$E_j = \frac{\sum_{i=1}^m |\lambda_i C_{ij}|}{\sum_{i=1}^m |\lambda_i|} \quad (4.22)$$

where $0 \leq E_j \leq 1$. The value of E_j reflects the difficulty in determining the j th parameter if only this parameter would be estimated.

This method, however, does not account for the quantity and quality of the measured data since no measurement errors and dynamics are considered. Possible linear dependencies between the parameters are evaluated by introducing a measure d_j for the linear dependency between a particular sen-

sitivity vector \bar{S}_j (containing the sensitivity of each variable to that particular parameter) and the other sensitivity vectors. Based on a linear combination of the sensitivity vectors of the other parameters, a sensitivity vector \tilde{S} is calculated which is at a minimum-distance to \bar{S}_j . Li et al. (2004) found that the sine of the angle between \bar{S}_j and \tilde{S} gives a good indication of the dependency of the j th parameter on the other parameters.

$$d_j = \sin \left[\cos^{-1} \left(\frac{\bar{S}_j^T \tilde{S}}{\|\bar{S}_j\| \times \|\tilde{S}\|} \right) \right] \quad (4.23)$$

In this equation $\|\cdot\|$ denotes the Euclidean norm ($\sqrt{\sum (\cdot)^2}$) and $d_j \in [0, 1]$. When d_j equals 1, the sensitivity vector \bar{S}_j is orthogonal to the vector space of the other sensitivity vectors and thus not linearly dependent. On the other hand, the j th parameter is linearly dependent on the other parameters if d_j equals 0.

Based on these two measures an identifiability index can be calculated:

$$I_j = E_j \times d_j \quad (4.24)$$

which can then be used as a measure for the identifiability of a parameter. This index includes the importance of the parameter with respect to the outputs and the linear dependency of the parameter on other parameters. Although this technique was developed for steady-state conditions, the authors also suggest modifications that could account for dynamics and non-linearity effects.

4.4.2 An improved procedure for parameter subset selection based on FIM related properties

4.4.2.1 Proposed procedure

Theoretically speaking, the Hessian matrix could be used to determine the identifiable parameters since it is the basis for the parameter estimation error covariance matrix and the correlation matrix which are used to evaluate the parameter estimation quality (Section 1.2.2). However, since this involves a significant number of simulations (Section 4.2), especially when a large number of parameters are considered, it would be impractical to use the Hessian matrix as a basis for parameter subset selection. A useful alternative would be the FIM since this matrix is directly related to the Hessian matrix, under certain conditions (Section 4.3). Based on this matrix, the following procedure is suggested to select an identifiable parameter subset and to calibrate the model:

1. Assume nominal parameter values θ_0 for the model. Typically, these are default model, literature or expert knowledge based values.

2. Calculate the FIM based on the nominal parameter values. This involves the calculation of the sensitivity functions of the measured variables to all model parameters p and includes the measurement uncertainty through the use of the measurement error covariance matrix.
3. Pre-selection of the most influential parameters, in order to prevent the combinatorial explosion of the number of parameter subsets in later steps of the procedure. Parameters can be ranked based on their δ^{msqr} value (Equation 4.19). Based on this ranking, no more than the 30 most influential parameters ($p_{influential}$) should be selected.
4. Subset calculations:
 - (a) Form subsets of parameters with size s ($2 \leq s \leq p_{influential}$) based on all possible combinations between the parameters.
 - (b) For each subset, compose a sub-sampled FIM of size $s \times s$ by selecting the corresponding elements from the complete FIM (not requiring an additional sensitivity analysis).
 - (c) Calculate the D-optimal criterion value for each of the sub-sampled matrices (product of the eigenvalues).
 - (d) Calculate the correlation matrix based on each of the inverted sub-sampled FIMs and retain the highest absolute (off-diagonal) correlation value.
5. Subset selection:
 - (a) Remove all subsets which have a highest correlation value larger than a predefined value (e.g. 0.5).
 - (b) From the remaining subsets, select the subset with the highest D-optimal criterion value.
6. Calibrate the model using the selected parameter subset from (5) and obtain the optimal values for the selected parameters.
7. Investigate possible bias effects of parameters that were not estimated.
 - (a) Analyse all parameter subsets containing one additional parameter on top of the already selected parameters from (5).
 - (b) Find the subset, and thus the additional parameter, which causes the lowest increase in highest correlation value and at the same time increases the D-optimal criterion value.
 - (c) Recalibrate the model using this extra parameter and use the already found optimal values from (6) as initial values for the parameters subset selected in (5).
8. Replace the current values of the estimated parameters in the model by their optimal values.

9. Based on the new nominal parameter values, repeat (2)-(8) until convergence is achieved.

4.4.2.2 Critical evaluation of the proposed procedure and comparison with other techniques

Compared to the other methods described in Section 4.4.1, the proposed technique requires limited expert knowledge and interaction. Actually, expert knowledge is only required to specify the measurement uncertainty (condensed in the measurement error covariance matrix) and the critical acceptable correlation value between the parameters.

Assessment of the dependencies between model parameters is performed using the correlation matrix as opposed to the modified E-optimal criterion suggested by Weijers and Vanrolleghem (1997), the collinearity index used by Brun et al. (2001) and the d_j measure proposed by Li et al. (2004). This approach allows for an easier definition of the cut-off value since most modellers are already used to quantify parameter dependencies through correlations. The correlation approach is preferred to the use of the modified E-optimal criterion because the latter is known to be case and unit dependent and does not allow for a general critical value to be specified. This problem was acknowledged by Weijers et al. (1996) and Weijers and Vanrolleghem (1997) where it was suggested to use heuristic methods or parameter estimation algorithms to detect at which point problems occurred. Petersen (2000) also showed that rescaling of the model parameters had a strong effect on modified E-optimal designs. For this reason, the modified E-optimal criterion was also not used by Reichert and Vanrolleghem (2001) in their comparison between the techniques proposed by Weijers and Vanrolleghem (1997) and Brun et al. (2001). The collinearity index used by Brun et al. (2001) is therefore a more appropriate measure because it allows to specify a critical value, which was empirically found to be 10 – 15. This value was successfully used by Brun et al. (2001), Reichert and Vanrolleghem (2001) and Brun et al. (2002). However, Omlin et al. (2001) estimated a parameter subset of 7 parameters resulting in a correlation of 0.96 between two of the parameters while the collinearity index of the entire subset was between 10 and 15.

An important aspect of the proposed procedure is its iterative nature, as was also suggested by Brun et al. (2002). Since the FIM is a local measure which depends on the nominal parameter values (through the local sensitivity functions), it is required to repeat the procedure at each calibration stage until convergence is achieved. Convergence is reached when the parameter selection procedure suggests an identical parameter subset as obtained in the previous iteration and no further progress is made in the parameter estimation objective function. The effect of the nominal parameter values on the parameter subset selection was also studied by Weijers and Vanrolleghem (1997). In this study, no important influence of the nominal parameter values could be detected. However, this might not always be

the case.

Investigating potential bias effects of fixed parameters is also a potentially useful step in the proposed procedure. Dependencies between parameters are a logical consequence of the structural and practical identifiability problems for which the proposed procedure is actually meant. Because of these dependencies, only a certain subset of parameters can be selected for estimation, the other parameters have to be assumed known. This is however not a realistic assumption and the assumed values will obviously influence the estimated values. It is therefore important to stress that the obtained parameter estimates are valid conditional on the fact that the fixed parameters are correct. In order to overcome this problem, it is suggested to investigate which of the fixed parameters could be included in the set without causing too much increase in correlation while at the same time providing a better model fit. The investigation of bias effects might also be useful in situations where estimated parameter values have unrealistic values caused by an “incorrect” fixed value of other parameters.

4.4.2.3 Testing the proposed procedure

In this section a step-by-step illustration of the proposed calibration procedure will be given, using the same virtually generated data as described in Section 4.3.2.

Step 1: Assume nominal parameter values

Initial parameter values were obtained by significantly perturbing the default AM2 parameter values. This was done to ensure a realistic case study where initial model simulations and measured data differ significantly. The initial values of the 6 state variables were obtained by running a long-term steady-state simulation using an averaged influent file and were kept unchanged throughout the remainder of the illustration. The “true” and initial model parameters are listed in Table 4.4. Figure 4.9 shows the virtually generated data together with the model simulations using the initial parameter values. The parameter estimation objective value corresponding to this initial state equals $5.97E+5$.

Step 2: Calculate FIM based on the nominal parameter values

Using the nominal parameter values, the FIM was calculated using the sensitivities of the 6 measured variables to the 13 parameters and the measurement error characteristics given by Table 4.3. The sensitivity calculations were performed using a central difference technique.

Step 3: Pre-selection of the most influential parameters

Since only 13 parameters are involved in this case study, no pre-selection of the most influential parameters was required.

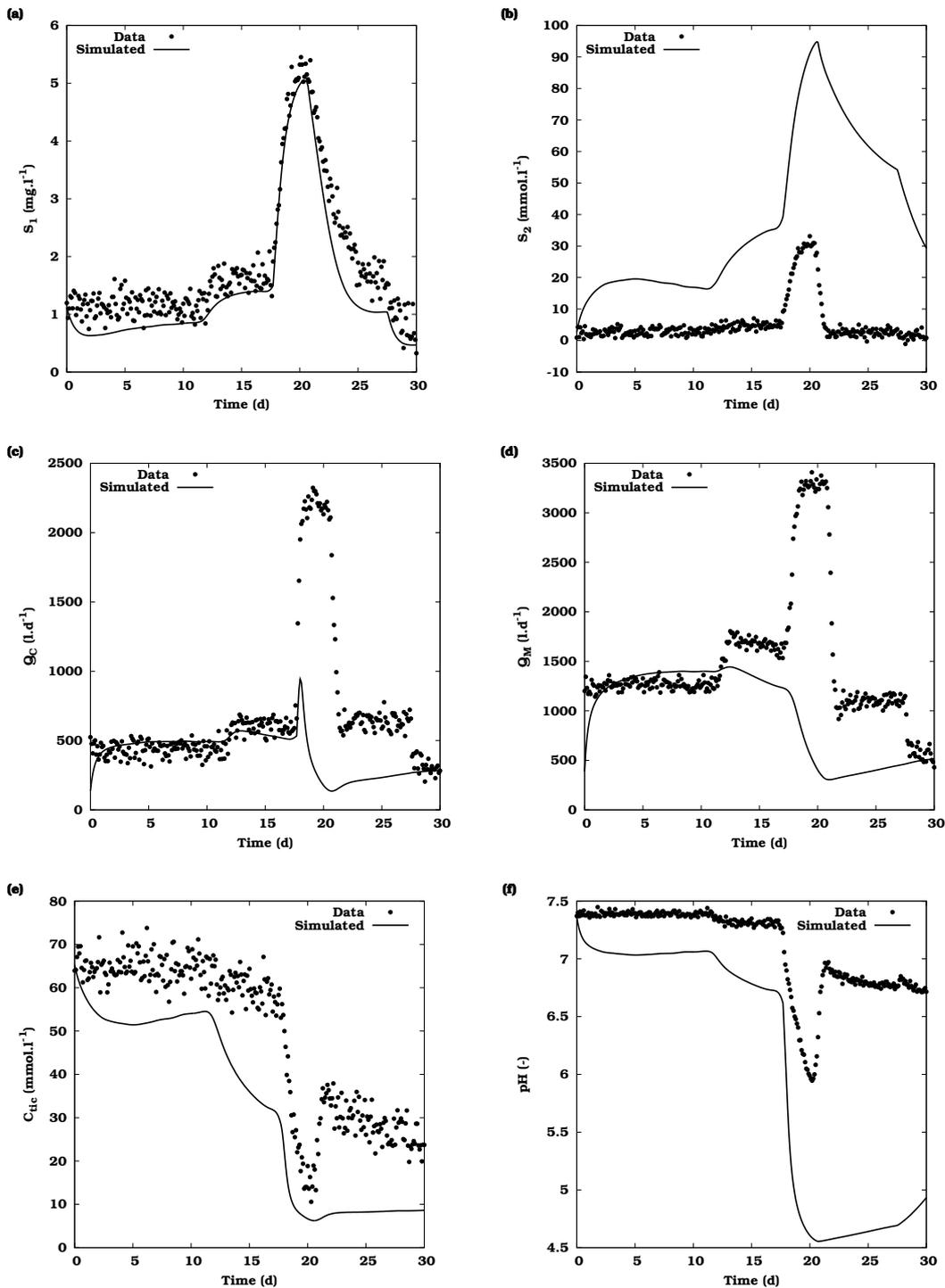


Figure 4.9: AM2 model simulations using the initial parameter values and virtually generated data based on the “true” parameter values and measurement error characteristics described in Table 4.3.

Table 4.4: Overview of the AM2 model parameters including the "true" model parameters (True), the values used as a starting point for the calibration (Initial), the optimal values after calibration (Optimal) and the optimal values after the investigation of possible bias effects (Optimal + bias).

Name	Unit	"True"	Initial	Optimal	Optimal + bias
k_1	–	42.14	60	60	60
k_2	$mmol.g^{-1}$	116.5	30	30	30
k_3	$mmol.g^{-1}$	268	130	130	149.3
k_4	$mmol.g^{-1}$	50.6	60	60	60
k_5	$mmol.g^{-1}$	343.6	200	200	200
k_6	$mmol.g^{-1}$	453	300	234.9	265.6
$\mu_{1,max}$	d^{-1}	1.2	1.5	1.27	1.29
$\mu_{2,max}$	d^{-1}	0.74	0.5	1.25	1.27
K_{S1}	$g.l^{-1}$	7.1	5	5	5
K_{S2}	$mmol.l^{-1}$	9.28	15	15	15
K_{I2}	$mmol.l^{-1}$	256	150	150	150
α	–	0.5	0.7	0.7	0.7
k_{1a}	d^{-1}	19.8	5	61.6	22.0

Step 4: Subset calculations

Based on the FIM calculated in step 2, sub-sampled FIMs were formed for all possible parameter subsets with sizes ranging from 2 – 13. For each parameter subset the D-optimal criterion was calculated. Next, each sub-sampled FIM was inverted and used to calculate the correlation matrix. From this matrix, the highest absolute correlation value was retained.

Figure 4.10 shows the calculated D-optimal criterion values as function of the highest correlation values for each of the parameter subsets. This clearly shows that only a small number of parameter subsets exist that have a highest correlation value which is relatively low. Already, this is a first sign that identifiability problems are to be expected. It can also be concluded that parameter subsets with a high ($> 1E+15$) D-optimal criterion value generally also have a high value for the highest correlation value.

The relation between the highest correlation values and the subset size is illustrated by Figure 4.11. Based on this figure, it is clear that subsets with increasing size also have increasing highest correlation values.

Step 5: Subset selection

At this point, all parameter subsets corresponding with a highest correlation value larger than 0.5 are eliminated. Figures 4.12 and 4.13 show, for the remaining parameter subsets, the highest correlation values and the D-optimal criterion values as function of parameter subset size. From these subsets, the subset with the highest D-optimal criterion value was chosen as the final subset which will be identified. In this case, the final subset consists of 4 parameters: the yield coefficient for CH_4 production (k_6), the

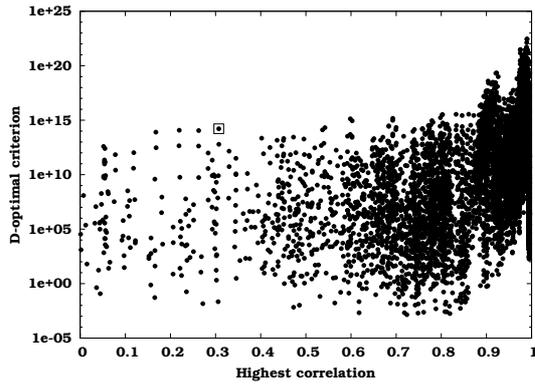


Figure 4.10: Calculated D-optimal criterion values as function of the highest correlation value for all parameter subsets. The final selected subset (in step 5) is indicated by a square.

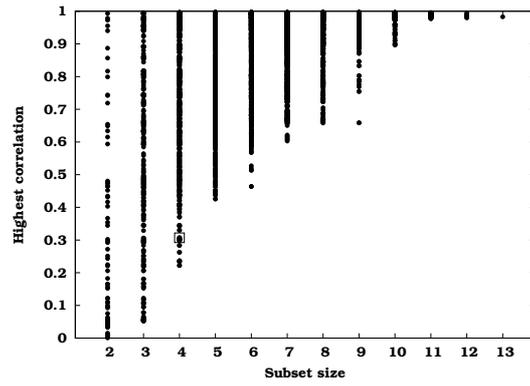


Figure 4.11: Highest correlation values as function of parameter subset size. The final selected subset (in step 5) is indicated by a square.

maximum growth rate for acidogenic biomass ($\mu_{1,max}$), the maximum growth rate for methanogenic biomass ($\mu_{2,max}$) and the liquid-gas transfer coefficient (k_{la}). The FIM, correlation matrix and corresponding D-optimal criterion and the highest correlation value are given in Table 4.5. The final selected parameter subset is also indicated in Figures 4.10, 4.11, 4.12 and 4.13 by a square symbol.

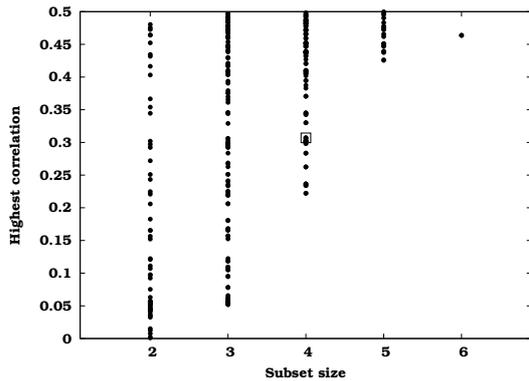


Figure 4.12: Highest correlation values as function of parameter subset size for parameter subsets with highest correlation values below 0.5. The final selected subset is indicated by a square.

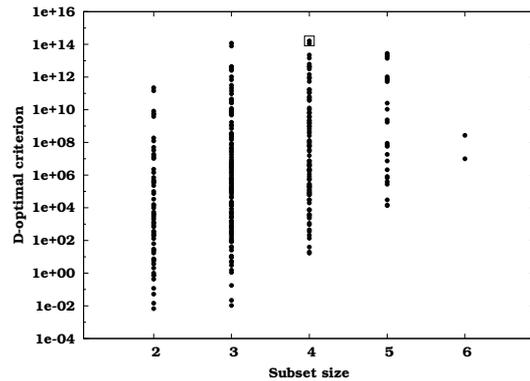


Figure 4.13: D-optimal criterion value as function of parameter subset size for parameter subsets with highest correlation values below 0.5. The final selected subset is indicated by a square.

Table 4.5: FIM, D-optimal criterion value, correlation matrix and highest correlation value for the final selected parameter subset.

FIM				
	k_6	$\mu_{1,max}$	$\mu_{2,max}$	$k_{l,a}$
k_6	$1.63E+00$			
$\mu_{1,max}$	$1.80E-01$	$2.07E+04$		
$\mu_{2,max}$	$1.43E+03$	$-2.53E+04$	$1.07E+07$	
$k_{l,a}$	$7.61E+00$	$-1.19E+02$	$1.71E+04$	$5.65E+02$

D-optimal criterion: $1.64E+14$

Correlation matrix				
	k_6	$\mu_{1,max}$	$\mu_{2,max}$	$k_{l,a}$
k_6	1.00			
$\mu_{1,max}$	-0.03	1.00		
$\mu_{2,max}$	-0.31	0.05	1.00	
$k_{l,a}$	-0.19	0.03	-0.15	1.00

Highest absolute correlation: 0.31

Step 6: Model calibration

In this step, the 4 parameters of the final selected parameter subset are calibrated using the Praxis optimisation algorithm (Brent, 1973). Parameters which were not estimated were kept at their initial value. The lowest weighted sum of squared errors objective value obtained was 7687, already quite close to the expected value of 1806 (see Section 4.3). Visual inspection of simulated values compared to the measured values (not shown here) showed that variable Q_C was the main responsible for this difference.

The optimal parameter values, 95% confidence interval, estimation errors and parameter correlations are listed in Table 4.6. The parameter estimation errors ($e_{\hat{\theta}}$) were calculated based on Equation 1.17 and express (as a percentage) the relative size of the 95% confidence interval with respect to the parameter value. For easy comparison with the "true" and initial values, the optimal parameter values are also listed in Table 4.4. Analysing this table shows that only the value of $\mu_{2,max}$ is close to the "true" parameter value. The significant differences observed for the other 3 parameters result from the deviations of the fixed parameters values from their default values. These deviations are propagated through various parameter dependencies to the estimated parameters. As a consequence, the small estimation errors found (Table 4.6), should be regarded with some caution. These can be regarded correct assuming the fixed parameters are the "true" parameters, which is not the case. From a comparison between the correlation matrices listed in Tables 4.5 (expected) and 4.6 (true) it can be concluded that the correlations derived from the sub-sampled FIM show the same charac-

teristics as the true correlations at the optimum. However, some (minor) differences can be observed, probably caused by the difference between the Hessian matrix and the FIM due to the non-optimal model fit. The agreement between the correlation values gives confidence to the applied method of selecting parameter subsets based on correlation values.

Table 4.6: Optimal parameter values, 95% confidence interval, estimation errors and parameter correlations for the final selected parameter subset.

Parameter	Value	95% confidence interval	Error ($e_{\hat{\theta}}$) (%)	Correlations			
				k_6	$\mu_{1,max}$	$\mu_{2,max}$	$k_{l,a}$
k_6	234.93	1.67	0.71	1.00			
$\mu_{1,max}$	1.27	0.021	1.63	-0.03	1.00		
$\mu_{2,max}$	1.25	0.0039	0.31	-0.15	0.01	1.00	
$k_{l,a}$	61.60	8.49	13.78	-0.21	0.03	-0.36	1.00

Step 7: Investigate potential bias effects

At this point, all parameter subsets containing one additional parameter on top of the already selected parameters are investigated in order to determine the extent of the influence of the fixed parameters on the parameters from the selected subset. For each of these subsets, 9 in total (13 – 4 remaining parameters), the D-optimal criterion is calculated and the correlation matrix analysed. An overview of this analysis is given in Table 4.7. In this table the name of the added parameter is listed together with the D-optimal criterion value of the extended subset and the parameter pairs whose correlation value is larger than the highest correlation value of the final selected parameter subset (0.31). Several important and interesting conclusions related to parameter dependencies can be drawn from this table.

Firstly, the correlation between parameter combination $k_6 \leftrightarrow \mu_{2,max}$ increases in nearly all extended subsets. This dependency can easily be derived from the model equations where k_6 (yield for NH_4 production) and μ_2 (the growth rate of methanogenic bacteria) always appear as a couple. This implies a connection between k_6 and $\mu_{2,max}$ because Haldane kinetics are assumed. However, this correlation is not very strong because μ_2 (and thus $\mu_{2,max}$) is also found in combination with other model parameters. The influence of using Haldane kinetics is obviously also very pronounced for parameter pairs $K_{S2} \leftrightarrow \mu_{2,max}$ and $K_{I2} \leftrightarrow \mu_{2,max}$. It should be noted that these parameter combinations would be better identifiable if the experiment would have excited S_2 more. Similar conclusions can be made for parameter pair $K_{S1} \leftrightarrow \mu_{1,max}$ which appears in the Monod growth term of the acidogenic bacteria.

Other important interactions between parameters can be observed for the triplet $k_2 \leftrightarrow k_6 \leftrightarrow \mu_{2,max}$ and the quadruplet $k_3 \leftrightarrow k_6 \leftrightarrow \mu_{2,max} \leftrightarrow k_{l,a}$. These

Table 4.7: Influence of an additional fixed parameter on the properties of the extended parameter subset.

Name	Parameter pairs	Absolute correlation	D criterion
k_1	$k_1 \leftrightarrow \mu_{2,max}$	0.47	$2.81E+13$
	$k_6 \leftrightarrow \mu_{2,max}$	0.40	
k_2	$k_2 \leftrightarrow k_6$	0.53	$3.33E+13$
	$k_2 \leftrightarrow \mu_{2,max}$	0.71	
	$k_6 \leftrightarrow \mu_{2,max}$	0.56	
k_3	$k_3 \leftrightarrow k_6$	0.71	$9.03E+14$
	$k_3 \leftrightarrow k_{Ia}$	0.78	
	$k_6 \leftrightarrow k_{Ia}$	0.64	
	$k_6 \leftrightarrow \mu_{2,max}$	0.32	
k_4	$k_4 \leftrightarrow k_{Ia}$	0.64	$3.04E+12$
	$k_6 \leftrightarrow \mu_{2,max}$	0.31	
k_5	$k_5 \leftrightarrow k_{Ia}$	0.80	$2.57E+14$
	$k_6 \leftrightarrow \mu_{2,max}$	0.31	
K_{S1}	$K_{S1} \leftrightarrow \mu_{1,max}$	0.99	$1.67E+15$
	$k_6 \leftrightarrow \mu_{2,max}$	0.31	
K_{S2}	$K_{S2} \leftrightarrow \mu_{2,max}$	0.95	$1.10E+16$
K_{I2}	$K_{I2} \leftrightarrow \mu_{2,max}$	0.94	$5.79E+13$
	$k_6 \leftrightarrow \mu_{2,max}$	0.35	
α	$\alpha \leftrightarrow \mu_{1,max}$	0.55	$3.03E+19$
	$\alpha \leftrightarrow \mu_{2,max}$	0.99	
	$\mu_{1,max} \leftrightarrow \mu_{2,max}$	0.55	
Reference:		0.31	$1.64E+14$

parameters are all related to processes involving VFA's (S_2) which are transformed by methanogenic bacteria to CH_4 and CO_2 .

Finally, the very high dependency between α (the fraction of biomass in the liquid phase) and the maximum biomass growth rates is explained by the fact that these parameters directly influence the biomass concentration in the reactor. The growth rates are related to biomass growth and α accounts for the efflux of biomass via the digester effluent.

Based on Table 4.7 it can be concluded that k_3 is the parameter which causes the least increase in highest correlation value ($0.31 \rightarrow 0.78$) while at the same time results in an increased D-optimal criterion value, compared to the D-optimal criterion value of the final selected subset. The model was recalibrated using the parameter subset extended with k_3 in order to try and find a better "fixed" value for k_3 to improve as such the overall model calibration. This calibration resulted in an objective value of 2891, a significant improvement compared to the calibration using the first selected subset. The close agreement between simulation and measurements is shown in Figure 4.14. The optimal values for this calibration are shown in Table 4.4. More detailed confidence information and correlations for the parameters of

the selected subset are shown in Table 4.8. This table shows that the values for k_6 and $k_{l a}$ have changed significantly compared to Table 4.6 and that the estimation errors for all parameters decreased. The correlation structure, however, remained almost identical.

Table 4.8: Optimal parameter values, 95% confidence intervals, estimation errors and parameter correlations for the final selected parameter subset after recalibration with the additional parameter k_3 .

Parameter	Value	95% confidence interval	Error ($e_{\hat{\theta}}$) (%)	Correlations			
				k_6	$\mu_{1,max}$	$\mu_{2,max}$	$k_{l a}$
k_6	265.59	1.17	0.44	1.00			
$\mu_{1,max}$	1.29	0.013	1.02	-0.03	1.00		
$\mu_{2,max}$	1.27	0.0027	0.21	-0.16	0.00	1.00	
$k_{l a}$	22.04	0.90	4.10	-0.22	0.05	-0.35	1.00

The quality of the model fit can also be investigated by calculating the residuals between the generated data and the model simulations. These residuals are shown in Figure 4.15 for each of the variables. Based on these graphs it can be concluded that the fit on C_{tic} and pH is very good, i.e. the residuals are randomly distributed around 0. For Q_C and Q_M the fit is also adequate for the first part of the data (until day 20). After that, the model simulations underpredict the measured data (positive residual values). For variables S_1 and S_2 the model fit is less good than for the other variables, although still very acceptable. This is clearly shown by the trend in the residuals for S_1 and the systematic overprediction of the model for S_2 (negative residual values). These deviations between the model predictions and the data are also reflected when the Hessian matrix and the FIM are calculated at the optimal parameter values. The ratios between the elements of both matrices are shown in Table 4.9. Theoretically, these ratios should be equal to 2. However, for some parameters a (slight) deviation from this value is observed, indicating that the model did not perfectly fit the data, in this case, due to misspecified fixed parameter values.

Table 4.9: Ratio between the Hessian matrix and the FIM calculated at the optimal parameter values listed in Table 4.8.

	k_6	$\mu_{1,max}$	$\mu_{2,max}$	$k_{l a}$
k_6	2.00			
$\mu_{1,max}$	1.92	1.87		
$\mu_{2,max}$	2.04	1.97	1.99	
$k_{l a}$	1.99	1.97	1.99	1.97

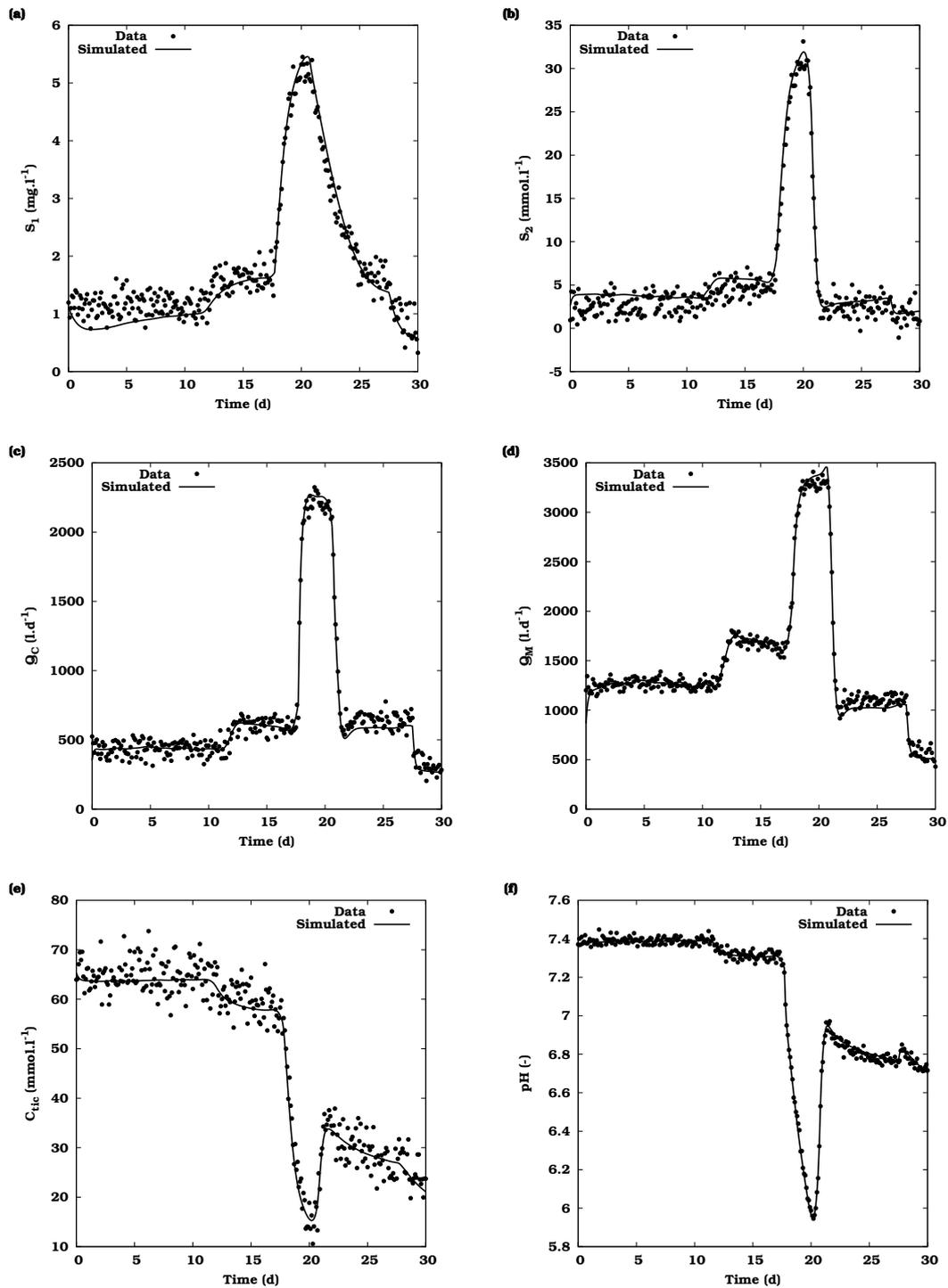


Figure 4.14: AM2 model simulations using the optimal parameter values (after including the bias effect of k_3) and virtually generated data based on the “true” parameter values and measurement error characteristics described in Table 4.3.

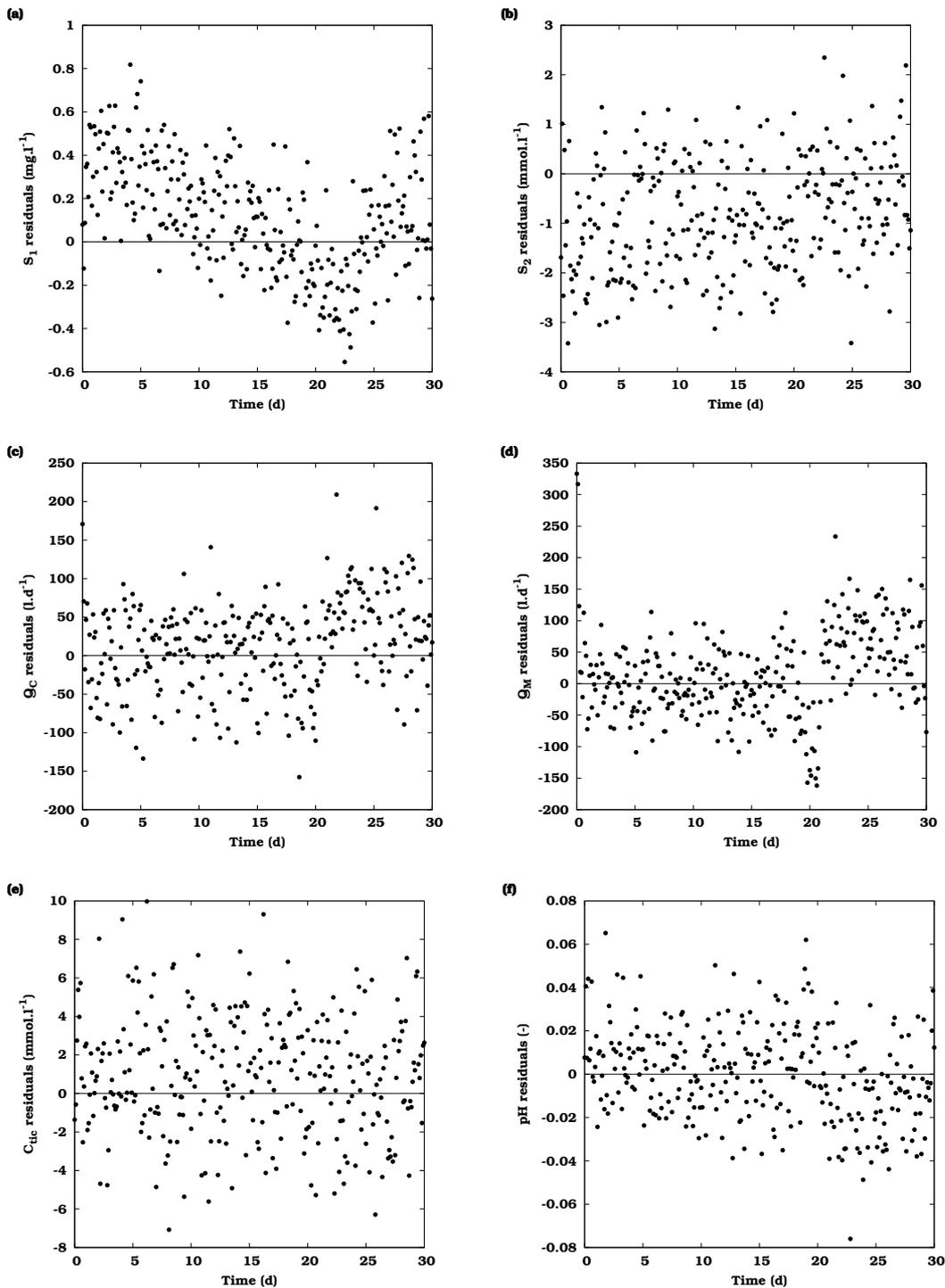


Figure 4.15: Residuals between the AM2 model simulations using the optimal parameter values (after including the bias effect of k_3) and virtually generated data based on the “true” parameter values and measurement error characteristics described in Table 4.3.

Steps 8-9: Transfer optimal parameter values to model and re-iterate

Based on the calibrated model (using the parameter values listed in the last column of Table 4.4) a new FIM was calculated. Based on this FIM, a new parameter subset selection was performed. This resulted in the same parameter subset as was found in the first iteration. It was therefore concluded that no further calibration was required. This also indicates that the parameter subset selection was rather insensitive to the nominal parameter values. However, this conclusion should not be generalised.

4.5 Conclusions

This chapter focused on the discussion of the relationship between the Hessian matrix and the FIM and the practical use of these matrices in parameter estimation.

As a first part, several numerical techniques to calculate the Hessian matrix were discussed. Attention was paid to the quadratic approximation and the Richardson's extrapolation technique. It was also demonstrated that the use of the complex-step derivative approximation method could be used to calculate the Jacobian matrix (first derivatives of the objective function) and could therefore significantly reduce the number of objective function evaluations required for the Richardson's extrapolation technique. A comparison of the three proposed methods revealed that each technique was able to produce a correct Hessian matrix although the combined complex-step approximation and Richardson's extrapolation technique was most economical with respect to the number of required objective function evaluations.

In a second part, the theoretical relationship between the Hessian and the FIM was presented. This relationship was illustrated in a numerical example in which a 19×19 Hessian and FIM were calculated. The ratio between the Hessian and FIM elements approached the theoretical value of 2 very closely, also confirming the correctness of the calculation of the Hessian matrix and the FIM.

The final part of this chapter discussed the problem of model identifiability and parameter subset selection. After a short introduction on identifiability, a critical review of several methods for selecting identifiable parameter subsets was given. A new method based on FIM-related properties was also proposed and successfully applied to an anaerobic digestion case study using artificially generated data. This practical application showed that an adequate model fit could already be obtained by estimating only 4 parameters out of the total of 13 model parameters. An important step in the procedure is the analysis of the effect of the values of the fixed parameters on the estimated parameter values. Based on this analysis, the fixed value of a 5th parameter was adjusted in order to improve the model fit. Although the proposed procedure proved to work, it has to be stressed that the use of the FIM D-optimal criterion as a measure for the parameter subset selection is probably not a ideal choice. Comparison of the D-optimal criteria

for parameter subsets with different sizes and different parameters cannot easily be mathematically justified.

Chapter 5

Single-objective optimal experimental design

5.1 Introduction

Designing experiments in an intelligent way can greatly improve the quality of the obtained data. It involves making choices related to the experimental degrees of freedom. It thus has to be decided whether, where and how the system under study will be manipulated and where, how and when measurements will be performed on this system. Very often, this is done based on expert knowledge or based on a manual or grid search of the available experimental degrees of freedom (Fadale et al., 1995; Vanrolleghem et al., 1995; Ossenbruggen et al., 1996; Baetens et al., 2000; Nahor et al., 2003). These types of search methods become very inefficient with increasing number of experimental degrees of freedom. Therefore, optimization algorithms can be used to solve these complex design problems. The type of optimization algorithm that should be used is determined by the nature of the design problem. In this chapter, the general optimal experimental design optimization problem will be described and the use of different types of optimization algorithms discussed. A practical application will also be presented in which measurement campaigns are designed for a sequencing batch reactor (SBR).

5.2 The optimal experimental design optimization problem

In Chapter 1, scalar properties of the Fisher Information Matrix (FIM) were used as measures for the information content of an experiment. To find the optimal experiment, an optimization algorithm can be used to find those values of the experimental degrees of freedom that minimize or maximize certain FIM properties. In order to understand the complexity of an optimal experimental design optimization problem it is first needed to understand how the different experimental degrees of freedom (manipulations and measurements) are related to the FIM, which is given by Equation 5.1.

$$FIM = \sum_{i=1}^N \left(\frac{\partial y_i}{\partial \theta}(\theta) \right)^T \times Q_i^{-1} \times \left(\frac{\partial y_i}{\partial \theta}(\theta) \right) \quad (5.1)$$

The FIM is composed of two components: sensitivity functions ($\partial y/\partial \theta$) and measurement errors (Q). A schematic view of the construction of a hypothetical FIM involving three measurement variables (y_1, y_2 and y_3) and two parameters (θ_1 and θ_2) is shown in Figure 5.1.

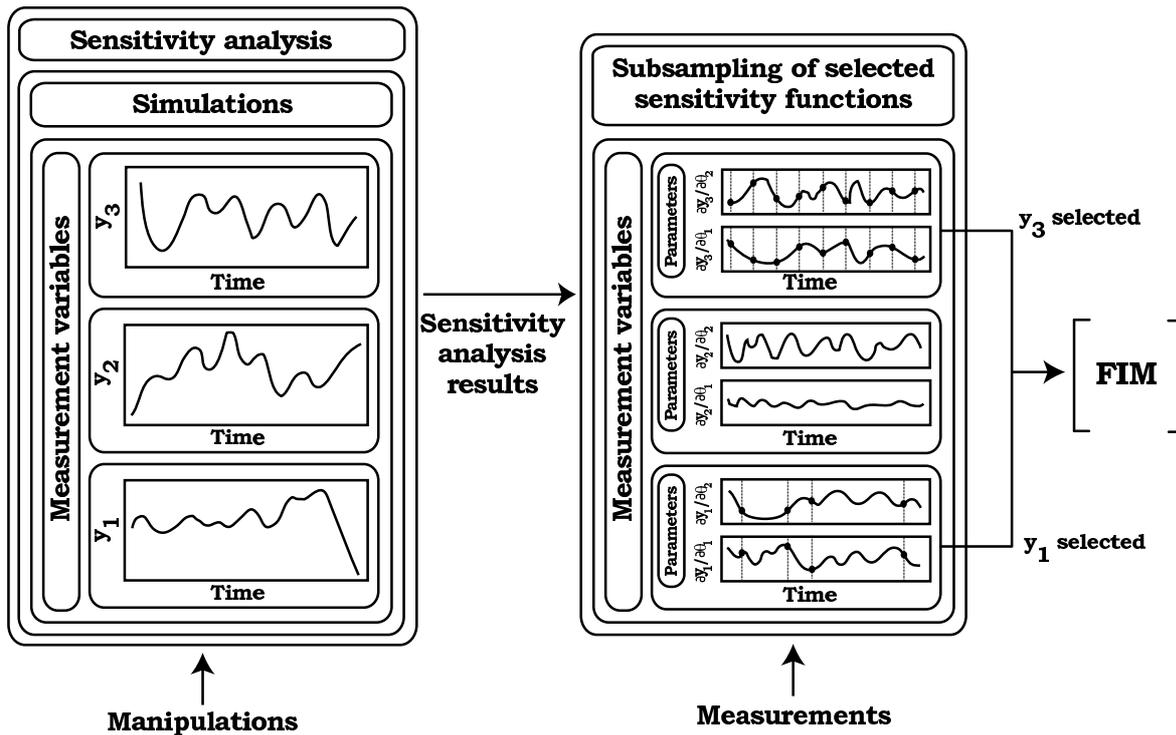


Figure 5.1: Schematic overview of the construction of a hypothetical FIM for a system involving three measurement variables (y_1, y_2 and y_3) and two parameters (θ_1 and θ_2). From the three measurement variables, only two (y_1 and y_3) are selected to form the FIM. The parts of the scheme where the manipulation and measurement experimental degrees of freedom act on are indicated.

One of the FIM components are local sensitivity functions which are the partial derivatives of the simulated model variables (3 in Figure 5.1) to the parameters of interest (2 in Figure 5.1). In the particular case of EAST, the sensitivity functions are calculated using finite differences (or the complex step derivative approximation method) since the equations of the sensitivity functions are assumed not to be available, as explained in Chapter 3. Therefore, model simulations need to be performed based on a model input, parameter values and initial conditions of the derived state variables. This is the point where the experimental manipulations have an influence on the FIM calculations. Experimental manipulations are related to excitation signals that will act on the system. Examples of these manipulations are input flow rate profiles, temperature set-points, influent concentrations, pulses of substrate, etc. Since the number of possible excitation signals is very large and very case specific, some creative thinking is required to come up with useful manipulations. For linear systems, techniques have been developed to determine which input signals would be useful (Ljung, 1999). Unfortunately, this theory has, to my knowledge, not yet been extended to the case of dynamic non-linear models. The manipulations are typically controlled by setting certain parameter values, inputs or initial conditions. In this way, the manipulations determine the trajectories of the model simulations and thus also the trajectories of the sensitivity functions.

The next step in calculating a FIM is to select which measurement variables are to be used, depending on the measurement experimental degrees of freedom. Depending on these experimental degrees of freedom, more sensitivity functions might have been calculated than will be used in the final FIM calculations. This can be the case, for instance, when a single measurement could be performed in one of several available locations (represented by different measurement variables) in a wastewater treatment plant. Depending on which measurement location is chosen, the corresponding measurement variable (and its sensitivities) will be used for the FIM calculation. This is also shown in Figure 5.1 where only two measurement variables (y_1 and y_3) out of three are used for the FIM calculation. Once it is decided which measurement variables will be used, the sensitivity functions need to be sampled at certain “measurement” points. This is also determined by the measurement experimental degrees of freedom. These sub-sampled sensitivity function values are then combined with the measurement errors and summed to form the final FIM.

The above discussion shows how the FIM is calculated based on the particular values of the experimental degrees of freedom. Depending on whether some of these need to be optimized (in order to minimize/maximize certain FIM properties), several cases can be distinguished.

The easiest case is when the values of all experimental degrees of freedom are known a priori and no optimization is required. In this case, the values of the manipulations need to be set in the model, and the sensitivity functions calculated (based on model simulations) at the required points in time for each required measurement variable. The FIM can then be calculated

based on these sensitivity values and the measurement errors.

The most complex case is when both manipulation and measurement experimental degrees of freedom require optimization. An optimization algorithm would come up with certain values for the manipulation and measurement experimental degrees of freedom. The values of the manipulation experimental degrees of freedom would then be used as a basis for the sensitivity analysis while the values of the measurement experimental degrees of freedom would be used to subsample the sensitivity functions and calculate the FIM. This would require a (costly) sensitivity analysis for each proposed set of manipulation and measurement experimental degrees of freedom. However, the knowledge of the fact that the different experimental degrees of freedom act on different stages of the FIM calculation can be used in an intelligent way to increase the efficiency of the optimization procedure. It is therefore proposed to split the complete optimization problem into a cascade of two optimization problems. An outer optimization loop which involves the optimization of manipulation experimental degrees of freedom and an inner optimization loop only responsible for the optimization of measurement experimental degrees of freedom. An overview of this idea is shown in Figure 5.2.

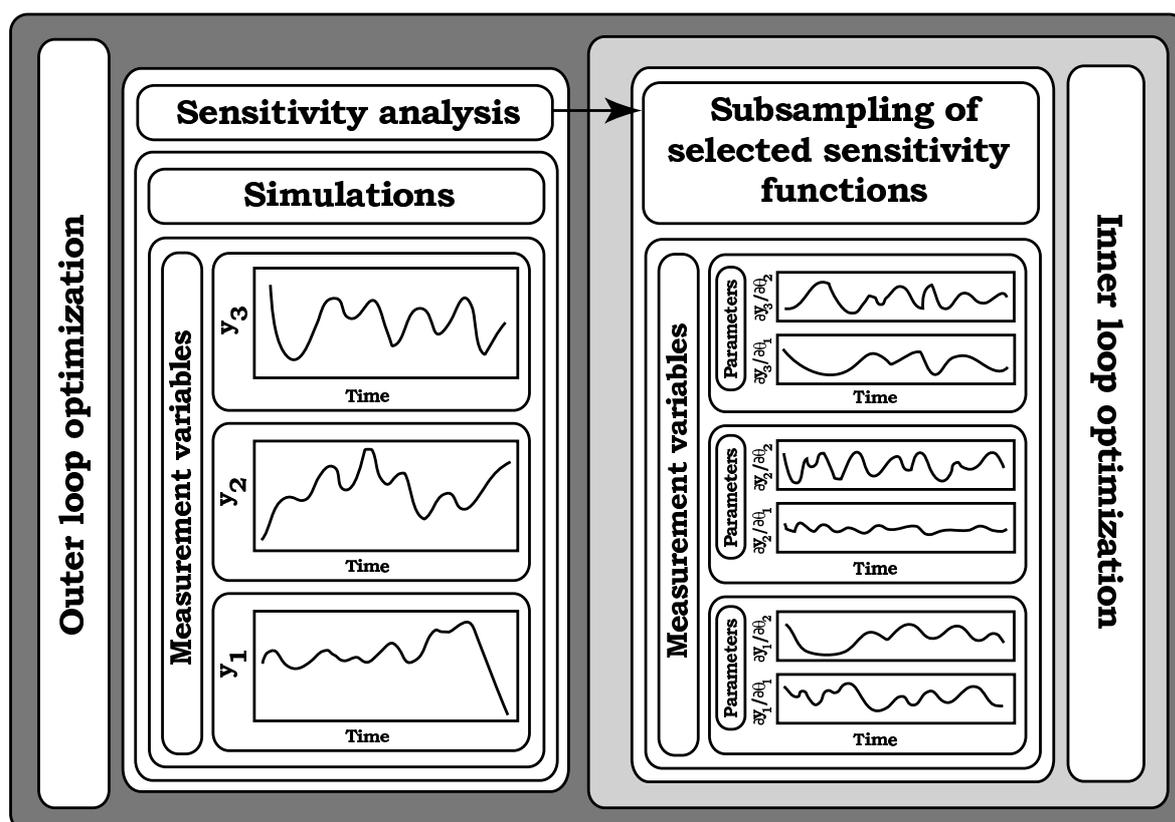


Figure 5.2: Layout of the optimal experimental design optimization problem consisting of two optimization loops. The inner loop optimizes measurements and the outer loop optimizes manipulations.

In practice, optimizations of manipulations would be handled by one optimization algorithm (outer loop) which would propose different sets of values for the manipulations. Each of these sets of manipulation values would then be used as the basis for a sensitivity analysis. Based on each of these sensitivity analyses another optimization algorithm is used to optimize the measurement experimental degrees of freedom (inner loop). This optimization takes less time to complete than the outer loop optimization because only points need to be sub-sampled from the (already available) sensitivity function trajectories in order to calculate the FIM. The inner loop optimization then results in a FIM corresponding to the optimal values for the measurement experimental degrees of freedom belonging to a certain set of values of the manipulation degrees of freedom. The outer loop optimization algorithm is then further responsible for finding the optimal values of the manipulation experimental degrees of freedom. It is expected that splitting up the optimization problem in this way will improve the optimization speed, convergence and numerical stability since only several relatively small search spaces need to be analysed compared to one extremely large search space (probably characterized by an increased number of local minima and discontinuities). Beside these benefits, another advantage is that different optimization algorithms or different algorithm settings can be used for each optimization loop.

Two other cases of experimental design optimization problems can be distinguished: (1) the case where only manipulation experimental degrees of freedom need to be optimized and (2) the case where only measurement experimental degrees of freedom need to be optimized. For the second case, only one sensitivity analysis needs to be performed (based on the fixed manipulation values) followed by a relatively fast optimization of the measurement variable selection and measurement point identification (inner loop optimization only). The case where only manipulation experimental degrees of freedom need optimization, involves the execution of a sensitivity analysis for each proposed set of manipulation values (outer loop optimization only). For each of these sensitivity analyses the FIM is calculated based on the fixed values of the measurement experimental degrees of freedom.

Other important factors which determine the nature of the experimental design optimization problem are the types of the variables that describe the experimental degrees of freedom and their constraints. Three categories of variables are distinguished here: continuous variables, discrete variables and variables described by a set of values. Examples of experimental degrees of freedom which can be described by a continuous variable are a temperature set-point of a reactor or a substrate concentration. Experimental degrees of freedom described by discrete variables are for example the sampling times of certain measurements. It would be practically useless to determine the time of optimal sampling with an accuracy of microseconds or lower if in practice the samples can only be taken every minute. The third category are variables which can only take a certain value from a set of available values. An example of this is the speed of a pump which only

has two settings: low or high. For continuous and discrete variables it is also important to be able to specify a lower and upper bound in order to prevent the optimization algorithm to propose values which are practically infeasible. All of these factors will determine the type of optimization algorithm that can be used to solve a certain design problem. This will be the topic of the next section of this chapter.

5.3 Using classical optimization algorithms to solve the OED problem

5.3.1 Short introduction

The “classical” optimization algorithms, often used in non-linear parameter estimation, are probably the most obvious first choice in order to try and solve an optimal experimental design (OED) problem. These algorithms have already been studied extensively and their strong and weak points are well documented. Beside this, their source code is readily available from Internet which makes them easy to incorporate in existing software. Since describing optimization algorithms is not the focus of this chapter, only a short overview will be given about the most used algorithms. More information and references to most existing algorithms can be found in Dochain and Vanrolleghem (2001).

Two main groups of “classical” optimization algorithms can be distinguished. A first group makes use of information about the gradient of the objective function with respect to the experimental degrees of freedom to be optimized. This is only analytically feasible in case the objective function can be written explicitly, but this is almost never the case. Therefore, the derivatives need to be approximated numerically, with all the corresponding problems. Well-known algorithms in this group are: the steepest descent method, the Gauss-Newton methods (e.g. SQP: sequential quadratic programming), the Levenberg-Marquardt method and quasi-Newton methods like BFGS (Broyden, Fletcher, Goldfarb and Shanno) and DFP (Davidon, Fletcher and Powell). Several applications of these algorithms in experimental design studies can be found in the literature, among others: Emery and Fadale (1996), Sedrati et al. (1999), Versyck et al. (1999), Bernaerts et al. (2000b), Bauer et al. (2000), Chung et al. (2000) and Faller et al. (2003).

A second group of optimization algorithms does not rely on derivative information of the objective function and are mostly easier to implement. Numerical problems associated with derivative calculations which can lead to convergence problems are also avoided. Typical algorithms in this category are: the Rosenbrock method, Brent’s algorithm also known as Praxis, the Simplex method and the Secant or DUD method. Some authors who have used these derivative-free algorithms to solve optimal experimental design problems are: Espie and Macchietto (1989), Baltes et al. (1994), Grijspeerdt

and Vanrolleghem (1999) and Cunha and Oliveira (2000). Two algorithms, Simplex and Praxis, which are implemented in EAST will now be discussed in more detail since they will be used in a comparative study in this chapter.

5.3.2 The Simplex optimization algorithm

The Simplex minimization method was first described by Spendley et al. (1962) and improved significantly by Nelder and Mead (1965) allowing irregular simplexes. A simplex is a geometrical figure consisting, in p dimensions, of $p + 1$ points (or vertices) interconnected by line segments forming polygonal faces. For a 2 variable optimization problem the simplex is a triangle. Each of these points represents a set of optimization variable values and corresponds with one objective function value. Starting from an initial simplex, elementary operations are performed on the vertex with the highest value (for a minimization problem) resulting in a new vertex in the p -dimensional space. Several elementary operations like reflection, expansion and contraction of the vertices can be used in order to migrate the simplex in the direction of the minimum, each time replacing the worst performing vertices by new ones. The algorithm is terminated if the relative difference between the objective function values of the vertices and the average value of the whole simplex is below a certain threshold value.

Figure 5.3 shows the progress of the Simplex algorithm for a hypothetical two-dimensional optimization problem. Contour lines represent points in the optimization variable space with equal objective function values. The initial simplex is given by points Y_0, Y_1 and Y_2 . Subsequent simplexes are formed by reflecting, expanding and contracting the simplex points until the final simplex consisting of points Y_{12}, Y_{13} and Y_{14} is reached. From this simplex, point Y_{14} has the lowest objective function value and is taken to be the minimum.

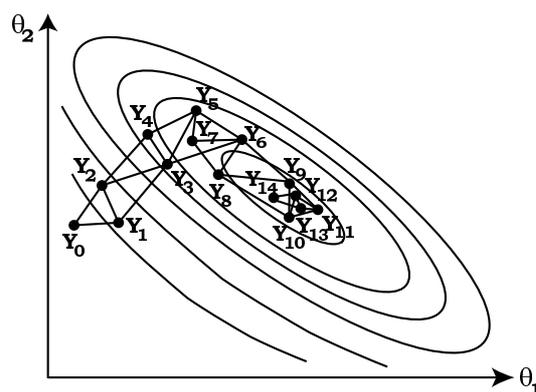


Figure 5.3: Example of the progress of the Simplex algorithm solving a two-dimensional optimization problem. The initial simplex consists of points 0, 1 and 2. Subsequent simplex points are numbered from 3 to 14. The final simplex consist of points 12, 13 and 14.

5.3.3 The Praxis optimization algorithm

The Praxis (principal axis) algorithm was developed by Brent (1973) and is an enhanced version of the direction set method of Powell (1964). It is based on a repeated combination of one-dimensional searches along a set of various directions. Updating the search directions is done by finding those directions that (1) will lead far along narrow valleys or (2) are “non-interfering” which means that they have the special property that a minimization along one direction is not “spoiled” by subsequent minimizations along another one. Brent also incorporated some random “jumps” into the procedure to avoid some “local minima” problems that Powell’s original algorithm suffered from.

In Figure 5.4 a typical sequence of searches is presented to illustrate the basic algorithm for a hypothetical two-dimensional optimization problem. The algorithm is started at an initial parameter set (point Y_0) and searches sequentially along each of the directions in direction set $\{n_1, n_2\}$ until a minimum is found. This leads to point Y_1 which is used to construct a new direction $n_3 = Y_0 - Y_1$. Minimum Y_2 is reached by searching along this direction. At this point, the oldest direction (n_1) is removed from the direction set and a new iteration is started with direction set $\{n_2, n_3\}$. This leads to point Y_3 which is again used to construct a new direction $n_4 = Y_2 - Y_3$. The search along this direction subsequently leads to minimum Y_4 . This sequence of steps continues until a stopping criterion is reached (e.g. maximum number of allowed iterations or lack of further decrease in objective function value).

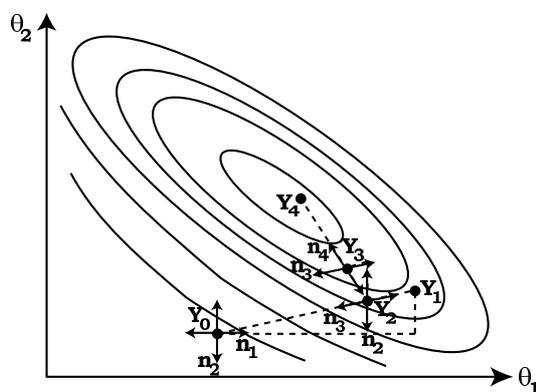


Figure 5.4: Example of the progress of the Praxis algorithm solving a two dimensional optimization problem.

5.3.4 Drawbacks of classical optimization algorithms

In Chapter 1 a relatively simple optimal experimental design problem was presented for the design of the feed time and concentration (two manipulation experimental degrees of freedom) of a fed-batch reactor. The objective function surfaces for the D-optimal and modified E-optimal FIM criteria (Figures 1.20 and 1.21) were characterised by many ridges and valleys

leading to several local minima. Since the “classical” optimization algorithms are also known as local optimization algorithms, problems related to the detection of local minima are to be expected when using these algorithms. It might therefore be needed to start the algorithm at different points in the experimental degrees of freedom space in order to find the global minimum. However, it can never be ensured that the true global minimum can be found, especially when the optimization problems become more complex. Several authors have encountered or acknowledge this issue for optimal experimental design related problems, e.g. Walter and Pronzato (1990), Munack (1991) and Emery and Fadale (1996).

As already discussed in Section 5.2, optimal experimental design problems can be formulated as combinations of continuous and discrete optimization variables which represent the experimental degrees of freedom that need to be optimized. This poses difficulties for the “classical” optimization algorithms which are mostly used to solve continuous problems and cannot handle this type of problem formulation.

Therefore, the next section will deal with the use of global optimization techniques, and more specifically genetic algorithms, in order to resolve the two main drawbacks of the “classical” optimization algorithms related to optimal experimental design optimization problems.

5.4 Using genetic algorithms to solve the OED problem

5.4.1 Classical genetic algorithms

5.4.1.1 Working principle and terminology

A genetic algorithm (GA) can be considered as a global optimization technique based on the genetic processes of life. Over many generations, natural populations evolve according to the principles of natural selection and “survival of the fittest”, first clearly stated by Charles Darwin in his “The Origin of Species”. Individuals in these populations compete with each other for resources such as food, water and shelter. Also, members of the same species often compete to attract a mate. Those individuals which are most successful in surviving and attracting mates will have relatively larger numbers of offspring. Poorly performing individuals will produce few or even no offspring at all. This means that the genes from the highly adapted, or “fit” individuals will spread to an increasing number of individuals in each successive generation. The combination of good characteristics from different ancestors can sometimes produce “super-fit” offspring, whose fitness is greater than that of either parent. In this way, species evolve to become more and more suited to their environment.

The mathematical analogon of these natural processes was first developed

by Holland (1975) and is described in detail in many other texts (e.g. Goldberg, 1989; Michalewicz, 1992). In contrast to the “classical” optimization algorithms, GAs work with populations of individuals and not single individuals, each representing a possible solution to a given problem. The representation of the individuals is case dependent and requires all optimization variables or genes (experimental degrees of freedom) to be encoded into a fixed-length string (also called chromosome) using a certain alphabet. Typically, the binary alphabet is used. Each of the individuals is assigned a fitness value by evaluating a user defined objective function (e.g. a FIM criterion value). A selection process then uses these fitness values to decide which chromosomes (parents) are used to form offspring. Next, the “parent” chromosomes are recombined using crossover and mutation operators resulting in offspring which will comprise the next generation of individuals. A crossover operator is typically applied to two parents and is used to swap genetic information between the parents and thus produces two new individuals inheriting some genes from each parent. Mutation is applied to each offspring individually and randomly alters one or more genes, potentially generating completely new individuals. Crossover and mutation operators are usually applied to the chromosomes selected for mating using a certain probability. This gives each individual a chance of passing on its genes without the disruptive nature of these operators. In other words, there is a probability that a very fit individual is able to transfer its genetic characteristics to the next generation without going through the potential destructive crossover and mutation processes. The process of selection, mating and creating new generations continues until the GA converges. One way to describe convergence is when the average fitness of all individuals in the population closely approaches or becomes equal to the fitness of the best individual.

As an illustration of the above described algorithm, a simple maximization problem of a parabolic objective function (Equation 5.2) is now presented. A GA is used to find the integer value x which maximizes the objective function y over the interval $0 - 15$. The maximum is situated at $x = 5$ and a graphical representation of the objective function is shown in Figure 5.5.

$$y = -x^2 + 10x + 100 \quad (5.2)$$

The starting point for each GA implementation is the encoding of the problem. In this case, x can be represented by a binary string of size 4. This covers the entire search region: chromosome 0000 represents $x = 0$ and chromosome 1111 represents $x = 15$. Once the encoding is fixed, an initial population of individuals can be constructed. In this case, a population size of 5 individuals was chosen and each of these individuals was chosen at random. The individuals are then assigned a fitness value based on the objective function evaluation, individuals closer to the maximum receiving higher fitness values. The binary (encoded), integer (problem formulation) and fitness values of the individuals of the first generation can be seen in Figure 5.6. Based on their fitness value, individuals 2 and 3 could be se-

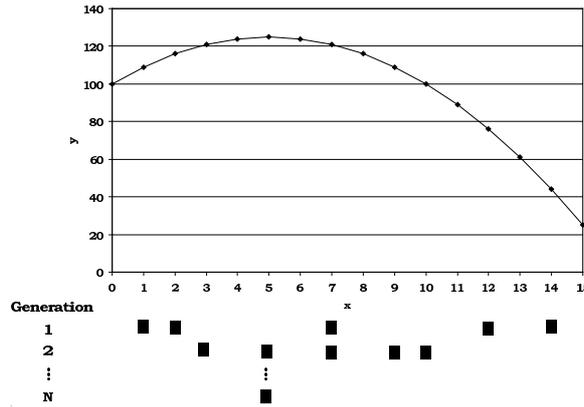


Figure 5.5: Parabolic test function and individuals of the GA population during the progress of the algorithm.

lected for crossover. This is also shown in Figure 5.6. A crossover point is selected at random and the parent chromosomes are split into two segments. The first segment of the first parent is then recombined with the second segment of the second parent and vice versa which leads to two new individuals. Next, the second offspring individual is selected for mutation. This results in flipping the value of a random bit of the string. The processes of selection, crossover and mutation are continued until a whole new population of individuals is formed. These are then assigned a fitness value by evaluating the objective function for each individual. Based on this new generation subsequent generations are created. From the evolution of the individuals of the different generations (shown in Figures 5.5 and 5.6) it can be seen that the individuals of the second generation are already situated closer to the maximum compared to the individuals of the first generation. This trend continues until the algorithm converges and all individuals have become the same, representing the maximum of the objective function.

Individual	Generation 1			Reproduction			Generation 2			... Generation N			
	Encoding	x	Fitness (y)	Encoding	x	Fitness (y)	Encoding	x	Fitness (y)	Encoding	x	Fitness (y)	
1	1100	12	76	crossover point			1010	10	100	0101	5	125	
2	0111	7	121	0111	→	0101	→	0101	5	125	0101	5	125
3	0001	1	109	0001	→	0011	→	0111	7	121	0101	5	125
4	1110	14	44	mutation			0011	3	121	0101	5	125	
5	0010	2	116				1001	9	109	0101	5	125	

Figure 5.6: Evolution of the population in a GA for a maximization problem of a parabolic objective function. The application of the crossover and mutation operator is illustrated for some individuals selected from generation 1.

GAs have been applied successfully in many fields. One of the reasons for their success is their ability to exploit the information accumulated about an initially unknown search space in order to bias subsequent searches into useful subspaces. This key feature makes them very attractive, particularly in large, complex and poorly understood search spaces like optimal

experimental design problems, where classical search tools are often inappropriate. However, only few applications of GAs in optimal experimental design problems are found in the literature (Syddall et al., 1998; Poland et al., 2001; Hariri et al., 2004).

5.4.1.2 Problems encountered when using binary-coded GAs

The binary representation used in the GAs described in Section 5.4.1.1 exhibits some drawbacks when applied to multidimensional, continuous problems requiring solutions with high numerical precision. Continuous optimization problems are characterized by the fact that the optimization variables can take any value. The required length of the bit-string to represent these problems becomes very large and the search-space is blown up. The dependency between the bit-string length, precision and search space size is given by:

$$P = \frac{b - a}{2^L - 1} \quad (5.3)$$

where P represents the precision (smaller values corresponding with higher precisions), L the length of the bit-string and a and b the lower and upper bound of the search space of a continuous variable which is to be represented by a binary string. Increasing the search space, increases P and thus decreases the precision of the encoding, if a fixed bit-string length is considered. For 100 variables with domains in the range $[-500; 500]$, where a precision of 6 digits after the decimal point ($P = 1E-06$) is required, the length of the binary solution vector is 3000 (100×30 ; $2^L = 1E+9 \rightarrow L = 30$ for each variable). This, in turn, generates an enormous search space of about $1E+1000$ (2^{3000}). Binary encoded GAs perform poorly in these cases. In the first stages, the algorithm will waste great efforts evaluating the less significant digits of the binary coded variables, although the neighbourhood of the global minimum has not yet been reached. Near the end of execution, the most significant digits will have converged and there is no need to waste efforts on them any more. However, this desired behaviour is not achieved by the GA because all bits are handled in a similar way. Some solutions have been suggested to overcome these problems by using different kinds of dynamic encodings, i.e. the encoding changes during the course of the algorithm execution (Whitley et al., 1991; Schraudolph and Belew, 1992; Oyama et al., 2001).

Another problem associated with binary encoding is the problem of redundancy. When a binary alphabet is assumed for coding variables belonging to a discrete set with a cardinal different from a power of two, some individuals may be redundant, e.g. their decodings correspond to values that do not belong to the domain of the variable. For example, consider a discrete variable which can take values from the range $[1; 5]$. To represent this problem in binary code, 3 bits are required. However, bit-strings 000, 110 and 111 do not correspond to values of the variable range. This causes problems when

executing the GA because applying crossover and mutation might result in redundant individuals. However, several solutions have been proposed to overcome these problems: (1) discard the redundant individuals, (2) assign a very low fitness value to the redundant individuals and (3) remap the redundant individuals to valid individuals.

5.4.2 Real-coded genetic algorithms

5.4.2.1 Background and working principle

The use of other encoding mechanisms beside binary-encoding (e.g. real-coded) has been controversial. Their rising use and success was somewhat surprising to researchers familiar with fundamental GA theory. The theoretical evidence which had shown that low cardinality alphabets were the most appropriate to be used, seemingly contradicted the empirical findings that other encoding types worked well in a large number of practical problems. Theoretical research performed by Goldberg (1991), Radcliffe (1991) and Radcliffe (1994) resulted in several theories which have opened the way to a better understanding of these types of GAs. However, it is beyond the scope of this work to go into more details on this particular aspect.

The basic idea behind real-coded GAs is that each variable of the optimization problem is represented by one real-value gene in the GA chromosome. This representation is particularly natural to describe optimization problems with variables in continuous domains. Examples of applications and in-depth information on real-coded GAs can be found in the excellent review of Herrera et al. (1998). The use of real-coded GAs makes it possible to use large or even unknown domains for the variables while the precision is only restricted to that of the computer on which the algorithm is executed. Since real-coded GAs are based on continuous variables, slight changes in the variables also cause slight changes in the objective function. This gives real-coded GAs the ability to locally “fine-tune” the solution.

5.4.2.2 Real-coded crossover and mutation operators

Beside the encoding of the optimization variables, the main difference between binary-coded and real-coded GAs can be found at the level of the crossover and mutation operators. The ability of these operators to split the chromosome at certain points or “flip” certain chromosome values has to be completely redefined when using real-values. Because of this, a whole range of different crossover and mutation operators was developed that describe how the genetic information of the parents is recombined to produce offspring in the case of crossover or how chromosome genes are altered in the case of mutation.

Most of the available crossover and mutation operators from literature are listed in Tables 5.1 and 5.2 respectively. These tables act as a reminder of

the diversity of the available operators. A taxonomical subdivision of the real-coded crossover operators can be found in Herrera et al. (2003) and a detailed description of each of these operators can be found in Rademaker (2004). For each the mutation and crossover operators classes, one operator will be described in more detail here in order to illustrate the general concept of real-coded GA operators.

Table 5.1: Real-coded GA crossover operators.

Name	Most relevant publication
Flat crossover	Radcliffe (1991)
Simple crossover	Wright (1991)
Arithmetical crossover	Michalewicz (1992)
BLX- α crossover (blended crossover)	Eshelman and Schaffer (1993)
Linear crossover	Wright (1991)
Discrete crossover	Muhlenbein and Schlierkamp-Voosen (1993)
Extended line crossover	Muhlenbein and Schlierkamp-Voosen (1993)
Extended intermediate crossover	Muhlenbein and Schlierkamp-Voosen (1993)
Wright's heuristic crossover	Wright (1990)
Linear BGA crossover	Schlierkamp-Voosen (1994)
Fuzzy connectives based crossover (FCB)	Herrera et al. (1994)
UNDX crossover (unimodal normal distribution)	Ono et al. (2000)
Guided crossover	Rasheed (1999)
Simplex crossover	Tsutsui et al. (1999)
Confidence interval based crossover	Boyer et al. (2001)

Table 5.2: Real-coded GA mutation operators.

Name	Most relevant publication
Random mutation	Michalewicz (1992)
Non-uniform mutation	Michalewicz (1992)
Real number creep	Davis (1991)
Muhlenbein's mutation	Muhlenbein and Schlierkamp-Voosen (1993)
Discrete modal mutation	Voight and Anheyer (1994)
Continuous modal mutation	Voight and Anheyer (1994)

A frequently used crossover operator is BLX- α (Eshelman and Schaffer, 1993). Offspring is generated based on the recombination of two parent chromosomes. Each of the genes or optimization variables of the offspring chromosome (o_i) is generated by choosing a randomly (uniformly) generated number from the interval:

$$[p_{i,min} - I_i \times \alpha; p_{i,max} + I_i \times \alpha] \quad (5.4)$$

This interval is formed based on the value of the i th gene of each parent, where $p_{i,min}$ is the smallest parent value and $p_{i,max}$ is the largest parent

value. The distance between both parents is given by $I_i = p_{i,max} - p_{i,min}$. A user defined factor α determines the extent of the interval around both parent values from which the offspring gene is selected. This value thus determines the tendency of the operator to generate individuals which could explore new zones in the search space. The influence of α on the interval is visualized in Figure 5.7.

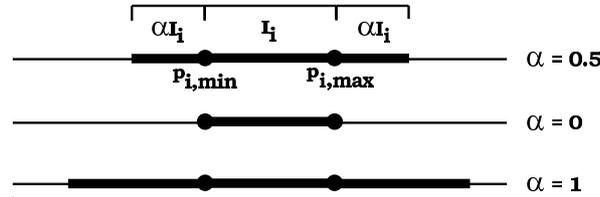


Figure 5.7: Graphical representation of the BLX- α crossover operator.

An example of a mutation operator is the non-uniform mutation operator (Michalewicz, 1992). This operator causes a gene (o_i) in an offspring chromosome to be mutated using following equation:

$$o'_i = \begin{cases} o_i + \Delta(t, b_i - o_i) & \text{if } \tau = 0 \\ o_i - \Delta(t, o_i - a_i) & \text{if } \tau = 1 \end{cases} \quad (5.5)$$

where τ is 0 or 1, randomly selected. The lower and upper bounds of the search domain of the i th gene are given by a_i and b_i respectively. The extent of the mutation is defined by the function $\Delta(t, y)$:

$$\Delta(t, y) = y \left(1 - r^{\left(1 - \frac{t}{g_{max}}\right)^u} \right) \quad (5.6)$$

In this function, t is the current generation of the GA, g_{max} the maximum number of generations, r a random number from the interval $[0; 1]$ and u a user defined parameter which determines the degree of dependency on the generation number. The function returns a value in the range $[0; y]$ such that the probability of returning a number close to zero increases as the algorithm advances. The size of the gene generating interval shall thus be smaller with the passing of generations. This property causes the operator to generate a uniform mutation in the initial space when t is small, and a very local mutation at a later stage, favouring local tuning.

Several studies have been performed in order to compare the different operators (e.g. Herrera et al. (1998); Blanco et al. (2001); Herrera et al. (2003)). From these results it is clear that the performance of the operators is largely case dependent and also much influenced by the user defined parameters. However, some crossover operators were found to perform generally better than others: BLX- α crossover, FCB crossover, Wright's heuristic crossover and linear crossover. The mutation operator which shows a good general performance is the non-uniform mutation operator.

5.4.2.3 Combined discrete and continuous optimization problems

It has been demonstrated that real-coded GAs are ideal candidates if an optimization problem consists of a large continuous search domain. However, this does not solve the problem of redundancy if discrete variables need to be optimized. Actually, this problem is even more pronounced for real-coded GAs compared to binary-coded GAs. An illustration of such a problem could be the optimization of the times at which samples need to be taken in an experiment. If due to physical constraints, samples can only be taken every 0.1 minutes, then a sampling time of for example 0.437 minutes would be a redundant value while a sampling time of 0.4 minutes would be valid.

As already mentioned in Section 5.4.1.2, several solutions have been suggested to solve this problem. Firstly, one could assign very low fitness values to the redundant individuals or even discard them. Although applicable for binary coded GAs, this solution is not useful when using real-coded GAs because the chance of generating an offspring individual which is not redundant, from two continuous parent values, is extremely small. Another solution might be to split up the optimization problem in different encodings, resulting in chromosomes which are partially binary-coded and partially real-coded. This requires the use of different genetic crossover and mutation operators for each encoding. Although this has been done (Sarkar and Modak, 2003), its implementation is rather cumbersome. A final solution is to remap the redundant individuals to valid individuals. Using this approach, the entire optimization problem can be solved as a real-coded problem and only the crossover and mutation operators need to be extended slightly in such a way that illegal values for genes that need to be represented by discrete values are modified to the nearest valid value before the fitness of the offspring is evaluated. This method was applied by Ono et al. (1999) and proved to work adequately. Therefore, it was also used in EAST to solve the redundancy problems associated with discrete variables and also to handle the boundary constraints of the continuous optimization variables.

5.5 Designing measurement campaigns for SBRs

5.5.1 Introduction

This section will deal with designing measurement campaigns for SBRs with the aim of improving parameter estimation. Designing experiments using the procedure depicted in Figure 1.17 requires the availability of an initially calibrated model. For this case study, the two-step nitrification/denitrification SBR model described in Section 2.2.5 will be used. This model was calibrated using expert knowledge, based on data gathered during a measurement campaign performed on the BIOMATH lab-scale SBR on

29/11/04 (Sin et al., 2004b). The data of the measurement campaign and simulations of the calibrated model are shown in Figures 2.31 to 2.34.

Before proceeding with optimal experimental design for parameter estimation, the technique described in Section 4.4.2 will be applied in order to select the identifiable parameters. This will be done based on the already available model and the data of the measurement campaign. Once the identifiable parameters are available, the model can be recalibrated. This recalibrated model and identifiable parameter set will then serve as the basis for the design of a future measurement campaign. During this case study the FIM D-optimal criterion will be used as the optimal experimental design objective in order to reduce the volume of the confidence region of the parameter estimates. Note that the settling and draw phase of the SBR cycle will not be used in the design of experiments because too much model uncertainty still exists at this point.

5.5.2 Selection of the identifiable parameters

The technique to select identifiable parameters described in Section 4.4.2 is only applicable to a set of no more than 30 parameters (in order to prevent a combinatorial explosion). Since the two-step nitrification/denitrification SBR model is characterized by 88 biologically related model parameters (without including the initial conditions of the variables), a prior selection needs to be made. This selection was based on expert knowledge and on a parameter importance ranking (Equation 5.7). Following parameters were removed based on expert knowledge:

- Stoichiometric parameters: these are assumed to be well-known (Henze et al., 2000).
- Decay parameters: it does not seem meaningful to identify these “slow” parameters from a single SBR cycle.
- Influent composition parameters: these can be better estimated based on an influent characterisation experiment.
- Temperature correction factors: these are assumed to be well-known and cannot be estimated from an experiment with constant temperature (Hao et al., 2002).
- Alkalinity related parameters: no alkalinity measurements were performed and alkalinity inhibition/saturation is not considered in the model.

Note that excluding these parameters at this stage does not mean that the model output is not sensitive to these parameters but rather that the information present in the data will not be attributed to these parameters. This preliminary removal of parameters resulted in a subset of 34 parameters. In

order to further reduce the number of parameters, an importance ranking of the 34 remaining parameters was made. This ranking was based on the sensitivity measure δ_j^{msqr} for each parameter θ_j (Brun et al., 2001):

$$\delta_j^{msqr} = \sqrt{\frac{1}{n} \sum_{i=1}^n \frac{1}{N_i} \sum_{j=1}^{N_i} \left(\frac{\partial y_{ij}}{\partial \theta_k} \times \frac{\theta_k}{y_{ij}} \right)^2} \quad (5.7)$$

This measure is based on the relative sensitivity of the n variables to the j th parameter ($\partial y_{ij}/\partial \theta_j \times \theta_j/y_{ij}$) summed over all measurement points N_i . Relative sensitivities are used here in order to account for the different magnitudes of the parameters and variables. The sensitivity functions were sampled at the time instances corresponding to each measurement point of the measurement campaign. High values of δ_j^{msqr} indicate that the model simulations and thus the measurements are very sensitive to the parameter while low values indicate that the variables will not be influenced by changes in this parameter. The results of these calculations are shown in Table 5.3. Using this ranking, the 4 least sensitive parameters were excluded from the list.

Selecting an identifiable parameter subset requires the calculation of the FIM. Therefore, some assumptions on the error characteristics of the variables measured during the measurement campaign should be made. These are summarized in Table 5.4. The measurement error standard deviation for each variable was fixed after an analysis of the residuals calculated from the data of the measurement campaign of 29/01/04 and the expert calibrated model.

At this point, it is noteworthy to mention that dissolved oxygen measurements will not be considered in this case study even though they can be expected to be very informative. The reasoning behind this is that the oxygen concentration is controlled at a specific set-point during the aeration phases of the SBR operation. The dynamics of the oxygen concentration are thus largely determined by the control action of the oxygen controller. This controller is also modelled and its parameters are tuned in order to mimic the behaviour of the real oxygen control of the SBR. However, the simulated oxygen profile that is obtained in this way does not correspond to the real oxygen profile. This will obviously spoil any attempt to calibrate the model based on these data by only tuning biological parameters. Therefore, it is not advisable to incorporate oxygen as a measurement variable in the FIM. Since the oxygen controller implementation in the model is inadequate, it might have been a better idea to use the real control actions of the SBR directly as inputs for the model, and in this way, circumvent the need to model the oxygen controller in detail.

Once the variables and parameters were selected and set up correctly, the 30×30 FIM which is used as the basis for the selection of an identifiable parameter subset could be calculated. Following the procedure described in Section 4.4.2 an identifiable parameter subset of 8 parameters was found,

Table 5.3: Parameter importance ranking for the two-step nitrification/denitrification SBR model. The parameters forming the finally selected identifiable parameter subset are marked by an “X” in column 3. The expert knowledge based calibrated parameters of Sin et al. (2004b) and the identifiable parameter subset found by Brun et al. (2002) are also listed. The first 30 parameters are separated by a horizontal line from the remaining parameters.

Parameter	δ_j^{msqr}	Identifiability procedure	Sin et al. (2004b)	Brun et al. (2002)
μ_{PAO}	10.377	X	X	X
Q_{PP}	10.188		X	X
μ_{NH}	9.608		X	*
μ_{NO}	5.626		X	*
Q_{PHA}	5.269	X	X	X
K_h	5.269			X
μ_H	4.790	X		
η_{fe}	3.855			X
$\eta_{NO_3 HET}$	3.010	X		
$\eta_{NO_3 PAO}$	2.978			
K_O	2.789	X		
$K_{O PAO}$	2.597	X	X	
K_X	2.460	X		
$\eta_{NO_2 HET}$	2.229		X	
$K_{O NO}$	2.130		X	
$K_{NO_2 NO}$	1.845		X	
$K_{O NH}$	1.780	X	X	
K_{NO_3}	1.758		X	
K_{PS}	1.747			
Q_{fe}	1.662			
K_{MAX}	1.506			
K_{NO_2}	1.494			
$\eta_{NO HYD}$	1.302			
K_P	1.037			
K_F	1.023			
K_{SA}	0.993			
K_{IPP}	0.902			
K_{PHA}	0.805			
K_{fe}	0.598			
$K_{NH NH}$	0.554		X	
K_{PP}	0.401			
K_{NH}	0.130			
$K_{NH NO}$	0.074		X	
K_a	0.035			

*: maximum autotrophic growth rate for a one-step nitrification model

Table 5.4: Error characteristics of the measurements performed on the SBR system.

Name	Description	Unit	Standard deviation	Error type
S_{NH_4}	Ammonium	$mg.l^{-1}$	0.4	absolute
S_{NO_3}	Nitrate	$mg.l^{-1}$	0.1	absolute
S_{NO_2}	Nitrite	$mg.l^{-1}$	0.1	absolute
S_{PO_4}	Phosphate	$mg.l^{-1}$	0.5	absolute

which are all listed in the upper half of the sensitivity importance ranking. Notice that some of the highest sensitive parameters were not selected due to correlations with other parameter. The parameters of this subset are marked by an “X” in column 3 of Table 5.3. Starting from the most sensitive parameter, they are: the maximum growth rate of PAOs (μ_{PAO}), the rate constant for PHA storage (Q_{PHA}), the maximum growth rate of the heterotrophic biomass (μ_H), the reduction factor for nitrate denitrification ($\eta_{NO_3\ HET}$), the oxygen saturation/inhibition constant for the heterotrophic biomass (K_O), the oxygen saturation/inhibition constant for PAOs ($K_{O\ PAO}$), the saturation/inhibition constant for particulate COD (K_X) and the oxygen saturation/inhibition constant for ammonium oxidizers ($K_{O\ NH}$).

Comparing this list with the results of the expert knowledge calibration performed by Sin et al. (2004b) (also shown in Table 5.3) reveals that 4 parameters (μ_{PAO} , Q_{PHA} , $K_{O\ PAO}$ and $K_{O\ NH}$) present in the automatically selected list were also used in the expert knowledge calibration. Also, notice that many of the most sensitive parameters were used in the calibration, indicating good expert knowledge about the parameter sensitivities. An analysis of the identifiable subsets found by Brun et al. (2002), who calibrated an ASM2d model for a conventional wastewater treatment plant, also showed some similarities (see Table 5.3). For example, the parameters Q_{PHA} and μ_{PAO} were also determined to be identifiable. Other highly ranked parameters were also included in their identifiable subset. However, it has to be mentioned that this model did not include two-step nitrification and denitrification. Therefore, the maximum autotrophic growth rate, which was also reported to be identifiable by Brun et al. (2002), is indicated by an “*” symbol for both types of nitrifiers in the two-step nitrification and denitrification model.

Based on the selected identifiable parameter subset, it was attempted to recalibrate the model using trial and error, starting from the calibrated parameter values of Sin et al. (2004b). However, despite considerable efforts, no improvement could be obtained compared to the fit of Sin et al. (2004b). This is most likely due to the fact that most parameters from the identifiable subset were already used during the expert knowledge calibration. It was therefore decided to use the expert knowledge based calibrated model together with the set of selected identifiable parameters as a starting point for the experimental design study.

5.5.3 Optimization of measurement times

5.5.3.1 Problem description

Probably the most important decision in designing a measurement campaign for an SBR process is the choice of measurement times in order to get data which can be optimally used for parameter estimation. Since several phases can be distinguished in an SBR cycle, the question can then be raised at which points in time measurements of a certain measurement variable should be performed. Should each measurement variable be measured in all phases, or only in certain phases? Answering this question will thus involve an inner loop optimization of the FIM calculations (see Figure 5.2). This involves performing a sensitivity analysis using the default operating conditions for the SBR model. Based on this sensitivity analysis an optimization is performed by sub-sampling the sensitivity function values at different locations in time. The algorithm which can be used to optimize this problem will depend on the definition of the experimental degrees of freedom (in this case the measurement times). In the sequel, several situations will be considered where the measurement points will be expressed either as continuous or discrete optimization variables.

5.5.3.2 Measurement points expressed as continuous variables

Description of the experimental degrees of freedom

In a first attempt to design a useful measurement campaign for the SBR process, the measurement points are defined as continuous variables. It will be assumed that measurements of ammonium, nitrite, nitrate and phosphate can be made. For each of these measurement variables, 10 distinct measurements will be performed. This problem thus involves an optimization of 40 experimental degrees of freedom.

The error characteristics for the available measurement variables will be taken as described in Table 5.4. By using these error characteristics, the measurement error covariance matrix (Q) can be calculated. The inverse of this matrix will be used as weights to scale the contributions of the different sensitivities of the different measurement variables to the final FIM.

Algorithm settings

The integration algorithm used to solve the model equations was the CVODE stiff solver with absolute and relative accuracy set to $1E-10$, using the Adam-Moulton linear multi-step method and functional iterative solver.

Sensitivity analysis was performed using central differences with a perturbation factor of 0.005 for all parameters except μ_{PAO} for which a perturbation factor of 0.001 was used.

Since continuous variables will be used for this optimization problem, the performance of several optimization algorithms can be compared: Simplex,

Praxis and a real-coded GA. Default settings were used for Simplex (step size 1, accuracy $1E-5$) and Praxis (accuracy $1E-5$). The GA settings are shown in Table 5.5 and are chosen partially based on experience and partially based on “default” settings from literature. The initial population of individuals was initialized at random.

Table 5.5: GA settings used for the SBR measurement time optimization problem.

Attribute	Value
Population size	400
Replacement %	10%
Selection scheme	Roulette wheel
Scaling	Linear (multiplier: 1.2)
Crossover probability	0.8
Crossover operator	BLX- α ($\alpha = 0.5$)
Mutation probability	0.01
Mutation operator	Non-uniform

During the optimization, no restrictions were imposed on the maximum number of allowed FIM evaluations. In order to halt the optimization algorithms, the stopping criterion accuracy was set to $1E-5$ for each algorithm. If the relative difference between the objective value of two particular solutions is below this value, the optimization is stopped. However, depending on the algorithm, the two solutions involved in the comparison will be different. For the GA, these two solutions are the best and the worst individual in the population. For Simplex, the worst and the best simplex point are used and for Praxis the best and previous best solution found.

In order to evaluate the convergence of the optimization algorithms, 10 repeated optimization runs were performed for each algorithm. For each of these runs, the initial values of the variables (measurement times) were initialized at random. For Simplex and Praxis an extra optimization run with uniform initialisation (initial measurement times equally spaced) was also performed.

Results and discussion

A summary of the results of the measurement time optimization using continuous variables is shown in Table 5.6. In this table the FIM D-optimal criterion value of the best solution from 10 optimization runs is shown for each optimization algorithm together with the number of required FIM evaluations. Beside the best solutions, the average and standard deviation for the D-optimal criterion and the number of FIM evaluations is also shown. For Simplex and Praxis the results of the optimization using a uniform initialisation of the initial measurement times are also given.

Before analysing the different “optimal” experimental designs, it is worthwhile to investigate the performance of the different optimization algorithms.

Table 5.6: Results of the SBR continuous measurement time optimization problem.

	D-optimal criterion			Number of FIM evaluations		
	Simplex	Praxis	GA	Simplex	Praxis	GA
Best	$3.00E+26$	$4.13E+27$	$3.59E+28$	6872	10586	145435
Average	$7.94E+25$	$1.16E+27$	$3.28E+28$	7042	4932	131461
Std. Dev.	$9.63E+25$	$6.51E+26$	$2.12E+27$	1791	1284	19020
Uniform	$1.11E+25$	$4.13E+27$		11992	10586	

From Table 5.6 it is clear that the classical optimization algorithms (Simplex and Praxis) converged to lower D-optimal criterion values as compared to the GA. This illustrates the ability of the GA to better locate the global optimum. However, it can also be seen (from the standard deviation) that the GA did not always converge to the same optimum. This can probably be improved by better tuning the GA through expert knowledge or trial and error. The analysis of the results also shows that the value of the standard deviation of the D-optimal criterion compared to the value of the average D-optimal criterion value for Simplex is much higher than for the other algorithms. This indicates that Simplex converged to several local optima which were far from each other (large standard deviation). Praxis and the GA converged to solutions which had more similar D-optimal criterion values. Comparing the performance of the classical algorithms, it can be seen that Simplex is outperformed by Praxis, which reached a best objective value being a factor 10 higher. Uniform initialization of the measurement times did not prove to be useful for Simplex. However, the uniform initialization led to the best solution for the Praxis optimizer.

The highest values of the D-optimal criteria were $3.00E+26$ and $3.59E+28$ for Simplex and the GA respectively. This corresponded, for this specific case, to covariance matrix diagonal elements which were 3 times lower (for the GA) for some parameters, resulting in an approximate decrease of the confidence interval on the parameter estimates by a factor 1.7. This is a significant improvement.

The better performance of the GA however has a drawback. Typically, the GA required approximately 100 times more FIM evaluations compared to the classical algorithms. However, one should remember that for this (inner loop) optimization problem only one sensitivity analysis needs to be performed. Therefore, FIM evaluations are relatively “cheap” and an increased number of evaluations is certainly justified since a better optimum can be located.

After this discussion of the D-optimal criterion values, the optimal measurement times themselves need to be analysed. This is done in Figure 5.8 which is divided into 4 sub-figures, each representing one measurement variable (phosphate, ammonium, nitrite and nitrate). Each sub-figure is divided into 3 parts. The top section shows the simulation results of the

measurement variable as function of time. The different phases of the SBR operation are indicated by vertical lines. The middle part of the graph shows the sensitivity functions of the measured variable to each of the identifiable parameters. At the bottom of the graph, the optimal measurement times determined using different optimization algorithms are shown. Measurement schemes 1, 2, 3 and 4 correspond to the best solutions found by using Simplex, Praxis, the GA using continuous measurement times and the GA using discrete measurement times. A discussion on the results of using discrete measurement times can be found in the next section.

Before discussing the actual optimal measurement times, it is interesting to analyse the shapes of all sensitivity functions. Most of these functions show distinct different shapes which confirms that the parameters considered in the experimental design study are little correlated. This indicates that the identifiable parameter selection procedure, executed prior to the optimal experimental design, was able to select a useful parameter subset.

From the optimal measurement times found by the Simplex optimizer, it can clearly be seen that the optimization resulted in a large quantity of non-informative measurement times. Measurement points are placed at times which have a very low corresponding sensitivity. These measurement times will therefore contribute little or nothing to the final FIM. The Praxis algorithm on the other hand performs significantly better. A clear tendency is observed of putting the optimal measurement times at places where sensitivities are high. This is clearly observable for all measured variables. When using the GA, it can be seen that a lot of measurement points are placed at the same locations. For ammonium measurements, for instance, only 2 distinct times are visible. All 10 possible measurements are distributed among these two measurement times. This illustrates that repetitions of measurements are useful to decrease the parameter estimation errors. It also illustrates the capability of the GA to better locate the global minimum compared to Simplex and Praxis.

5.5.3.3 Measurement points expressed as discrete variables

Description of the experimental degrees of freedom

From the optimization using measurement times characterized by continuous variables, it was observed that many optimal measurement times were found to be repetitions at the same point. Indeed, independent samples taken at the same time would contain much information. Statistically this may be correct, but some practical issues have to be considered as well. Concentrating all measurements in one place is only safe if there is much confidence that the model simulations are representative for the system behaviour. If the system would behave differently, the information content of the real measured data might not live up to these high expectations. This is an important issue for optimal experimental design and will be discussed further in Chapter 8. Besides, samples taken during a measurement campaign typically have a large enough volume to allow plenty of repetitions.

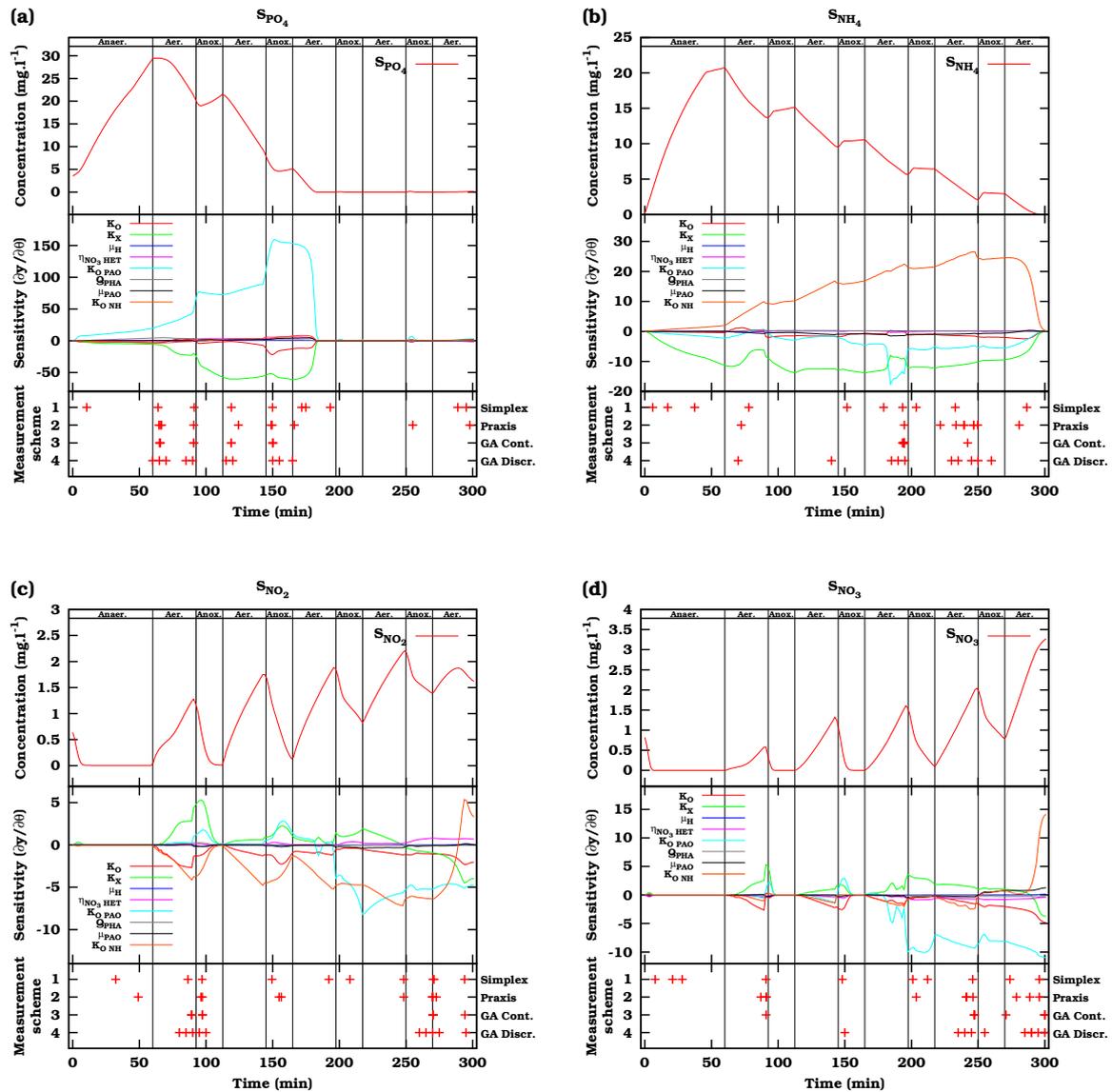


Figure 5.8: Optimal measurement times for measurements of phosphorous (a), ammonium (b), nitrite (c) and nitrate (d). Each sub-figure is divided into 3 parts. The top part shows the simulation results, the middle part the sensitivity functions of the measurement variable to the identifiable parameters and the bottom part the optimal measurement times for different designs. The different phases of the SBR operation are indicated by vertical lines.

Whether these will be performed will depend on the actual cost associated with a single measurement. Because of the above-mentioned arguments, the optimization will be performed again but this time the measurement times will be described by discrete variables which are required to have a minimum distance of 5 minutes between them and no replications are allowed. This interval between the measurements is a practical requirement since the samples need to be membrane filtered to stop further reactions from taking place. Since continuous variables are no longer considered, only the GA can be used to solve this design problem. In order to prevent measurement repetitions, the crossover and mutation operators had to be extended. This was done by making sure that each GA gene (representing the measurement points) of an offspring which is formed and coincides with an already occupied measurement time is assigned the closed available valid value.

Algorithm settings

For this optimization problem, identical GA settings were used as for the optimization using continuous variables (Table 5.5). Also, the initial population of individuals was initialized at random. The convergence behaviour of the GA was also studied by performing 10 repeated optimization runs.

Results and discussion

The objective values and the required number of FIM evaluations of the optimization of measurement times described by discrete variables are listed in Table 5.7. Comparing the results of Table 5.6 and Table 5.7 reveals that the magnitude of the D-optimal criterion value is somewhat lower in the discrete case. This is caused by the fact that repetitions at the optimal measurement locations cannot fully be replaced by measurements in the same neighborhood. However, this type of design has more practical value. It can also be seen that the GA required considerably less FIM evaluations in the discrete case which is caused by the discretization of the search domain. Analyzing the average values and standard deviations of the D-optimal criterion and the number of evaluations, it can be seen that the average is relatively close to the best solution and that the standard deviation is small. This indicates that the GA converged to solutions which were relatively close to each other.

Table 5.7: Results of the SBR discrete measurement time optimization problem.

	D-criterion	Number of FIM evaluations
Best	$7.59E+27$	61705
Average	$6.71E+27$	50152
Std. Dev.	$6.07E+26$	10361

Looking at the measurement times themselves (measurement scheme 4 in Figure 5.8). A trend can be observed which is similar to the continuous case.

Measurement times are placed at points with high sensitivities. However, since no repetitions were allowed, measurement points were more spread out over the neighborhood in which the optimal points were found in the continuous case. It can also be seen that when all possible measurement points in the neighborhood of the optimal point in the continuous case are “occupied”, the other measurements are shifted to the “second best” location. This is clearly illustrated for ammonium measurements where samples should primarily be taken in the third and fourth aerobic phase.

An analysis of the measurement times reveals that not all measurement variables should be measured throughout the entire cycle. Phosphate measurements should primarily be measured during the first and second aeration and anoxic phases. Ammonium sensitivity to the parameters increases towards the end of the cycle, favoring measurements in the third and fourth aeration and anoxic phases. For nitrite, two distinct measurement regions can be discerned: (1) the first aeration and anoxic phase and (2) the last aerobic and anoxic phase. Nitrate measurements should primarily be concentrated in the fourth aeration and anoxic phase and the last aeration phase.

5.5.4 Optimization of experimental manipulations

Problem description

Beside optimizing measurement schemes, experimental manipulations can also be used to increase the information content of the acquired data. Changing experimental manipulations influences the dynamics of the system under study and thus has implications for the sensitivities of the measured variables to the parameters. In this part of the case study, measurement times will be assumed fixed and only experimental manipulations will be optimized. This corresponds to an outer loop optimization problem. During the optimization, every objective function evaluation requires a sensitivity analysis to be performed based on model simulations using certain values for the experimental manipulations. The sensitivity values at fixed measurement times are then used to calculate the FIM. This type of experimental design optimization is more computer intensive than an inner loop optimization since for each proposed solution, a new sensitivity analysis needs to be performed.

Description of the experimental degrees of freedom

Two experimental degrees of freedom related to the SBR operational conditions were chosen for optimization: the set-point which controls the oxygen concentration in the first 4 aeration phases ($S_{O,set}$) and the influent feed volume to be added to the anoxic phases (V_{anox}). Both experimental conditions are described by continuous variables constrained between a lower and upper bound. The default values for these two operational conditions are listed in Table 5.8 together with the lower and upper bounds that were

used during the optimization. The oxygen set-point was chosen between 0.2 and 2 $mg.l^{-1}$. The lower bound of the total feed volume to be added to the anoxic phases equals 0. This corresponds to the situation where all the influent (33 l) is added at the beginning of the cycle during the anaerobic phase and no influent is added during the anoxic phases. The upper bound corresponds to the situation where all the feed is added to the anoxic phases (33/4 l added in each anoxic phase) and nothing at the beginning of the cycle.

Table 5.8: Default operational conditions, lower and upper bound for the oxygen set-point in aeration phases 1 to 4 ($S_{O,set}$) and the total feed volume to be added to the anoxic phases (V_{anox}).

	Unit	Default value	Lower bound	Upper bound
$S_{O,set}$	$mg.l^{-1}$	0.5	0.2	2
V_{anox}	l	7.9	0	33

This optimization problem assumes fixed measurement times. Two designs using different amounts of samples were thereby considered. A total of 10 or 30 samples, for each measurement variable, were uniformly distributed over the SBR cycle (300 min), excluding the settling and draw phases, resulting in one measurement every 30 or 10 minutes respectively. A more in-depth discussion on the number of samples and the associated costs is given in the next chapter.

Algorithm settings

The integration algorithm used to solve the model equations was the CVODE stiff solver with absolute and relative accuracy set to $1E-10$, using the Adam-Moulton linear multi-step method and functional iterative solver.

Sensitivity analysis was performed using central differences with a perturbation factor of 0.005 for all parameters except μ_{PAO} for which a perturbation factor of 0.001 was used.

The outer loop optimization was solved using a real-coded GA with settings listed in Table 5.9. These settings correspond to the settings of the measurement time optimization problem except for the population size. This GA setting is taken smaller because only two optimization variables are considered in this case. Also, the initial population of individuals was initialized at random. As was also done for the measurement time optimization, 10 repeated optimization runs were executed in order to evaluate the algorithm convergence.

Table 5.9: GA settings used for the SBR experimental manipulation optimization problem.

Attribute	Value
Population size	100
Replacement %	10%
Selection scheme	Roulette wheel
Scaling	Linear (multiplier: 1.2)
Crossover probability	0.8
Crossover operator	BLX- α ($\alpha = 0.5$)
Mutation probability	0.01
Mutation operator	Non-uniform

Results and discussion

Tables 5.10 and 5.11 list the results of the optimization. These tables show the best, average and standard deviation of the oxygen set-point, the anoxic feed volume, the FIM D-optimal criterion and the required number of FIM evaluations when 30 or 10 fixed measurement points are used respectively.

Table 5.10: Results of the SBR manipulation optimization problem using 30 fixed measurements for each measurement variable.

	$S_{O,set}$ ($mg.l^{-1}$)	V_{anox} (l)	D-criterion	Number of FIM evaluations
Best	1.55	13.15	$3.51E+29$	1332
Average	1.44	13.73	$2.59E+29$	1236
Std. Dev.	0.16	1.94	$5.66E+28$	431

Table 5.11: Results of the SBR manipulation optimization problem using 10 fixed measurements for each measurement variable.

	$S_{O,set}$ ($mg.l^{-1}$)	V_{anox} (l)	D-criterion	Number of FIM evaluations
Best	0.92	4.01	$1.06E+26$	994
Average	0.90	4.19	$1.03E+26$	1319
Std. Dev.	0.02	0.27	$3.22E+24$	447

Comparing both tables, it can be seen that the highest D-criterion value reached, differs significantly. This was expected since in the first case 120 samples are taken compared to only 40 in the second case. It can also be seen that the optimal values for the experimental manipulations differ significantly. This illustrates the dependency of the design on the measurement schedule. From the standard deviation on the D-criterion values it can be concluded that the GA did not always converge to the exact same solution. However, if the average and standard deviations of the optimization variables are investigated, it can be seen that almost identical experimental

conditions were found. The number of FIM evaluations can also be regarded as relatively consistent. This number is much smaller than was observed for the measurement time optimization since only 2 variables required optimization in this case. However, this amount still represents a very large computational burden, since each FIM evaluation (taking approximately one minute on a PIII 1 GHz system) required one complete sensitivity analysis, each requiring 16 model simulations for a central difference sensitivity analysis.

For the situation where 30 samples are taken for each measurement variable, the optimal values for the experimental degrees of freedom were found to be 1.55 mg.l^{-1} and 13.15 l for the oxygen set-point and for the anoxic feed volume respectively. In the situation where only 10 samples were taken for each measurement variable, the experimental degrees of freedom were 0.92 mg.l^{-1} and 4.01 l . From these results it can be concluded that a higher oxygen set-point (compared to the default operating conditions) results in an increased information content. However, for the feed volume of the anoxic phases no general conclusion can be given.

How the settings of the experimental manipulations influence the system in practice is illustrated in Figure 5.9 which represents the best solution found during the optimization. This figure is organized in a similar way as Figure 5.8 except that at the bottom of the graph, the measurement times of the 30 uniformly distributed measurement points are shown.

Comparing the model simulations for the optimal operating conditions with those under default operating conditions, it can be seen that phosphate is consumed much faster in the optimal case. Phosphate is already completely removed in the second aeration phase, while it is only completely removed in the third aeration phase for the default operating conditions. For ammonium no significant difference can be detected in the concentration profile. For nitrite and nitrate it can be seen that the decrease in nitrite concentration (compared to the default operating conditions, Figure 5.8) at the end of the cycle results in an increased nitrate concentration.

The changes in system behaviour also have implications on the sensitivity functions. Some of the sensitivity functions have increased in absolute value, leading to a higher expected information content of the data. For phosphate this is noticeable in the second aeration phase and the first anoxic phase, but also in the last anoxic phase. The increased sensitivity of ammonium is most pronounced in the last anoxic and aerobic phase, especially for parameters $K_{O\text{ PAO}}$, $K_{O\text{ NH}}$ and K_X . For nitrite and nitrate, similar conclusions hold.

5.5.5 Combined optimization of experimental manipulations and measurement times

Description of the experimental degrees of freedom

As a final illustration, a combination of manipulation and measurement ex-

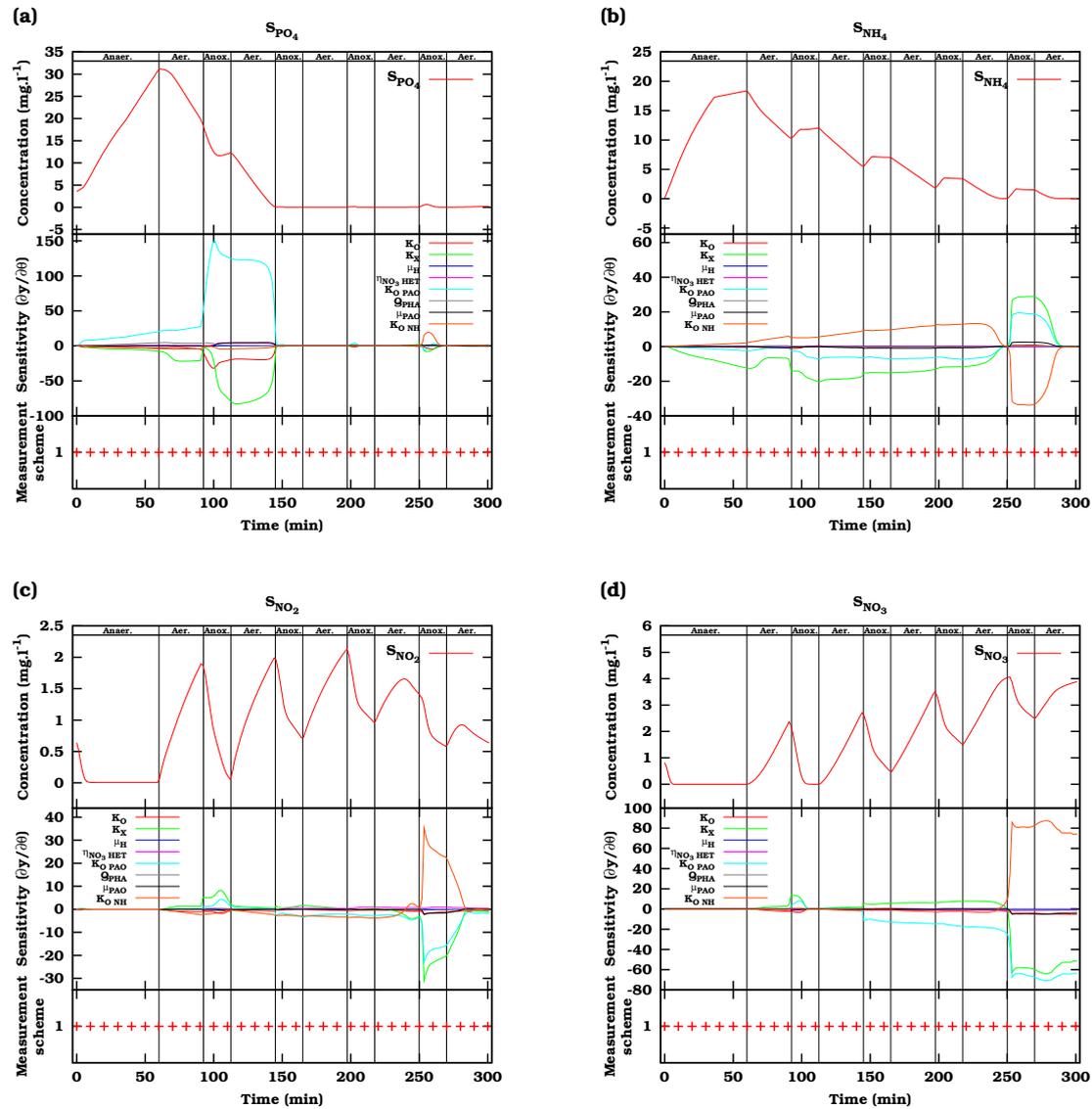


Figure 5.9: Graphical representation of the optimal experimental manipulation design for phosphorous (a), ammonium (b), nitrite (c) and nitrate (d). Each sub-figure is divided into 3 parts. The top part shows the simulation results as a function of time. The middle part shows the sensitivity functions of the measurement variable to the identifiable parameters. The bottom part shows the location of the 30 fixed measurement times for each measurement variable. The different phases of the SBR operation are indicated by vertical lines.

perimental degrees of freedom will be optimized. As described before, two experimental degrees of freedom related to the SBR operational conditions were chosen for optimization: the set-point which controls the oxygen concentration in the first 4 aeration phases ($S_{O,set}$) and the influent feed volume to be added to the anoxic phases (V_{anox}). For each of the proposed operational conditions, an optimization was performed of 10 measurement times (represented by discrete optimization variables) for each measurement variable. This optimization problem is thus a combined outer/inner loop optimization problem and requires a significant amount of computations. For each proposed set of operational conditions, one complete sensitivity analysis should be performed (outer loop) which is then used to optimize the measurement times (inner loop).

The manipulation and measurement experimental degrees of freedom were encoded in an identical way as described above. The reader is referred to Section 5.5.4 and 5.5.3.3 for details on the manipulation and measurement experimental degrees of freedom respectively.

Algorithm settings

The optimization of the inner and outer loop is performed using (nested) real-coded genetic algorithms, each having different settings. The GA settings for the inner loop GA can be found in Table 5.5, the settings for the outer loop GA can be found in Table 5.9. For each of the algorithms, the individuals of the initial population were initialized at random.

Results and discussion

The optimization of the combined manipulation and measurement experimental degrees of freedom problem converged after approximately 2.5 days of calculations on a 3 GHz Pentium 4 system. A total number of 873 sensitivity analyses (outer loop optimization) were performed. For each sensitivity analysis, on average 45000 FIM evaluations were needed to optimize the measurement times (inner loop optimization) resulting in a total of about 40 million FIM evaluations for the combined optimization problem.

The optimal values for the experimental manipulations ($S_{O,set}$ and V_{anox}) are listed in Table 5.12 together with the D-optimal criterion value of the optimal experiment. The optimal measurement scheme associated with these experimental manipulations is shown in Figure 5.10.

Table 5.12: Results of the SBR combined manipulation and measurement optimization problem using 10 (optimized) measurements points for each measurement variable.

$S_{O,set}$ ($mg.l^{-1}$)	V_{anox} (l)	D-criterion
1.26	0.14	$9.92E+28$

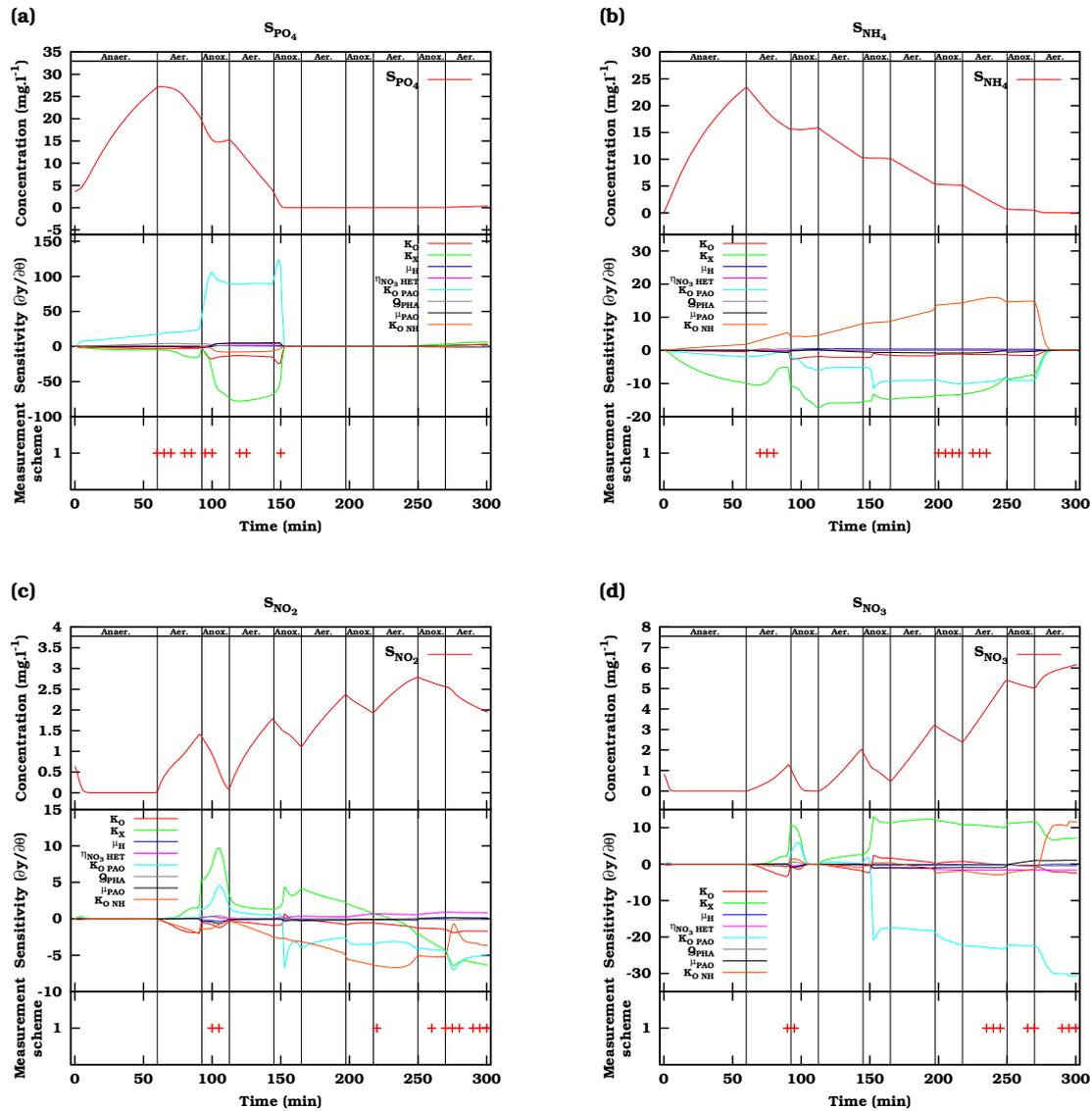


Figure 5.10: Graphical representation of the combined optimal experimental manipulation and measurement design for phosphorous (a), ammonium (b), nitrite (c) and nitrate (d). Each sub-figure is divided into 3 parts. The top part shows the simulation results. The middle part shows the sensitivity functions of the measurement variable to the identifiable parameters. The bottom part shows the location of the 10 measurement times for each measurement variable. The different phases of the SBR operation are indicated by vertical lines.

An analysis of Table 5.12 reveals that the oxygen set-point in the SBR should be 1.26 mg.l^{-1} and the feed volume to be added to the anoxic phases should be 0.14 l in order to get data with an optimal information content (related to the D-optimal design criterion). For the feed volume, it can be concluded that this amount is negligible, indicating that the entire feed volume should preferably be added to the first SBR phase, the filling or anaerobic phase. Comparing these results with Tables 5.10 and 5.11, which list the results of the experimental manipulation optimization using fixed measurement times, again shows that the optimal experimental manipulation values largely depend on the chosen measurement layout. Also, notice that the design using 30 fixed measurement times (for each measurement variable) only yields a slightly higher information content compared to the design with 10 optimized measurement points (for each measurement variable). This illustrates the benefits of optimizing the measurement times in combination with the experimental manipulations.

The added benefit of optimizing the experimental manipulations can be derived by comparing Tables 5.7 and 5.12. As already mentioned, Table 5.12 presents the results of the combined optimization of manipulations and measurements, whereas Table 5.7 only represents results of an optimization of measurement experimental degrees of freedom, identical to the measurement experimental degrees of freedom of the combined case. This comparison shows that an approximately 10 times higher D-optimal criterion value is reached by considering the combined optimization which corresponds, in this case, to a 30 % decrease of the parameter estimation error.

For the optimal measurement scheme (Figure 5.10) associated with the optimal experimental manipulation values listed in Table 5.12 it can be concluded that the measurement points are located in regions of high sensitivity. Similar conclusions were found for the measurement point optimizations described in Section 5.5.3.

5.5.6 Practical significance of the optimal experimental designs

An important aspect of optimal experimental design which was not discussed so far is the practical meaning of the D-optimal criterion values. In order to evaluate whether the optimal experiment that was found would be useful in reality, it is important to have an idea about the parameter estimation errors that would be achieved when a model would be calibrated based on data gathered from the optimal experiment. Making use of the relationship between the FIM, the Hessian matrix and the parameter estimation error covariance matrix (Section 4.3), a predicted parameter estimation error covariance matrix can be calculated using Equation 5.8.

$$C = \sigma^2 FIM^{-1} \quad (5.8)$$

In this equation σ^2 is an approximation of the residual variance calculated from the known measurement error characteristics, the number of measurement points and parameters. According to Equation 1.17, the parameter estimation error ($e_{\hat{\theta}}$) is calculated based on the ratio between the parameter estimation error standard deviation and the parameter value. The predicted parameter estimation errors that would result from the optimal experiments described by Tables 5.6 (Simplex, Praxis, GA Cont), 5.7 (GA Disc), 5.10 (Manip 10), 5.11 (Manip 30) and 5.12 (Combined) are listed in Table 5.13. In this table, the D-optimal criterion for each of the designs is also shown.

Table 5.13: Predicted parameter estimation errors that would result from a model calibration based on data of a measurement campaign performed according to the results listed in Tables 5.6 (Simplex, Praxis, GA Cont), 5.7 (GA Disc), 5.10 (Manip 10), 5.11 (Manip 30) and 5.12 (Combined). For each of the designs, the D-optimal criterion value is also given.

	Estimation errors ($e_{\hat{\theta}}$) (%)						
	Simplex	Praxis	GA Cont	GA Disc	Manip 30	Manip 10	Combined
μ_{PAO}	8.41	9.01	7.46	8.97	4.52	8.85	4.38
Q_{PHA}	5.44	3.41	3.00	3.26	1.80	5.60	2.91
μ_H	7.70	15.81	12.47	17.30	19.51	21.89	7.70
$\eta_{NO_3 HET}$	18.74	20.10	17.00	18.61	5.34	20.97	10.63
K_O	17.14	16.89	13.49	16.32	9.58	19.21	12.27
$K_{O PAO}$	16.39	15.19	10.73	12.73	6.22	16.33	8.48
K_X	23.99	21.86	17.84	20.18	6.20	27.15	15.46
$K_{O NH}$	17.14	13.95	11.36	13.63	11.05	19.42	9.63
D-criterion	3.00E+26	4.13E+27	3.59E+28	7.59E+27	3.51E+29	1.06E+26	9.92E+28

Analysing the results reveals that the estimation errors are correlated with the D-optimal criterion values (as was expected). Comparing the best (Manip 30) and the worst design (Simplex) shows that the predicted estimation errors have been halved, a significant improvement. It can also be seen that the design combining the optimization of manipulations and measurements has estimation errors which are similar to those of the Manip 30 design. However, the first requires only one third of the measurements. This clearly shows the advantage of the optimization of both manipulations and measurement locations.

It should be noted that Table 5.13 only provides information about the parameter estimation confidence intervals and does not reflect the correlations between the parameters. However, correlations between the parameters are expected to be fairly low since the identifiable parameter subset selection was based on a procedure which eliminates strongly correlated parameter combinations.

Important to mention is also that the experimental design performed in this

chapter assumes that the model structure is correct and that the model parameters which were used as the basis for the design are relatively close to the “real” system. If this is the case, then the proposed experimental design should deliver data with the expected information content. If uncertainty exists concerning the model structure, optimal experimental design for model selection could be applied (Chen and Asprey, 2003). If, on the other hand, the assumed parameter values are uncertain, robust experimental design might be required (see Chapter 8).

5.6 Conclusions

This chapter illustrated the complexity of the optimal experimental design optimization problem. It was suggested to split up the optimization problem into two nested optimization loops in order to reduce the amount of required calculations and improve the convergence properties of the optimizer used. This solution proved to be very useful when applying genetic algorithms to solve the complex optimal experimental design optimization problem. It was illustrated that a real-coded genetic algorithm could be used to solve problems of various natures, even combinations of continuous and discrete optimization variables.

For the illustrated case, the GA outperformed classical optimization algorithms by being able to locate the global optimum in a consistent way. However, the GA did not always converge to the exact same optimum. This problem might be solved by better tuning the GA settings and operators. Using adaptive methods to control these GA parameters during the course of the optimization would be a valuable addition to the performed research. Preparing for the practical illustration, it was discovered that dissolved oxygen measurements could not be used in the experimental design since the model implementation of the oxygen controller proved inadequate to describe the real oxygen dynamics in the reactor. This measurement was therefore omitted from the experimental design calculations.

For the SBR measurement campaign design it could be concluded that the measurement point optimization is very useful for gaining data with a high information content. It was also found that values of the optimal experimental manipulations are largely influenced by the imposed measurement scheme. Different optimal operating conditions were found for measurement schemes of 10 or 30 uniformly distributed measurement points over the entire SBR cycle. Finally, the combined optimization of measurement and manipulation experimental degrees of freedom proved to be most valuable. However, substantial computations were required to solve this problem.

If a combined optimization is not feasible, optimization of measurements should be preferred over manipulation optimizations with uniformly distributed measurements since the latter provides less information due to the fixed nature of the measurement points. It also has to be mentioned that

changing the operational conditions (experimental manipulations) of a system might change its behaviour in such a way that the used model and the optimal design is no longer representative for this system.

Chapter 6

Multi-objective optimal experimental design

6.1 Introduction

Finding a useful experimental design often involves taking into account several, sometimes conflicting, objectives. These objectives can for example be the FIM optimal criteria which represent different properties of the FIM and thus also different properties of the parameter estimation confidence region. In Section 1.3.3, it was already illustrated that the optima for the D-optimal and modified E-optimal FIM criteria corresponded with different optimal experimental designs. The choice of which criterion to use is mostly determined by the individual preferences of the researcher concerning certain properties of the confidence region of the parameter estimates. Beside the FIM criteria, other factors, like experimental cost and system performance, can also be considered as useful experimental design objectives. Finding experimental designs which are optimal with respect to several criteria can be accomplished by performing a multi-objective optimization of the experimental degrees of freedom. In this chapter, a brief overview of multi-objective optimization and its terminology will be given, followed by an application of a multi-objective genetic algorithm for the design of sequencing batch reactor (SBR) measurement campaigns.

6.2 Multi-objective optimization

As the name suggests, multi-objective optimization deals with several objectives which need to be minimized and/or maximized. This is accomplished

by choosing the correct values of several decision variables. In the context of optimal experimental design for parameter estimation, these decision variables are the experimental degrees of freedom. Typically, the experimental degrees of freedom are bounded by certain limits and constraints and all valid values for the experimental degrees of freedom can be described by the decision variable space. For each point of this decision space, a corresponding value for each of the objectives can be calculated. These values then describe the objective space. The shape of the objective space is controlled by the nature of the objectives and the bounds and constraints of the decision variables. This concept is illustrated in Figure 6.1 for a problem consisting of two experimental degrees of freedom (Ψ_1, Ψ_2) and two objectives (J_1, J_2) which need to be minimized.

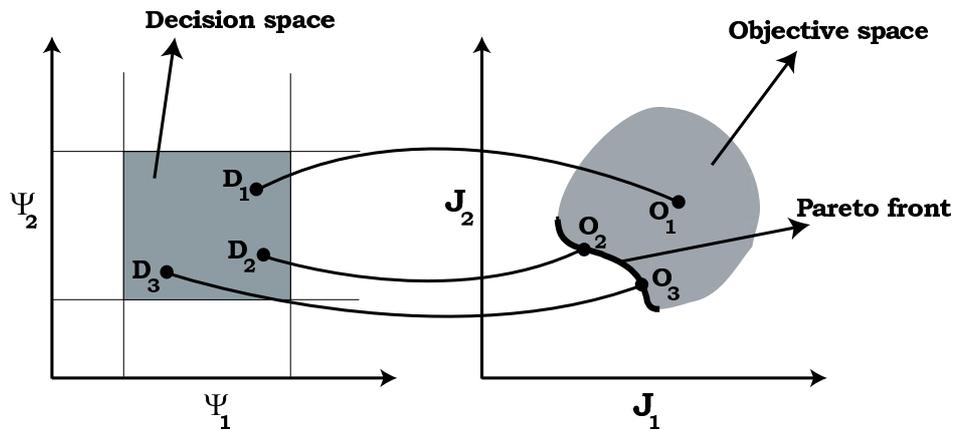


Figure 6.1: Schematic overview of the mapping of the decision space onto the objective space for a problem consisting of two experimental degrees of freedom (Ψ_1, Ψ_2) and two objectives (J_1, J_2) which need to be minimized. The Pareto front is indicated by a bold line.

This figure clearly illustrates the mapping of several points of the decision space (D_1, D_2 and D_3) onto points of the objective space (O_1, O_2 and O_3). Comparing O_1 to O_2 and O_3 shows that the values of each of the objectives of O_2 and O_3 are lower than the objective values of O_1 . Therefore, O_1 is said to be dominated by O_2 and O_3 . Similarly, O_2 and O_3 are said to dominate O_1 since each of their objective values is lower than those of O_1 . If O_2 and O_3 are compared, it is clear that O_2 has a higher value for J_2 but a lower value of J_1 . Neither O_2 or O_3 are thus dominated by each other or O_1 and therefore these points are called non-dominated. Because both objectives are important, it cannot be concluded which of the two solutions is better. Since the concept of domination allows to compare solutions with multiple objectives, most multi-objective optimization methods use this domination concept to search for non-dominated solutions. For the entire feasible objective space, a set of solutions of non-dominated points exists which are called Pareto-optimal. Each of the points in this set is optimal in the sense that no improvement can be achieved in one objective (by selecting another set of decision variable values) without resulting in a degradation of at least

one of the remaining objectives. This set of solutions, often also called the Pareto front, is indicated by a bold line in Figure 6.1.

6.3 Classical methods for solving the multi-objective optimization problem

Probably the simplest and most widely used classical method to tackle the multi-objective optimization problem is the weighted sum method. This method scalarizes a set of objectives into a single objective by multiplying each objective with a user-supplied weight. Although the basis for this method is simple and intuitive, it does introduce a not-so-simple question. What values of the weights must be used in order to prevent one objective dominating the other(s)? Unfortunately, there is no unique answer to this question. The answer depends on the importance of each objective in the context of the problem and on the magnitudes of the objectives. This implies that we should know, to a certain extent, the behaviour of each of the objectives, which is a very expensive process (computationally speaking). Using this weighted sum method, points of the Pareto front can be determined by solving several optimizations for which the multiple objectives are combined into a single objective each time using different weights. In other words, for each combined-objective optimization, a different relative contribution of the objectives to the combined objective value is considered. However, sets of uniformly distributed weights do not guarantee that a uniformly distributed set of Pareto solutions will be found. Since this mapping is unknown it is difficult to set the weight vectors in order to obtain Pareto-optimal solutions in the desired region of the objective space.

Beside the weighted sum method, other classical methods exist to solve the multi-objective optimization problem. The most important are: the ϵ -constraint method, the weighted metric methods, Benson's method, the value function method and goal programming methods. An overview of these methods is given by Deb (2001) who also discusses the advantages and disadvantages of each method.

Related to optimal experimental design for parameter estimation only a limited number of applications of the weighted sum method have been found in literature (Versyck and Van Impe, 2000a; Berkholz et al., 2000; Atkinson and Bogacka, 2002).

Versyck and Van Impe (2000a) proposed a criterion in which the FIM D-optimal and the modified E-optimal criteria were combined in order to design an optimal feed rate for a fed-batch reactor. Hereby, each criterion was logarithmically scaled in order to reduce the difference in magnitude between the criteria and was weighted by a factor β ranging from 0 to 1. The entire range of weights was investigated and expert knowledge and practical considerations were used to select the final optimal value for β .

The research performed by Berkholz et al. (2000) was focused on estimation

of kinetic parameters in fermentation processes. The inverse of the modified E-optimal criterion was combined with a criterion describing the process performance of the reactor (with respect to the amount of end-product formed). Both objectives were scaled using the maximal objective value found during a single-objective optimization of each criterion separately. Next, a combined criterion was formed by weighting the scaled objectives by a factor ω ranging between 0 and 1. Several experiments were designed using arbitrarily chosen weight factors in order to put more or less emphasis on parameter estimation or on process performance.

Finally, Atkinson and Bogacka (2002) constructed compound design criteria in order to be able to make a choice between estimating reaction rates or the reaction order of chemical combustion. Again, the different objectives were logarithmically scaled and weighted by a factor α between 0 and 1.

Classical ways of handling multi-objective optimization problems are known to have some drawbacks. Firstly, they require the choice of certain multi-objective tuning factors like weights, ϵ vectors, target vectors and various others. Secondly, each of these methods requires many combined-objective optimization runs each resulting in one Pareto-optimal solution. A possible solution for these problems is the use of genetic algorithms since these methods have the useful characteristic of processing populations of solutions rather than single solutions.

6.4 Multi-objective genetic algorithms

6.4.1 Introduction

In 1985 the first application of a GA to a multi-objective optimization problem was published by Shaffer (1985). Since then, multi-objective GAs have, and still are, a topic of intensive investigation in the evolutionary algorithms research community. This has resulted in several reviews and comparison papers of the existing algorithms, among others: Fonseca and Fleming (1995), Zitzler et al. (2000), Coello (2000), Van Veldhuizen and Lamont (2000), Deb (2001) and Tan et al. (2002). Although different classification schemes have been suggested, one way to classify the algorithms is evaluating whether Pareto-dominance is used to select individuals from the population or not. Apparently, most research has been focused on the Pareto-based methods.

6.4.2 Basic requirements for Pareto-based multi-objective GAs

Two major problems must be addressed before a genetic algorithm can be successfully applied to a multi-objective optimization problem. Firstly, how should fitness be assigned and how should the selection of the individuals

be guided towards the Pareto-optimal set. Secondly, how should population diversity be maintained in order to prevent premature convergence and to find well distributed solutions along the Pareto front.

A solution to the first problem was proposed by Goldberg (1989). It consisted of finding the non-dominated individuals of the population and assigning them the highest rank. A collection of individuals with equal rank is also called a non-dominated front. From the remaining individuals, the next set of non-dominated individuals is determined and assigned the next highest rank. This process is continued until the entire population is ranked. This assures that the individuals which are the most dominating have the highest reproductive chances. Several modifications of this general idea have been proposed and implemented by different researchers. In Section 6.4.3 an implementation of this strategy will be illustrated for the multi-objective optimization algorithm NSGA-II.

The second problem can be solved by using the concept of niching or sharing. This concept was originally developed for single-objective optimization in order to find and maintain multiple optima (i.e. multi-modal problems). This means that in addition to finding the global optimum, there is also an interest to find a number of locally optimal solutions. The concept of sharing was introduced by Goldberg and Richardson (1987) and was based on degrading the fitness of similar solutions in order to maintain diversity in the population. Assume a population of N individuals and q optima to be located. Since the population will need representatives for each optimum, somehow the available slots (individuals) need to be shared equally among the different optima. If in a certain generation, an optimum has more than the expected number of representatives, this will come at the cost of another optimum. Thus, the fitness of each of these representatives needs to be degraded in such a way that in the overall competition not all of these representatives become chosen for reproduction. On the other hand, the fitness of representatives of an under-represented optimum must be emphasized in order to gain a higher probability to produce offspring. Practically, this idea can be implemented in several ways. An example of such an implementation will be given in Section 6.4.3 for the multi-objective optimization algorithm NSGA-II.

6.4.3 NSGA-II

The multi-objective optimization algorithm NSGA-II developed by Deb (2001) was implemented in EAST and will be used in this chapter. This choice was based on literature review and its widespread successful application in many research domains. This algorithm evolved from the Non-dominated Sorting Genetic Algorithm (NSGA) developed by Srinivas and Deb (1994) but few similarities remained in NSGA-II. A schematic representation of the procedure of the NSGA-II algorithm is shown in Figure 6.2.

The working principle of the NSGA-II algorithm can be broken down into several steps:

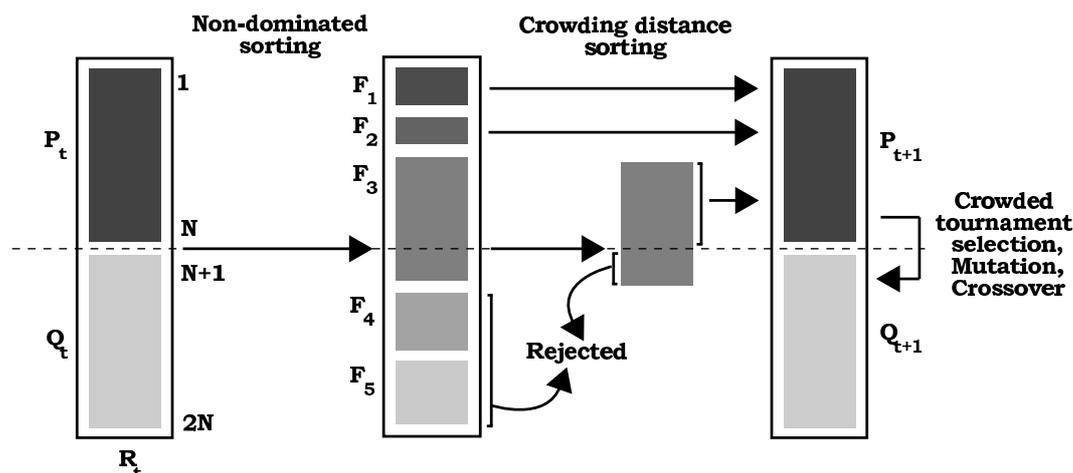


Figure 6.2: Schematic procedure of the NSGA-II multi-objective optimization algorithm (adapted from Deb, 2001).

Step 1:

A parent (P_t) and offspring population (Q_t) each containing N individuals are combined to form one big population of $2N$ individuals (R_t). Then, non-dominated sorting is applied to R_t as proposed by Goldberg (1989). This procedure results in a subdivision of R_t in several non-dominated fronts ($F_1 \dots F_5$ in Figure 6.2) where each individual is assigned a fitness value equal to its non-domination level (1 being the best level assuming minimization of the objectives). Although the sorting of the population of size $2N$ requires more calculations than sorting on Q_t alone, it does allow for a global non-domination check among the parents and the offspring.

Step 2:

Once the sorting is completed, the new parent population P_{t+1} is filled using individuals from different fronts starting from the best non-dominated front. At a certain point during the filling, a front will be encountered that cannot be completely accommodated in the new parent population (F_3 in Figure 6.2). Instead of arbitrarily discarding some members of this front, it would be wise to use some kind of sharing strategy to choose those members which reside in the least crowded region of that front. This procedure ensures the selection of a diverse set of individuals from this front. To get an estimate of the density of the solutions surrounding a particular solution i , the average distance to the two solutions on either side of solution i along each objective is calculated, also called the crowding distance. Large crowding distances indicate isolated solutions which are preferred in order to preserve population diversity and uniform distribution of the solutions along the Pareto front. Based on these calculations, the remaining empty slots of the new parent population are filled with individuals having large crowding distances.

Step 3:

In a final step, a new offspring population Q_{t+1} is formed using a crowded tournament selection operator, crossover and mutation. The selection operator compares two solutions and returns the winner of the tournament. Each solution is associated with two attributes: a non-domination rank in the population and a crowding distance. A solution i wins from another solution j if the rank of solution i is better than that of solution j . If both solutions have the same rank, the solution which has a larger crowding distance wins the tournament.

Step 4:

Based on P_{t+1} and Q_{t+1} , steps 1 to 3 can be repeated in order to evolve the algorithm until a user-defined number of generations is reached and the algorithm execution is halted.

6.5 Designing SBR measurement campaigns using multi-objective GAs

6.5.1 Problem description

The optimal experimental design performed in Section 5.5 for a SBR system focused on a single objective: the FIM D-optimal criterion. In this section, multiple objectives will be considered for the design of measurement campaigns for the same SBR system, modelled by the two-step nitrification/denitrification model described in Section 2.2.5. As a starting point for the experimental design the expert knowledge calibrated model and identifiable parameter subset (8 parameters) determined in Section 5.5.2 will be used. The measurements (ammonium, nitrate, nitrite and phosphate) for which a measurement scheme will be designed, are listed in Table 6.1. This table also lists the standard deviations of the measurements (measurement errors) which were determined from an analysis of the residuals calculated from data gathered from the measurement campaign of 29/01/04 and the expert calibrated model. These error characteristics are used to compose the measurement error covariance matrix required for the FIM calculations. Finally, the table also shows, for each measurement variable, the costs associated with a single measurement performed using a Dr. Lange cuvette test.

In contrast to part of the measurement campaign design in Chapter 5, no optimization of experimental manipulations will be considered here: default operating conditions are assumed.

Table 6.1: Measurement variables, error characteristics and measurement costs to be used in the design of a SBR measurement campaign.

Name	Description	Unit	Standard deviation	Error type	Cost of a single measurement
S_{NH_4}	Ammonium	$mg.l^{-1}$	0.4	absolute	3 EUR
S_{NO_3}	Nitrate	$mg.l^{-1}$	0.1	absolute	3 EUR
S_{NO_2}	Nitrite	$mg.l^{-1}$	0.1	absolute	2 EUR
S_{PO_4}	Phosphate	$mg.l^{-1}$	0.5	absolute	3 EUR

6.5.2 Algorithm settings

The following paragraphs describe the settings of the different algorithms used while performing optimal experimental design in this chapter.

The two-step nitrification/denitrification model was solved using the CVODE stiff solver with absolute and relative accuracy set to $1E-10$, using the Adam-Moulton linear multi-step method and functional iterative solver.

Sensitivity analysis was performed using central differences with a perturbation factor of 0.005 for all parameters except μ_{PAO} for which a perturbation factor of 0.001 was used.

The multi-objective optimizations were performed using NSGA-II described in Section 6.4.3. The GA population was chosen to be 100 individuals. For a multi-objective optimization this number determines the number of individuals forming the Pareto-optimal front and thus equals the number of non-dominated solutions which will be presented to the user to make a final decision on what experiment to perform. The NSGA-II algorithm was stopped after 1000 GA generations and the final population was taken as the Pareto-optimal front. Other relevant GA settings are listed in Table 6.2.

Table 6.2: GA settings used for the multi-objective SBR measurement campaign design.

Attribute	Value
Population size	100
Selection scheme	Crowded tournament
Scaling	Linear (multiplier: 1.2)
Crossover probability	0.8
Crossover operator	BLX- α ($\alpha = 0.5$)
Mutation probability	0.01
Mutation operator	Non-uniform

6.5.3 Multi-objective D-optimal and modified E-optimal design

6.5.3.1 Objectives

The two objectives considered in this section are the FIM D-optimal and modified E-optimal criteria. These “conflicting” criteria are related to different properties of the parameter estimation confidence regions. Minimizing the volume of this confidence region can be achieved by maximizing the D-optimal criterion (determinant of the FIM). On the other hand, minimizing the modified E-optimal criterion results in a more circular shape of the confidence region, reducing as such the parameter correlations. However, since this criterion is only related to the shape of the confidence region and not to the volume, it is possible to obtain circular confidence regions with very large volumes. A graphical illustration of these FIM criteria was already shown in Figures 1.18 and 1.19.

6.5.3.2 Experimental degrees of freedom

Similar to the optimal experimental design performed in Section 5.5.3, the location of 10 measurements for each measurement variable will be optimized. These 40 measurement locations are described by discrete variables with a minimum of 5 minutes between two consecutive measurements to allow sufficient time for sample filtering. Also, no repetitions of measurements are allowed.

6.5.3.3 Results and discussion

An illustration of the evolution of the individuals of the GA population during the course of the multi-objective optimization is given by Figure 6.3. This figure shows the D-optimal and modified E-optimal values of the individuals of several GA populations. The population is shown at generations 0, 50, 100, 150 and 1000, being the final generation. From these data it is clear that the initial population contains mostly non-optimal individuals in the sense that the D-optimal criterion values are relatively low and the modified E-optimal criterion values are reasonably high. It can also be seen that with increasing generations, the solutions evolve towards higher D-optimal values and lower modified E-optimal values. Between generations 150 and 1000 not much improvement is made, indicating that the final population (generation 1000) can be regarded as a good approximation of the Pareto front. Notice that the coverage of low D-optimal and modified E-optimal values seems poor. However, this is due to the logarithmic scaling of the axes.

The Pareto front is enlarged in Figure 6.4. Analysis of the front shows that individuals for a wide range of D-optimal and modified E-optimal values are present, indicating that the NSGA-II algorithm was able to spread out the

individuals adequately over the entire front. From the front it becomes clear that low values (optimal) of the modified E-optimal criterion are associated with low values of the D-optimal criterion (non-optimal) and vice versa. This illustrates the conflicting nature of both objectives.

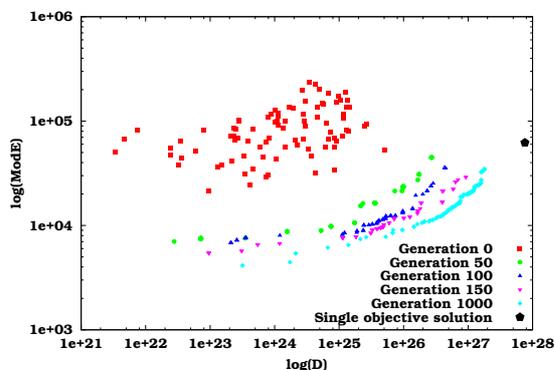


Figure 6.3: D-optimal and modified E-optimal values for individuals of different GA generations for a multi-objective optimization of measurement times in a SBR measurement campaign.

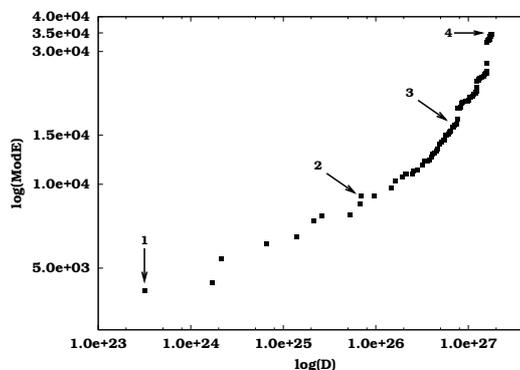


Figure 6.4: D-optimal and modified E-optimal values for individuals of the Pareto-optimal front of a multi-objective optimization of measurement times in a SBR measurement campaign. Several selected solutions indicated by arrows are discussed in the text.

Each of the solutions in the front represents a non-dominated solution of the optimization problem. At this point, the experimenter should choose the optimal experiment by considering a trade-off between the objectives. Four individual solutions of the front will be analysed more closely. These are indicated by arrows 1 to 4 in Figure 6.4. Individual 1 represents the solution with the lowest modified E-optimal value. The solution with the highest D-optimal value is represented by individual 4. Individuals 2 and 3 represent intermediate solutions on the Pareto-optimal front. Each of the selected individuals in Figure 6.4 represents a measurement scheme of 40 measurement points (10 for each measurement variable). These measurement schemes are shown in Figure 6.5 which is composed of 4 sub-figures, each representing one measurement variable (phosphate, ammonium, nitrite and nitrate). Each sub-figure is divided into 3 parts. The top part shows the simulation results of the measurement variable as a function of time. The different phases of the SBR operation are indicated by vertical lines. The middle part of the graph shows the sensitivity functions of the measurement variable to each of the considered parameters. The bottom part shows the optimal measurement schemes for the selected individuals of the Pareto front.

For individual 4, the measurements are mostly placed at points which correspond with high sensitivities. A more or less identical measurement

scheme was found for the single-objective optimization of the measurement times (Section 5.5.3 and Figure 5.8 measurement scheme 4) although in the multi-objective case a lower D-optimal value was reached: $1.80E+27$ (multi-objective) compared to $7.59E+27$ (single-objective). This difference is probably caused by the fact that the multi-objective optimization does not primarily focus on finding the extreme solutions but rather focuses on finding the Pareto-optimal front. The solution of the single-objective optimization is also shown in Figure 6.3 as the pentagon symbol ($D=7.59E+27$, modified $E=6.21E+4$). As expected, it is situated on a line following the curvature of the Pareto front.

In contrast to individual 4, the measurement scheme associated with individual 1 (modified E-optimal value equal to $4.14E+03$) is more spread out over the entire SBR cycle. This type of measurement scheme is more difficult to interpret visually when many variables and parameters are considered. Modified E-optimal designs place measurements at locations where certain variables are sensitive to certain parameters and insensitive to other parameters. In this way, the data collected at these points only contains information about some parameters. This helps to reduce parameter correlations when identifying a model based on these data. The measurement schemes for individuals 2 and 3 are intermediate when compared to those of the extreme solutions although they show more characteristics of individual 4 than of individual 1.

In order to better understand the impact of the different measurement schemes on the parameter estimates it is useful to calculate the predicted parameter estimation error standard deviation. As was also done in Chapter 5, the relationship between the FIM, the Hessian matrix and the parameter estimation error covariance matrix (Section 4.3) can be used to calculate a predicted parameter estimation error covariance matrix:

$$C = \sigma^2 FIM^{-1} \quad (6.1)$$

In this equation σ^2 is an approximation of the residual variance calculated from the known measurement error characteristics, the number of measurement points and parameters. According to Equation 1.17, a parameter estimation error ($e_{\hat{\theta}}$) can be calculated based on the ratio between the parameter estimation error standard deviation and the parameter value. The predicted parameter estimation errors that would result from the measurement campaigns associated with the 4 selected individuals of the Pareto front are listed in Table 6.3. This table also shows the D-optimal and modified E-optimal criterion values for the different measurement schemes. Analysing the data clearly shows that the parameter estimation errors decrease with increasing D-optimal criterion values.

Based on the parameter estimation error covariance matrix, a correlation matrix which can be used to interpret the predicted correlations between the parameters can also be constructed using Equation 1.15. This information is shown in Table 6.4 which lists the predicted (absolute) parameter

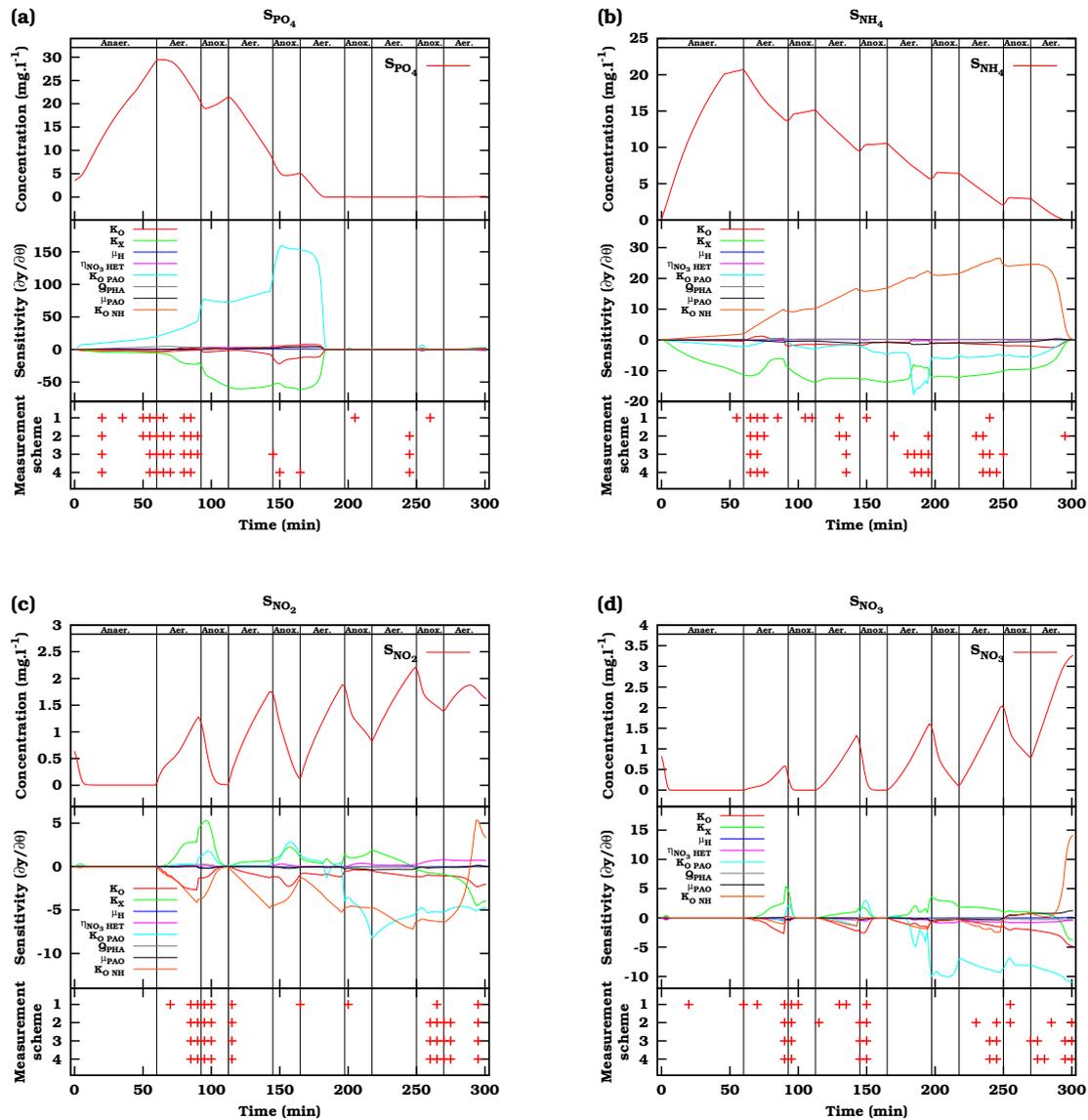


Figure 6.5: Optimal measurement times for measurements of phosphorous (a), ammonium (b), nitrite (c) and nitrate (d). Each sub-figure is divided into 3 parts. The top part shows the simulation results, the middle part the sensitivity functions of the measurement variable to the identifiable parameters and the bottom part the measurement times for different points of the Pareto front indicated in Figure 6.4. The different phases of the SBR operation are indicated by vertical lines.

Table 6.3: Predicted parameter estimation errors that would result from the measurement campaigns given by 4 selected individuals or measurement schemes (indicated in Figure 6.4) of the Pareto front for the multi-objective D-optimal and modified E-optimal design.

	Estimation error ($e_{\hat{\theta}}$) (%)			
	Scheme 1	Scheme 2	Scheme 3	Scheme 4
μ_{PAO}	19.72	8.82	6.43	6.43
Q_{PHA}	6.41	4.43	2.82	2.59
μ_H	14.46	12.93	11.37	11.61
$\eta_{NO_3 HET}$	26.71	17.69	14.33	14.38
K_O	26.45	20.48	13.03	11.68
$K_{O PAO}$	62.67	33.10	15.35	11.37
K_X	30.65	26.30	17.88	18.52
$K_{O NH}$	26.45	17.35	12.33	12.37
D	$3.18E+23$	$6.89E+25$	$7.58E+26$	$1.80E+27$
ModE	$4.14E+03$	$9.05E+03$	$1.72E+04$	$3.46E+04$

correlation characteristics that would result from the measurement campaigns given by the 4 selected individuals (measurement schemes) of the Pareto front. For each measurement scheme the maximum (absolute) correlation found between two parameters is shown together with the average (absolute) correlation and its standard deviation. Surprisingly, the maximum correlation decreases with an increase in modified E-optimal criterion whereas theoretically the opposite would be expected. However, the average correlation between all parameters remains more or less equal. This is probably a result of the identifiable parameter subset selection which focused on selecting parameters with low mutual correlations. Due to this situation, the experimental design optimization is unable to find a design which can further de-correlate the parameters.

Table 6.4: Predicted (absolute) parameter correlation characteristics that would result from the measurement campaigns given by 4 selected individuals or measurement schemes (indicated in Figure 6.4) of the Pareto front for the multi-objective D-optimal and modified E-optimal design.

Scheme	Absolute correlation			D	ModE
	Maximum	Average	Standard deviation		
1	0.85	0.28	0.20	$3.18E+23$	$4.14E+03$
2	0.79	0.29	0.23	$6.89E+25$	$9.05E+03$
3	0.58	0.23	0.17	$7.58E+26$	$1.72E+04$
4	0.59	0.22	0.16	$1.80E+27$	$3.46E+04$

6.5.4 Experimental cost versus D-optimal design

6.5.4.1 Objectives

In contrast to the appreciation in practice, research studies focusing on optimal experimental design often neglect experimental cost. Gathering large amounts of data often requires expensive measurement campaigns. Therefore, it would be useful to incorporate this objective directly into the design. Multi-objective optimization allows the experimenter to get a clear idea about the trade-off between experimental cost and information content. In this case, information content will be expressed by the D-optimal FIM criterion (to be maximized) whereas experimental cost (to be minimized) will be calculated based on the number of measurements for each measurement variable and their associated costs, listed in Table 6.1.

6.5.4.2 Experimental degrees of freedom

For this illustration, a measurement interval (between measurements) was optimized for each measurement variable separately. These intervals were described by discrete variables and were allowed to vary between 5 and 300 minutes with discrete steps of 1 minute. A measurement interval of 5 minutes would result in 61 measurement points uniformly distributed over the entire SBR cycle (300 minutes, excluding settling and draw phase). Setting the measurement interval to the upper bound, 300 minutes, would result in only two measurement points: one at time 0 and one at time 300.

6.5.4.3 Results and discussion

The multi-objective optimization of the measurement intervals using NSGA-II resulted in a Pareto front shown in Figure 6.6. The shape of the Pareto front illustrates the conflicting nature of both objectives. Large D-optimal criterion values are associated with large cost, while less expensive experiments result in lower information content. Although this is an intuitively logical conclusion, the availability of the Pareto front allows the experimenter to make an intelligent choice in which he can find a suitable trade-off between information content and experimental cost.

In order to gain more insight in the experiments associated with the different individuals of the Pareto-optimal front, 3 individuals will be studied in more detail. These individuals are indicated by arrows in Figure 6.6. For each of the selected individuals, the optimal measurement intervals for the measurement variables are listed in Table 6.5. In this table, the number of uniformly distributed measurements, over the entire SBR cycle, is indicated between square brackets. The D-optimal criterion values and the total experimental costs are also shown. From these results it is clear that the experiment with the highest information content is also the most expensive experiment (individual 3) and requires that each variable is measured every

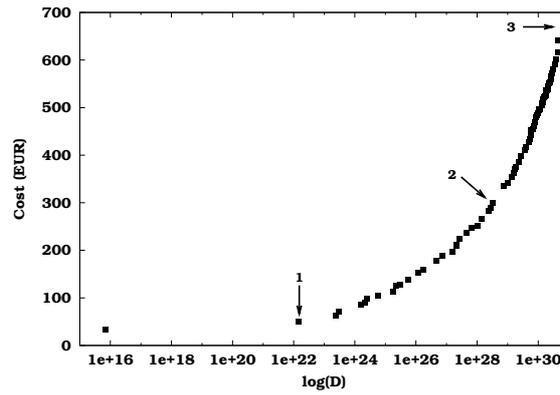


Figure 6.6: D-optimal and experimental cost values for individuals of the Pareto-optimal front of a multi-objective optimization of measurement intervals in a SBR measurement campaign. Several selected solutions, indicated by arrows, are discussed in the text.

5 minutes (the lower bound) resulting in a total of 244 measurements. In contrast to that, the least expensive experiment requires only 19 measurements but also has the lowest information content. The experiment which corresponds to individual 2 of the Pareto front has an associated cost of 300 EUR and results in a D-optimal criterion value of $3.27E+28$ and requires 117 measurements to be taken. Also notice that this experiment allows more nitrite (NO_2) measurements to be performed since the cost associated with these measurements is lower compared to that of the other measurement variables. Another benefit of this design is that the increased number of nitrite measurements will also provide much process insight since nitrite is an important intermediate in the nitrification process.

Table 6.5: Optimal measurement interval and number of measurements for 3 selected individuals of the Pareto front shown in Figure 6.6. The D-optimal criterion value and total experimental cost are also shown for each measurement scheme.

Scheme	Interval (min) [# measurements]				D	Cost (EUR)
	NH_4	NO_2	NO_3	PO_4		
1	205 [2]	59 [6]	92 [4]	44 [7]	$1.43E+22$	51
2	87 [4]	6 [51]	13 [24]	8 [38]	$3.27E+28$	300
3	5 [61]	5 [61]	5 [61]	5 [61]	$6.33E+30$	671

Before an experimenter/modeller can make a sound choice on which experiment to perform it should first be quantified how the information content (FIM D-optimal criterion value) relates to the parameter estimation errors. As described above, a predicted parameter estimation error can be calculated based on the inverse of the FIM using Equation 6.1. These errors are listed in Table 6.6 for each of the selected individuals of the Pareto front (measurement schemes) together with their associated D-optimal criterion value and experimental cost. Clearly, measurement scheme 1, which has

the lowest cost and D-optimal criterion value, would result in parameter estimates with too large errors. On the other hand, measurement scheme 3 has the lowest errors but also the highest associated cost: 13 times more expensive than measurement scheme 1. However, the parameter estimation errors have not decreased with a similar factor. Therefore, measurement scheme 2 might be a better choice since its parameter estimation errors are higher, but still very acceptable, compared to those of measurement scheme 3 and this for only half the experimental cost.

Table 6.6: Predicted parameter estimation errors that would result from the measurement campaigns given by 3 selected individuals or sampling schemes (indicated in Figure 6.6) of the Pareto front for the multi-objective optimization of the D-optimal design and the experimental cost.

	Estimation error ($e_{\hat{\theta}}$) (%)		
	Scheme 1	Scheme 2	Scheme 3
μ_{PAO}	23.89	7.43	3.82
Q_{PHA}	9.83	2.50	1.95
μ_H	30.96	9.93	8.71
$\eta_{NO_3 HET}$	47.88	11.02	8.12
K_O	49.89	10.55	7.34
$K_{O PAO}$	39.34	9.86	7.61
K_X	49.33	13.20	8.92
$K_{O NH}$	29.23	9.72	6.60
D	$1.43E+22$	$3.27E+28$	$6.33E+30$
Cost (EUR)	51	300	671

It should also be noted that the optimal experiment which was found at the end of Chapter 5, after a combined optimization of manipulation and measurement experimental degrees of freedom resulted in a D-criterion value of $9.92E+28$. This result was found using only 10 measurements for each measurement variable with an associated cost of 110 EUR. In that case however, an optimization of the timing of the 10 measurements was performed whereas the case described in this section focused on uniformly distributed measurement points. A combined optimization of the number of required measurements for each measurement variable and the timing of these measurements during the experiment would require a GA with individuals of variable size, something which is not yet supported in the multi-objective GA implementation of EAST.

6.6 Conclusions

This chapter illustrated the use of multi-objective genetic algorithms in order to solve optimal experimental design problems when different objectives

are considered. In contrast to other multi-objective techniques, these algorithms do not require weights or other steering parameters that are needed to turn the multi-objective problem into a single-objective problem. Through the use of multi-objective GAs, the front of Pareto-optimal solutions can be determined. This front can then be used to allow the researcher to make a sound choice with respect to which experiment will be performed based on a trade-off between the different objectives.

An illustration was given for the design of a SBR measurement campaign in which several objectives were considered: the D-optimal and modified E-optimal design criterion and experimental cost. A real-coded implementation of the NSGA-II genetic algorithm was successfully used to determine the Pareto front for combinations of these objectives. The knowledge about the Pareto front could then be used to select a practically usable measurement scheme. Although the illustration in this chapter was rather simple, the method could also be used to solve more complex design problems including model prediction uncertainty and process performance.

Chapter 7

Automatic optimal experimental design

7.1 Introduction

Optimal experimental design based on FIM properties has been applied numerous times using the iterative procedure shown in Figure 1.17. The first step of this procedure is the calibration of the model based on initial data. Next, new experiments are proposed, simulated and evaluated until the optimal experiment is found. After the optimal experiment is performed in reality, the model is recalibrated and the procedure repeated if required. The main reason why the above-mentioned procedure is still not very often used is that it involves different, rather complicated mathematical and practical steps in order to obtain the calibrated model. This chapter will therefore focus on the development of a methodology which automates the entire procedure, making expert interventions almost unnecessary.

7.2 An automatic optimal experimental design methodology

Figure 7.1 illustrates the proposed automated optimal experimental design procedure. In order to apply this procedure in practice an extension was programmed to EAST. This extension makes use of existing EAST modules like simulation, optimization, sensitivity analysis and optimal experimental design and is described in Section 2.1.2.7.

Two main parts can be distinguished in the procedure. The left side of Figure 7.1 represents reality where experiments are conducted and data collected. The right side of the figure represents the computer system where

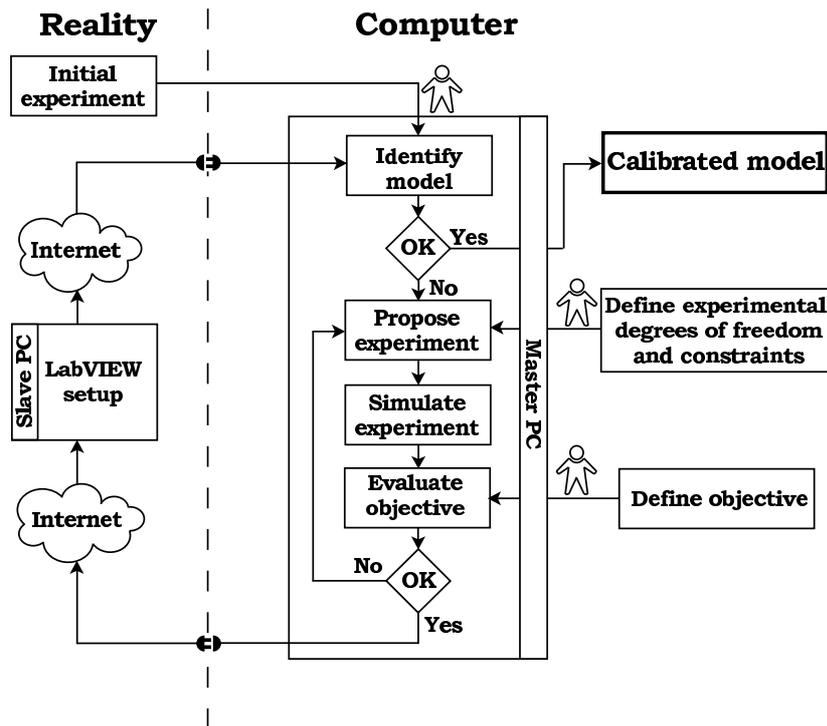


Figure 7.1: Schematic view of the automatic optimal experimental design methodology. Places where user interaction is still required are indicated by human symbols (adapted from Dochain and Vanrolleghem, 2001).

models are calibrated, experiments are simulated and their potential information content evaluated. User interaction is only required once, at three points in the proposed procedure; these are indicated by human symbols in Figure 7.1.

The starting point of the experimental design procedure is the identification of the model using data of an initial experiment. At this point user interaction is still very important as automation seems as yet unfeasible. First, the user has to select an appropriate model structure. Next, the identifiability of the proposed model should be studied in order to find the parameter combinations which can be uniquely estimated based on the available data. This can be done by any of the methods described in Section 4.4. Parameters for which the output sensitivities are very small or highly correlated should not be estimated but fixed at a default value. Once the parameter subset is determined, the model can be fitted to the data of the initial experiment. User interaction is also required to specify the experimental degrees of freedom and constraints. These can be classified into two types: measurements and manipulations (see Chapter 5). The user needs to specify which variables can be measured and which measurement frequencies and measurement locations in time or space are allowed, possibly within certain constraints. Also, the measurement error characteristics of each measurement need to be specified. Experimental manipulations concern all actions that can be taken to manipulate the experimental setup in some way. These

include the initial conditions and experimental inputs like imposed flow rate, temperature or concentration profiles. The choice of the experimental degrees of freedom has important implications on the choice of optimizer that has to be used to solve the design problem. For continuous optimization problems classical optimization algorithms can be used. However, if experimental degrees of freedom have been defined which can only take certain discrete values, other algorithms which can cope with this kind of restrictions have to be used. Ideal candidates for this are genetic algorithms (Goldberg, 1989; Herrera et al., 1998) because of their ability to optimize combined continuous and discrete problems. The use of genetic algorithms to solve optimal experimental design problems was already discussed in Chapters 5 and 6.

The final decision the user has to make is to define the objective that will be used to evaluate the proposed experiments. These objectives can be any of the FIM properties that are listed in Table 1.4.

Once the experimental degrees of freedom/constraints and the objective are specified the experimental design procedure can be started. Based on the initially calibrated model, new experiments are proposed using the specified experimental degrees of freedom. These experiments are simulated and the objective evaluated. The optimization of the experimental degrees of freedom is continued until an optimal experiment is found, optimal in the sense that it minimizes or maximizes the objective, depending on the choice of the objective. If the optimal experiment has been found it can be performed in reality. The EAST automatic optimal experimental design module is able to transfer the values for the experimental degrees of freedom of the optimal experiment through TCP-sockets over the Internet (using Qt's network capabilities). In this way it can communicate with every software package that is equipped with this form of communication interface. In this work LabVIEW (National Instruments, USA) is used to perform the experiment and collect the data. Once the optimal values of the experimental degrees of freedom have been received by LabVIEW, the experiment is performed automatically. After the experiment is finalized LabVIEW transfers the acquired data through the same network interface to the EAST automatic experimental design module. Once the data is received, the model is recalibrated using the parameter values of the already available model as initial guesses. It is then decided whether the procedure needs to be repeated or not. Depending on the chosen objective criterion, this decision can be made on the basis of the accuracy of the parameter estimates or the correlations among them. In this work, the proposed procedure is stopped when the largest correlation is lower than 0.5 if the modified E-optimal criterion is used, or when the largest 95 % confidence interval is smaller than $\pm 20\%$ of the parameter value if another FIM criterion is used. The procedure is also stopped when the correlations or confidence intervals for the latest experiment show no further improvements compared to the previous one.

7.3 Application to a hybrid respirometer

7.3.1 Experimental setup

In order to illustrate the automatic experimental design procedure, respirometric experiments were performed using an integrated sensor for monitoring aerobic and anoxic activated sludge activities (Sin et al., 2003). Only a short description of the sensor will be given here since the focus of this chapter is more on the experimental design procedure. Full details about the experimental setup can be found in the above-mentioned paper.

The setup of the hybrid respirometer shown in Figures 7.2 and 7.3 consists of an aeration (2.5 l) and a respiration (1.0 l) vessel. A cooling system is used to control the temperature of the reactors. Data acquisition, pH-control and data processing are implemented in LabVIEW (LabVIEW 6.1i, National Instruments, USA). In the aeration vessel dissolved oxygen, nitrate and pH are measured while in the respiration chamber only dissolved oxygen is measured. Oxygen uptake rate (OUR) data were calculated from the two DO-trajectories as described in Petersen et al. (2001). The data acquisition frequency of the sensors is set to 3 seconds. High frequency noise, known to be present in the weak analog signals of the electrodes of the setup, is filtered using a low pass Savitzky-Golay least square polynomial filter (Press et al., 1992) through a LabVIEW Matlab script node (Matlab R12, The MathWorks Inc., USA). The pH is controlled within a narrow pH band of ± 0.03 .



Figure 7.2: Picture of the hybrid respirometer installed at the BIOMATH lab.

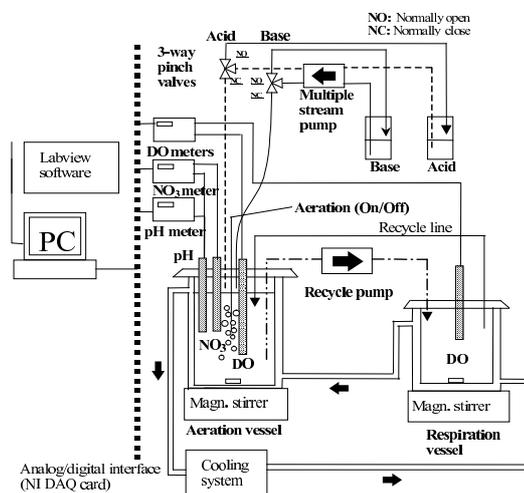


Figure 7.3: Schematic overview of the hybrid respirometer (from Sin et al., 2003).

Experiments were performed with pulse additions of $NH_4^+ - N$ added to the aeration vessel of the respirometer using a pump. For these experiments

biomass from the Maria Middelaes wastewater treatment plant (Gent, Belgium) was used.

Strong autocorrelations were found in the oxygen uptake rate data. This is shown in Figures 7.4 and 7.5. Figure 7.4 shows the measured oxygen uptake rate (after filtering) for an experiment where an ammonium pulse was added to an activated sludge sample. It also shows the model fit of a one-step nitrification model to the data (see further). The difference between the model prediction and the measurements, often called residuals (ϵ), are shown in Figure 7.5. From this graph, it is clear that the residuals show an oscillatory pattern and are not randomly distributed which would be expected if the noise would have had white characteristics. In order to determine the extent of the autocorrelation, the autocorrelation function (Equation 7.1) was calculated (Dochain and Vanrolleghem, 2001). The autocorrelation $r(\tau)$ with time lag τ quantifies the dependency of the variable at time t on the variable at time $t - \tau$.

$$r(\tau) = \frac{\sum_{t=\tau}^{n-\tau} \frac{\epsilon(t-\tau)\epsilon(t)}{n-2\tau}}{\sum_{t=1}^n \frac{\epsilon^2(t)}{n}} \quad (7.1)$$

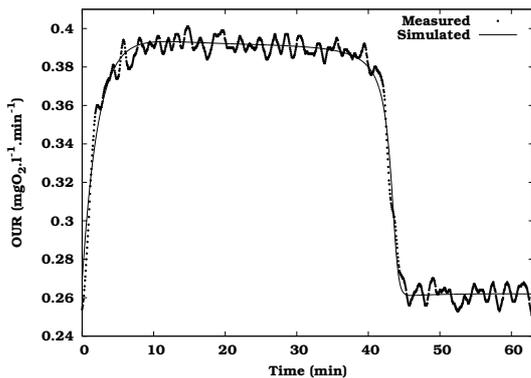


Figure 7.4: Oxygen uptake rate (OUR) data (after filtering) obtained from an ammonium pulse addition to an activated sludge sample and the model fit of a one-step nitrification model.

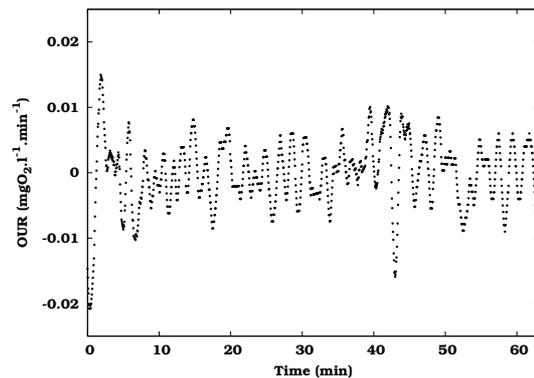


Figure 7.5: Residuals between the model simulation and the measured data shown in Figure 7.4.

In order to test the significance of the autocorrelation, the value at each time lag is compared to a limit value of $\pm N(0, 1)/\sqrt{n}$, where $N(0, 1)$ is the standard normal distribution and n is the sample size. For a significance level $\alpha = 0.05$ this limit value equals $1.96/\sqrt{n}$. Autocorrelations located within the band determined by the limit value are said to be insignificant. The autocorrelation function for the experimental data discussed above is shown in Figure 7.6 together with the band determined by the limit values equal to ± 0.055 . From this figure it is clear that the noise is not white because then the autocorrelation should have immediately dropped within the ± 0.055

band. Significant autocorrelation was found up to a lag of ± 40 seconds (every 13th data point). This was automatically corrected in LabVIEW by sub-sampling each 13th data point and resulted in the data and residuals shown in Figures 7.7 and 7.8.

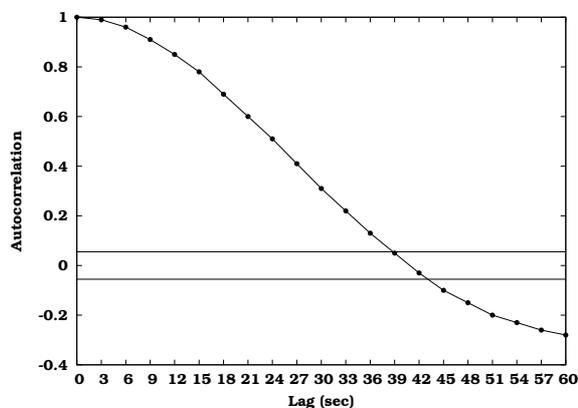


Figure 7.6: Autocorrelation function for the data shown in Figure 7.4. The horizontal lines represent the band beyond which autocorrelation is significant.

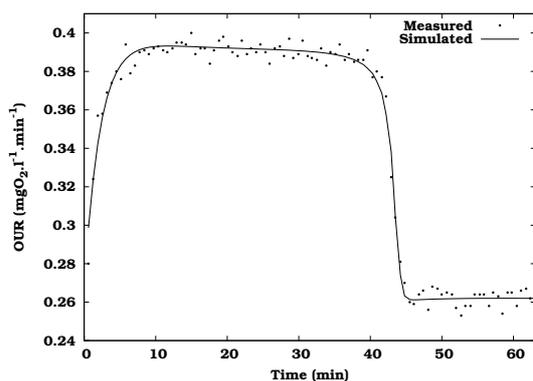


Figure 7.7: Sub-sampled oxygen uptake rate (OUR) data obtained from an ammonium pulse addition to an activated sludge sample and the model fit of a one-step nitrification model.

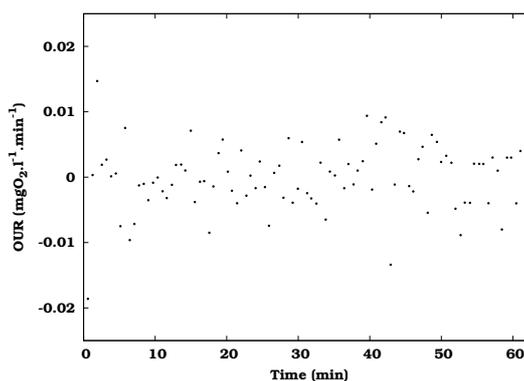


Figure 7.8: Residuals between the model simulation and the measured data shown in Figure 7.7.

7.3.2 LabVIEW communication

In order to perform automatic experiments, communication between the EAST automatic experimental design module and the LabVIEW VI (virtual instrument) which controls the experimental setup, should exist. At the LabVIEW side, this is accomplished by embedding the VI of the experimental setup in a general communication VI (Figure 7.9). Different stages in this communication process can be distinguished:

1. As a first step, LabVIEW, installed on the slave PC, is required to connect to the master PC running the EAST automatic optimal experimental design module. This is accomplished by opening a TCP socket network connection to the IP address of the master PC.
2. Once the connection is established, the slave PC enters a loop and waits until information is received from the master PC.
3. If the information received from the master PC is valid, the values of the experimental degrees of freedom are decoded from the incoming data stream.
4. Next, the experimental degrees of freedom are transferred to the sub-VI which controls the experimental setup. Based on these experimental degrees of freedom the experiment is performed and data collected.
5. After the experiment has finished, the collected measurements are encoded into a data stream which is sent to the master PC running the EAST automatic optimal experimental design module. If this is completed successfully, the VI again enters the waiting loop in order for the process to repeat itself (steps 2-5).
6. Once the automatic optimal experimental design module has successfully calibrated the model, the connection with the master PC is closed.

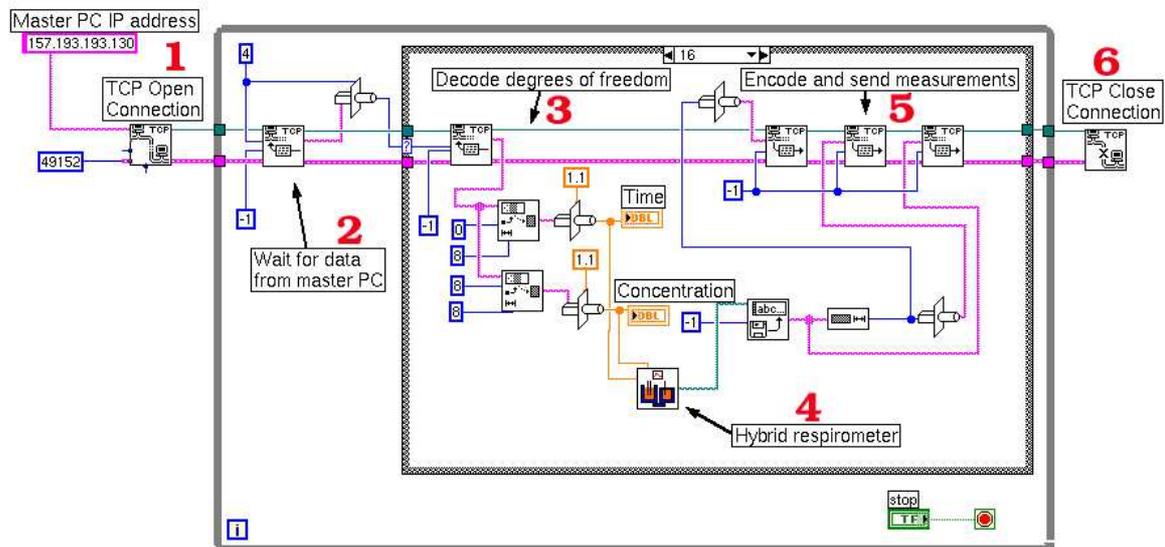


Figure 7.9: General LabVIEW communication VI (virtual instrument) connecting the EAST automatic optimal experimental design module and the VI controlling the experimental setup.

The use of this general communication VI enables any experimental system controlled by a VI to be used in the developed automatic optimal experimental design procedure. The only modifications that have to be made to

the original VI are the creation of the inputs for the experimental degrees of freedom and an output for the collected measurements.

7.3.3 Identification of a one-step nitrification model

To illustrate the proposed procedure, it is applied to the one-step nitrification model described in Section 2.2.3 and using the respirometer setup described in this chapter. As a first step in the procedure, the model was calibrated with OUR data of an initial experiment (Figure 7.10). For this experiment, lasting 35 minutes, a pulse of $0.5 \text{ mgNH}_4^+ - \text{N}$ was added to the aeration vessel at 12 minutes after the start of the experiment. Five parameters were selected for parameter estimation: μ_{max, NH_4} , K_{NH_4} , Y_{NH_4} , τ and X_H since it has been shown that these parameters are theoretically identifiable based on OUR measurements without assuming biomass growth (Petersen, 2000; Marsili-Libelli and Tabani, 2002). In order to fit the endogenous respiration rate, the heterotrophic biomass concentration X_H was selected for parameter estimation while b_H and f_P were kept at their default value of $1.39E-04 \text{ min}^{-1}$ and 0.2 respectively. The autotrophic biomass concentration X_{NH_4} is also assumed to be known and was fixed here at a value of 40 mg.l^{-1} . This value was calculated based on steady-state assumptions and an assumed default yield, sludge retention time, hydraulic retention time and organic load (Metcalf and Eddy, 1991). Figure 7.10 shows the model fit to the data of the initial experiment. The calibrated parameters and their confidence intervals are listed in Table 7.1. From the table it is clear that the confidence intervals of the parameters are extremely large. In order to improve this, the automatic optimal experimental design procedure was run.

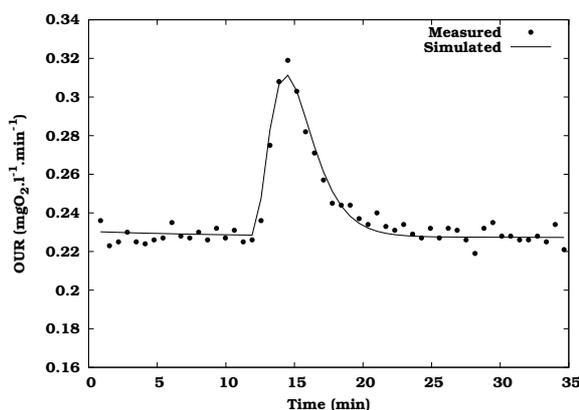


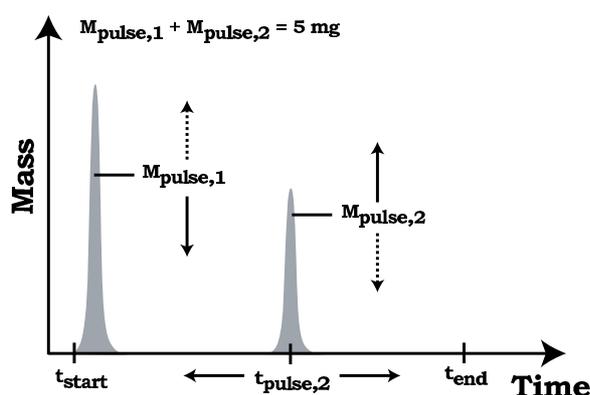
Figure 7.10: Measured OUR of the initial experiment (injection of $0.5 \text{ mgNH}_4^+ - \text{N}$ at $t = 12 \text{ min}$) and simulated OUR of the calibrated one-step nitrification model.

Table 7.1: Values and confidence intervals for the estimated parameter based on the data of the initial experiment.

Parameter	Value	95 % confidence interval (δ)
Y_{NH_4} ($mgCOD.mgN^{-1}$)	$4.16E-02$	± 1.95 (4682 %)
μ_{max,NH_4} (min^{-1})	$3.28E-03$	$\pm 3.05E-02$ (927 %)
K_{NH_4} ($mgN.l^{-1}$)	4.28	$\pm 2.18E+02$ (5118 %)
τ (min)	1.93	± 2.04 (106 %)
X_H ($mgCOD.l^{-1}$)	1854	± 18.89 (1.02 %)

7.3.4 Experimental degrees of freedom and constraints

The next step in the proposed procedure is to fix the experimental degrees of freedom and constraints. It was decided to perform an experiment with two pulse additions of $NH_4^+ - N$ where the total amount ($M_{pulse,1} + M_{pulse,2}$) of $NH_4^+ - N$ was limited to 5 mg in order to limit biomass growth (Grady et al., 1996). The duration of the experiment was fixed to 110 minutes ($t_{end} - t_{start}$) and the experimental degrees of freedom to be optimized were chosen to be (1) the amount of the second addition and (2) the time instant of the second pulse addition. The first pulse of $NH_4^+ - N$ (given by an amount equal to 5 mg minus the amount of the second pulse) was always added at the start of the experiment (t_{start}). These experimental degrees of freedom and constraints are schematically shown in Figure 7.11. The pulse additions were modelled as narrow Gaussian peaks in order to prevent numerical problems during model simulations and optimal experimental design. Since the optimization problem only consists of continuous experimental degrees of freedom, the Simplex method (Nelder and Mead, 1965) could be used as optimization algorithm. This algorithm converges faster for this type of problems.

**Figure 7.11:** Schematic representation of the experimental degrees of freedom and constraints of a two pulse $NH_4^+ - N$ addition experiment.

7.3.5 Objective selection

The final user interaction needed before the automatic procedure can be launched, is the choice of the objective criterion that will be used to quantify the information content of a proposed experiment. In this case, the D-optimal design criterion was chosen in order to decrease the overall confidence region of the parameter estimates. The automatic experimental design procedure is stopped when the largest 95 % confidence interval is smaller than ± 20 % of the parameter value or when the confidence interval shows no improvement compared to the previous experiment.

7.3.6 Results and discussion

Now that the initially calibrated model, the experimental degrees of freedom, constraints and the objective are available, the first iteration of the automatic procedure can be started. Based on the parameters obtained from the first experiment, an optimal experiment was searched for by the Simplex optimization algorithm using the experimental degrees of freedom and constraints. The experiment that was found to be optimal was the addition of almost all $5 \text{ mgNH}_4^+ - N$ in the first pulse at the start of the experiment while a negligible amount had to be added using the second pulse. For this experiment the D criterion was $2.07E+15$. Once the optimal experiment was found, the values of the experimental degrees of freedom for the optimal experiment were transferred automatically to the LabVIEW setup (slave PC) and the experiment was started. One hundred and ten minutes later, when the experiment ended, the data were transferred to the master PC. As soon as the data were received, the model was recalibrated using the parameter values of the already available model as initial estimates. Again, the calibration was performed using the Simplex optimization method. Figure 7.12 shows the data of this experiment together with the model fit. Table 7.2 provides the parameter values found after calibration together with their 95 % confidence interval. From the table it is clear that the confidence in the parameters has increased considerably compared to the calibration results of the initial experiment (Table 7.1).

Next, a second iteration of the procedure was started because the largest estimation error was still above 20 %. Due to the changed (recalibrated) model parameters and the local nature of FIM-based optimal experiments, a different optimal experiment was found. This time it consisted of one pulse of $4.17 \text{ mgNH}_4^+ - N$ at the start of the experiment and another pulse of $0.83 \text{ mgNH}_4^+ - N$ after 48.32 minutes. This corresponds to an addition just after the time when the first pulse of $\text{NH}_4^+ - N$ is completely taken up by the biomass. The FIM D-optimal criterion for this experiment increased to $1.26E+23$. Vanrolleghem et al. (1995) also obtained an optimal experiment with extra pulse addition after the substrate was almost exhausted for a respirometric experiment similar to the one described here. Again, the experimental degrees of freedom were transferred to the slave PC and the

experiment was performed. After the experiment, the data were sent to the master PC and the model was recalibrated. Figure 7.13 and Table 7.2 show the data and the results of the model calibration. From the table it is clear that the parameters have not changed much but, probably surprisingly, the errors on the parameter estimates have increased. Although the information content of the second optimal experiment was higher (see above), the model fit was less good than the fit on the first optimal experiment. This worse fit is a result of a model structure problem. Indeed, the model was unable to optimally fit the acquired data and the increased residual error propagated in the higher parameter estimation errors. Note that the FIM D-optimal criterion is independent of the actual residuals while the parameter confidence is (see Equation 1.12). Since no further improvement was made in parameter confidence, the optimal experimental design procedure was stopped at this point.

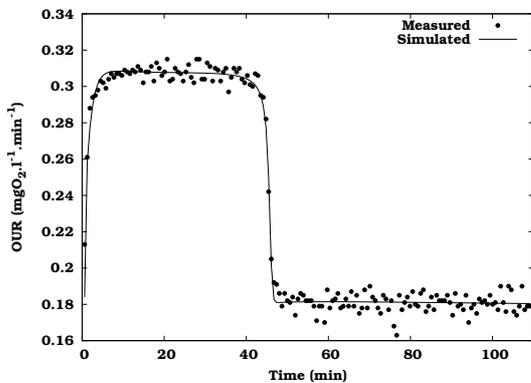


Figure 7.12: Model calibration based on the OUR data of the first designed experiment.

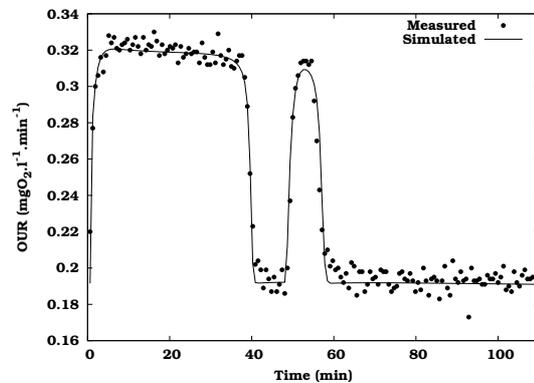


Figure 7.13: Model calibration based on the OUR data of the second designed experiment.

Table 7.2: Values and confidence intervals for the estimated parameter based on the designed experiments.

Parameter	First designed experiment		Second designed experiment	
	Value	95 % confidence interval (δ)	Value	95 % confidence interval (δ)
Y_{NH4} ($mgCOD.mgN^{-1}$)	1.01	$\pm 3.45E-02$ (3.40 %)	$8.24E-01$	$\pm 5.36E-02$ (6.50 %)
$\mu_{max,NH4}$ (min^{-1})	$9.41E-04$	$\pm 3.23E-05$ (3.44 %)	$7.31E-04$	$\pm 4.73E-05$ (6.47 %)
K_{NH4} ($mgN.l^{-1}$)	$8.65E-03$	$\pm 1.82E-03$ (21.1 %)	$6.66E-03$	$\pm 1.55E-03$ (23.3 %)
τ (min)	1.39	$\pm 1.52E-01$ (10.9 %)	1.04	$\pm 1.21E-01$ (11.6 %)
X_H ($mgCOD.l^{-1}$)	1472	± 11.75 (0.82 %)	1561	± 16.22 (1.04 %)

Looking at the parameter values in Table 7.2 it is clear that the yield parameter has a quite different value from the often reported $0.24 mgCOD.mgN^{-1}$

for autotrophic biomass. In order to find an explanation for this deviating value, the area below the OUR profile can be calculated. This area corresponds to $(4.57 - Y_{NH_4}) \times S_{NH_4}(0)$ where $S_{NH_4}(0)$ is the added $NH_4^+ - N$ concentration which is assumed to be known (Petersen et al., 2001). For the first designed experiment, the integrated surface corresponds to $5.55 \text{ mgO}_2.l^{-1}$ which is identical to the calculated value ($5.56 \text{ mgO}_2.l^{-1}$) using a yield of $1.01 \text{ mgCOD.mgN}^{-1}$ and assuming the concentration of $1.56 \text{ mgNH}_4^+ - N.l^{-1}$ ($5 \text{ mgNH}_4^+ - N$ in 3.2 l) to be known. If we assume a default yield of $0.24 \text{ mgCOD.mgN}^{-1}$ and calculate the $NH_4^+ - N$ concentration that would correspond to an oxygen utilization of $5.55 \text{ mgO}_2.l^{-1}$, it becomes $1.28 \text{ mgNH}_4^+ - N.l^{-1}$. This would be as if the amount available to the nitrifiers is significantly less than the assumed value. This gap in $NH_4^+ - N$ could be explained by several things: (1) adsorption of ammonium on biomass, (2) assimilation of ammonium by heterotrophic biomass, (3) inaccuracies of the ammonium addition pump. Indeed, the effect of these processes can be significant because only a small amount of $NH_4^+ - N$ is added to the system.

7.3.7 Evaluating the optimization of the experimental degrees of freedom

In the previous section the optimisation of the experimental degrees of freedom was performed using the Simplex algorithm. In Chapter 5 however, it has been shown that this optimizer does not always converge to the global optimum. It would therefore be interesting to investigate the D-criterion objective surface in detail to determine whether the global optimum was found during the automated procedure.

In Figures 7.14 and 7.15 the objective surface for the FIM D-criterion is shown as a function of the two experimental degrees of freedom: the mass of the second pulse and the time instant of the second pulse. Remember that the constraint on the sum of the masses of both $NH_4^+ - N$ pulses equals 5 mg . In order to construct the surface, a grid of combinations of the experimental degrees of freedom was composed. The experimental degrees of freedom at each grid point were used to simulate an experiment and calculate the FIM D-optimal criterion. The difference between both figures is the model used to simulate the experiments. In Figure 7.14 the model calibrated on the basis of data obtained during the initial experiment (Table 7.1) was used while for Figure 7.15 the calibrated model based on data obtained during the first designed experiment (Table 7.2 2nd column) was used. The difference between the shape of both objective surfaces clearly demonstrates that the FIM only yields locally optimal experiments which depend on the nominal parameter values used. In other words, the FIM is largely influenced by the parameter values of the model used to calculate it. A closer look at Figure 7.14 shows that the highest value of the D-optimal criterion ($2.07E+15$) can be reached by different combinations of the experimental degrees of freedom. Optimal experiments are found at the left, back

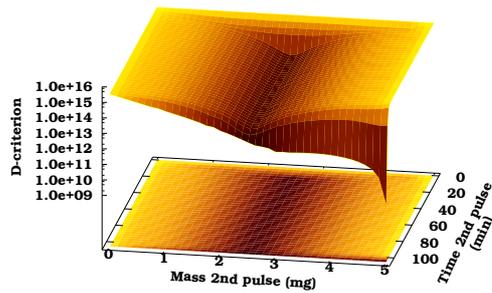


Figure 7.14: D-optimal criterion objective surface as a function of the experimental degrees of freedom calculated using the parameters calibrated on the data of the initial experiment.

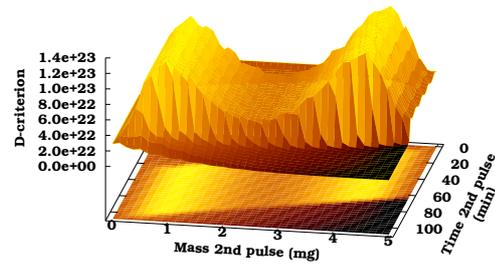


Figure 7.15: D-optimal criterion objective surface as a function of the experimental degrees of freedom calculated using the parameters calibrated on the data of the first designed experiment.

and right borders of the surface. The left border corresponds to experiments with second pulses of 0 mg and thus first pulses of 5 mg . The optimizer also converged to this optimum during the automatic optimal experimental design procedure. The back border of the objective surface corresponds to experiments with time instances for the second pulse equal to 0 min . In this situation, both pulses coincide and the total mass of 5 mg is again supplied as one large pulse. From the right border of the surface it can be seen that experiments where the total mass is concentrated in the second pulse (no mass in the first pulse) are also optimal. It can also be seen that the time instant at which the single pulse is added is not very important. However, the pulse should not be given near the end of the experiment because then valuable information is lost. This is clearly visible by the sharp decrease in objective function value in the right front of the surface.

For Figure 7.15 only one optimum exists for a D-optimal criterion value of $1.26E+23$, the value also found by the optimization algorithm. This value corresponds to an experiment with a second pulse of 0.83 mg added at 48.32 minutes after the start of the experiment. However, another peak exists in the objective surface. This peak corresponds to an experiment with a second pulse of 4.13 mg added after 62.11 minutes and has a D-optimal criterion value of $1.19E+23$. Although this experiment is slightly less optimal, it is essentially the same but inverted: a small first pulse and a large second pulse in contrast to the large first pulse and the small second pulse of the optimal experiment.

In this case, it was fortunate that the optimizer was able to detect the global optimum since it was started at initial values corresponding with the optimal values of the experimental degrees of freedom of the first designed experiment. If other initial conditions would have been used, the sub-optimal

experiment might have been found. This again illustrates the drawbacks of the use of local optimization algorithms in optimal experimental design.

7.4 Conclusion

In this chapter an automatic optimal experimental design procedure was proposed. In this procedure, user interaction and expert knowledge is only required at the beginning and all subsequent steps of the iterative search for the best parameter estimates can be performed automatically. These steps include (1) finding the optimal experiment, (2) performing the experiments in practice and (3) recalibrating the model. The proposed procedure was implemented in EAST and interacts with a slave LabVIEW-PC. This slave PC controls the experiment and can be located anywhere in the world. The proposed procedure was applied successfully to the calibration of a one-step nitrification model using respirometric data. After an initial model calibration, an optimal experiment was designed using the parameter values obtained, and performed automatically by transferring the experimental degrees of freedom through the Internet to the LabVIEW experimental setup. The acquired data was sent back to the experimental design module where the model was automatically recalibrated and a second and final experiment designed, performed and used for a final recalibration. This procedure resulted in significantly smaller confidence intervals for the estimated parameters. After the automatic optimal experimental design procedure, the D-criterion objective surfaces were studied to gain more insight in the designed experiments. It was found that the shape of the surfaces was largely dependent on the model parameter values used to design the experiments, again stressing the local nature of FIM-based optimal experiments.

Chapter 8

Robust experimental design

8.1 Introduction

An important aspect of optimal experimental design for parameter estimation that has not been focused on so far in this thesis is the dependency of the design on the model parameters. Indeed, the basis for optimal experimental design is the Fisher Information Matrix (FIM) which is calculated from sensitivity functions, the partial derivatives of the model variables to the parameters. For non-linear models, these partial derivatives are still function of the model parameters which means that the FIM is directly influenced by the parameter values themselves. Therefore, all experimental designs based on FIM properties are called local designs. The effectiveness of the design thus depends on how close the model parameters are to those of the real system. This poses a problem, since the final goal of optimal experimental design for parameter estimation is to find the true system parameters, or at least approximate them. Cochran (1973) rephrases this problem nicely:

The statistician can say to the experimenter: “You tell me the values of the parameters and I promise to design the best experiment for estimating them”. Upon which the experimenter replies: “Who needs you”.

This issue is a fundamental one for non-linear design problems. Several approaches have been proposed in literature to deal with the robustness of the experimental design to parameter uncertainty and will be discussed briefly in the next section of this chapter. Next, in a simulation study of

a fed-batch bioreactor (Section 2.2.1), several of these techniques will be implemented and compared with a new approach.

8.2 Overview of existing techniques

8.2.1 Sequential design

The first and most often used approach to overcome design problems related to parameter uncertainty is to design experiments in a sequential way by alternating parameter estimation and experimental design. Each estimation improves the knowledge of the system parameters and this knowledge can then be used to improve the quality of the next experiment to be performed. Many authors acknowledge the usefulness of this approach (e.g. Ford et al., 1989; Walter and Pronzato, 1990; Atkinson, 2003). The general optimal experimental design procedure described in Section 1.3.3 is based on this principle. An application of sequential design was discussed in Chapter 7 for the estimation of parameters related to activated sludge nitrification. However, some drawbacks are associated with this technique. Firstly, it might not be possible to perform multiple (sequential) experiments on the same system due to limitations in time or resources. Secondly, it is not guaranteed that the parameters will converge to the “true” values. For some cases including the example in Chapter 7, fast convergence was obtained (Bernaerts et al., 2000a; Versyck and Van Impe, 2000b) but this might not always be the case.

8.2.2 Discrete design

In the framework of optimal measurement scheduling (time and place) a design can be made more robust by spreading measurements in the neighborhood of the design points found in the “classical” way. Deviations between the model parameter values and the “true” system values will result in a shift of the points of highest information content. This effect could be compensated by a more distributed measurement scheme.

As was seen in Chapter 5, when measurement times were optimized using continuous variables, many optimal measurement points coincided. However, by using discrete variables, a design was found in which measurement points were distributed around the optimal points. This design was found to have more practical value and is at the same time more robust to misspecified model parameters. A similar approach was followed by Grijspeerdt and Vanrolleghem (1999) who designed experiments in order to estimate parameters of a bacterial growth model. In their research, 4 distinct measurement times with several repetitions at each of these optimal points were found. A more distributed sampling scheme around these points resulted in improved parameter estimates due to the fact that the parameters of the model which was used for the experimental design were different from reality.

8.2.3 Expected value approach

Optimizing the mathematical expectation of the scalar properties of the FIM (design criteria) over the population of possible parameter values is a natural extension of the classical approaches toward more robust designs. Up until 1987 no direct application of this idea has been attempted because of the complexity of the required computations (Walter and Pronzato, 1987). However, since then several applications have been reported for linear and relatively simple non-linear models (Walter and Pronzato, 1987; Tod and Rocchisani, 1996; Foracchia et al., 2004). Applications to more complex non-linear dynamic models (ODE systems), however, are very scarce: only one application could be found in literature (Asprey and Macchietto, 2002). This type of robust experimental design has mainly been applied to the D-optimal FIM criterion:

$$J_D(\theta, \psi) = \det [FIM(\theta, \psi)] \quad (8.1)$$

where θ represents the model parameters and ψ the experimental degrees of freedom. An experiment ψ_D is said to be D-optimal if it satisfies:

$$\psi_D(\theta) = \arg \left\{ \max_{\psi \in \Psi} [J_D(\theta, \psi)] \right\} \quad (8.2)$$

in which $\psi_D(\theta)$ are the values of the optimal experimental degrees of freedom, chosen from the domain of all possible values of the experimental degrees of freedom Ψ , for which the D-optimal criterion is maximized.

Now, we assume a prior distribution or probability density function of the parameters θ : $Pr(\theta)$. The criterion J_D therefore becomes a random variable with its own distribution. One way to optimize it, is to maximize the expectation of J_D given $Pr(\theta)$. An experiment ψ_{ED} is said to be ED-optimal if it satisfies (Tod and Rocchisani, 1996):

$$\psi_{ED}(\theta) = \arg \left\{ \max_{\psi \in \Psi} [E [J_D(Pr(\theta), \psi)]] \right\} \quad (8.3)$$

with

$$E [J_D(Pr(\theta), \psi)] = \int_{\Theta} J_D(\theta, \psi) Pr(\theta) d\theta \quad (8.4)$$

The expectation is calculated with respect to the prior distribution $Pr(\theta)$, over the entire parameter space Θ and the maximization is performed over the entire design space Ψ . Similarly, experiments are said to be EID- or API-optimal if they satisfy (Tod and Rocchisani, 1996):

$$\psi_{EID}(\theta) = \arg \left\{ \min_{\psi \in \Psi} \left[E \left[\frac{1}{J_D(Pr(\theta), \psi)} \right] \right] \right\} \quad (8.5)$$

$$\psi_{API}(\theta) = \mathit{arg} \left\{ \begin{array}{l} \mathit{max} \\ \psi \in \Psi \end{array} [E [\log (J_D (Pr(\theta), \psi))]] \right\} \quad (8.6)$$

These criteria lead to different experiments and thus naturally the question is raised of which criterion to use? In Walter and Pronzato (1987) EID-optimality is shown to have advantages over ED-optimality since minimizing $E [1/J_D (Pr(\theta), \psi)]$ can be viewed as minimizing the average value of the square of the volume of the asymptotic parameter estimation confidence region. According to Tod and Rocchisani (1996) the API-optimal criterion maximizes the median of J_D over the parameter domain, whereas the ED-optimal criterion maximizes the mean. The EID-optimal criterion will be dominated by the largest $1/J_D$ values over the population, thus the smallest J_D values, corresponding to “realities” whose parameters are most difficult to identify. In this context, “realities” is used to define a system with a certain set of “true” parameter values.

Since optimal robust designs can almost never be obtained analytically, numerical methods need to be used. Typically, numerical integration using a discretized version of the probability density function is used to evaluate the expectation (Tod and Rocchisani, 1996). This solution is numerically quite burdensome if fine discretizations are used and many parameters are involved. Therefore, Asprey and Macchietto (2002) use a multi-dimensional quadrature rule for approximating the multiple integral (over all parameters). Another alternative would be to use Monte Carlo techniques to approximate the expectation, requiring however a large number of realizations before convergence is achieved.

In order to prevent the numerical integration, Walter and Pronzato (1987) present a stochastic optimization algorithm which does not require the evaluation of the expected values and is based on the gradient of the FIM to the experimental degrees of freedom. This method was found to be very sensitive to the initial values of the experimental degrees of freedom and was also found to converge very slowly in some situations (Tod and Rocchisani, 1996). It was therefore coupled to a random search algorithm in order to provide a decent starting point for the stochastic optimization algorithm (Foracchia et al., 2004).

Although very appealing, these criteria have not been widely used, for two main reasons. Firstly, the prior probability distributions are generally unknown. Secondly, this type of robust design is good in the average sense but can be poor for some values of the parameters that are associated with very low probability values.

8.2.4 Maximin approach

Another approach to robust experimental design aims at determining the experiment ψ_{MMD} that optimizes the worst possible performance for any value of θ belonging to the parameter domain Θ (Pronzato and Walter, 1988):

$$\psi_{MMD}(\theta) = \arg \left\{ \max_{\psi \in \Psi} \left[\min_{\theta \in \Theta} [J_D(\theta, \psi)] \right] \right\} \quad (8.7)$$

In other words, for a proposed design, find the parameters for which the D-optimal criterion value is the lowest, i.e. this D-criterion value determines the worst possible obtainable information content for this specific design. Next, find the design which maximizes this worst D-optimal criterion value.

For this technique, the prior information on the parameters is limited to the knowledge of the parameter domain Θ , i.e. the upper and lower bounds of the parameters. No information on the distribution is necessary.

This approach for robust design is only recently being applied due to the computational burden introduced by the nested optimization. Applications of this technique using classical non-linear optimization algorithms are given in King and Wong (1998) for the design of heteroscedastic statistical models and in Asprey and Macchietto (2002) for the design of a fed-batch measurement campaign.

8.3 Simulation study

8.3.1 System setup and data generation

In this section, robust optimal experimental design will be applied to the stirred fed-batch bioreactor presented and described in Sections 1.1 and 2.2.1. For the remainder of the calibrations and optimal experimental design performed in this chapter, this model structure will be used and assumed correct. The model parameters, initial conditions and operational conditions are listed in Table 1.1. As indicated in the table, the reactor is operated in batch mode for the first 6 hours, after which the operation is switched to fed-batch mode and substrate is fed to the reactor for an additional 4 hours. For this system, two measurement variables will be considered: substrate concentration S ($g.l^{-1}$) and biomass concentration X ($g.l^{-1}$). The simulation results for both variables were used to create virtual data, assuming a measurement error of 5 and 2 % for substrate and biomass respectively and a measurement interval of 5 minutes. These data are shown in Figures 8.1 and 8.2. The relative measurement errors were taken identical as in Chapter 1 and are based on an analysis performed by Baltes et al. (1994). The use of this type of measurement error has obvious consequences for the weights that will be used during parameter estimation and optimal experimental design. The weights corresponding to this particular situation are shown in Figure 1.16.

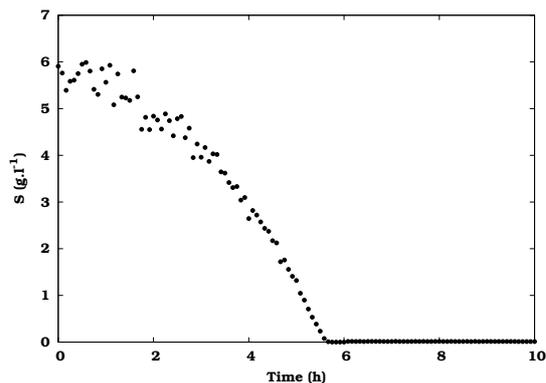


Figure 8.1: Artificially generated substrate measurements for a fed-batch experiment.

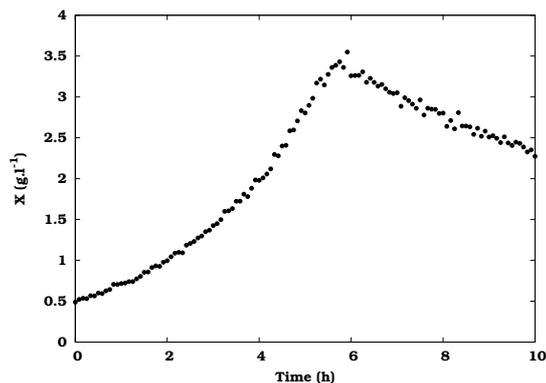


Figure 8.2: Artificially generated biomass measurements for a fed-batch experiment.

8.3.2 Experimental degrees of freedom and objective

The experimental degrees of freedom considered in this case are the time instances of 10 measurements for each of the measurement variables (S and X). For these measurements, no repetitions are allowed and a minimal period of 5 minutes between two consecutive measurements should be maintained (discrete optimization variables). It has to be mentioned that this property can also be regarded as a form of robust experimental design (Section 8.2.2). In order to optimize these experimental degrees of freedom a genetic algorithm (GA) will be used.

In this study, 4 model parameters will be considered for optimal experimental design: μ_{max} , K_S , Y and K_d . The objective used in this study will be the minimization of the confidence region of these parameters (D-optimal design). Therefore, all robust experimental design criteria will be based on D-optimality.

8.3.3 Initial design and calibration

As a starting point for the robust optimal experimental design, a classical D-optimal experimental design was performed. At this point, the parameter values are still unknown and an initial guess needs to be used. These initial guesses are listed in Table 8.1 (initial values) and are deliberately chosen far from the true values (from which the virtual data is generated). Based on these parameter values, the experimental degrees of freedom (20 measurement times) were optimized using a GA with settings shown in Table 8.2. At the optimal measurement times found after optimization (D-optimal criterion value: $1.07+21$), the virtual data was sub-sampled. These data are shown in Figures 8.3 and 8.4 for substrate and biomass respectively. As is shown in the figures, most substrate measurements are located around 2 hours, while one measurement is situated at 7.5 hours. For the

biomass measurements, two regions with each 5 measurements can be distinguished: a first group of measurements around 2 hours and a second group at the end of the experiment.

Table 8.1: Results of the initial fed-batch model calibration.

Parameter	Unit	True value	Initial value*	Estimated value	95 % conf. interval (δ)	Estimation error ($e_{\hat{\theta}}$) (%)
μ_{max}	h^{-1}	0.39	0.7	0.3537	0.0510	14.42
K_S	$g.l^{-1}$	0.06	0.006	0.0627	0.0117	18.65
Y	$gX.gS^{-1}$	0.54	0.3	0.4240	0.1398	32.98
K_d	h^{-1}	0.037	0.004	0.0011	0.0482	4376.39

*: optimal experimental design starting point

Table 8.2: GA settings used for the fed-batch measurement time optimization.

Attribute	Value
Population size	100
Replacement %	10%
Selection scheme	Roulette wheel
Scaling	Linear (multiplier: 1.2)
Crossover probability	0.8
Crossover operator	BLX- α ($\alpha = 0.5$)
Mutation probability	0.01
Mutation operator	Non-uniform

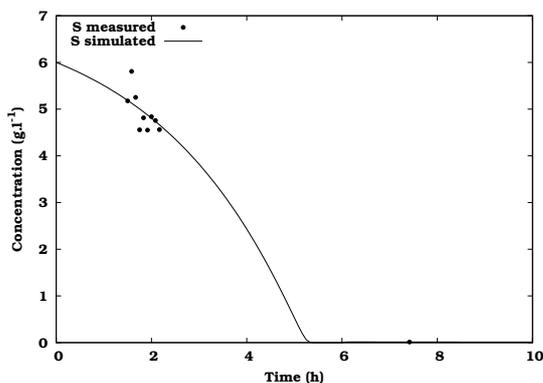


Figure 8.3: Substrate measurements at the optimal measurement times together with the simulation results of the calibrated fed-batch model.

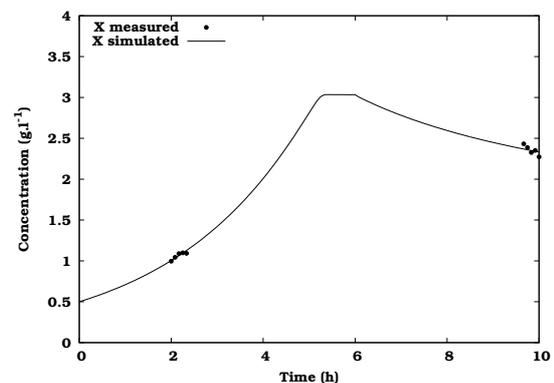


Figure 8.4: Biomass measurements at the optimal measurement times together with the simulation results of the calibrated fed-batch model.

Based on the data of the optimal experiment, the model was calibrated using the Praxis optimization algorithm. For this calibration it is assumed that

the initial values for substrate and biomass are known: $S_0 = 6 \text{ mg.l}^{-1}$ and $X_0 = 0.5 \text{ mg.l}^{-1}$. The results of the calibration are shown in Table 8.1. The table shows the estimated parameter values, their 95 % confidence interval and the estimation error (95 % confidence interval divided by the parameter value, multiplied by 100). From these results, it is clear that the true values could not be found and that the estimation errors are rather large, especially for K_d . These results nicely illustrate the influence of the parameter values of the model used in the experimental design on the outcome of the calibration. The optimal experiment which was found, would only have been truly optimal if the model parameters would have equaled the true system values.

Although the quality of the preliminary parameter estimation is not very high, it does provide important information on the bounds of the parameters between which the true parameter values are likely to be situated. In order to build in additional safety, the 99 % confidence intervals were also calculated (Table 8.3) and used to define the lower and upper bound of the parameters. Since the 99 % confidence intervals were used, there is a 99 % probability that the true parameter values are situated somewhere between these bounds. It also has to be mentioned that the lower bound of the confidence interval for K_d extended beyond 0 and was therefore truncated to 0 since negative K_d values are not acceptable.

Table 8.3: Lower and upper parameter bounds calculated based on 99 % confidence intervals (initial calibration).

Parameter	Estimated value	Variance	99 % conf. interval (δ)	Lower bound	Upper bound
μ_{max}	0.3537	$5.79E-04$	0.0703	0.2835	0.4241
K_S	0.0627	$3.04E-05$	0.0161	0.0466	0.0788
Y	0.4240	$4.35E-03$	0.1927	0.2313	0.6167
K_d	0.0011	$5.16E-04$	0.0664	0	0.0675

8.3.4 Implemented algorithms and settings

8.3.4.1 Expected value approach

The expected value approach to robust experimental design (Section 8.2.3) was implemented in EAST. A schematic overview of the implemented algorithm is shown in Figure 8.5. For this implementation, the expected value of the robust design criteria (ED, EID and API) was approximated using Monte Carlo techniques since the average obtained in this way is an estimator for the expectation value (Foracchia et al., 2004). Monte Carlo simulation requires samples or shots to be taken from the parameter space. Each of these shots is a vector of size p containing a randomly selected value for

each of the parameters. These parameter combinations are then used to evaluate the objective function (J_D in case of ED-optimal design).

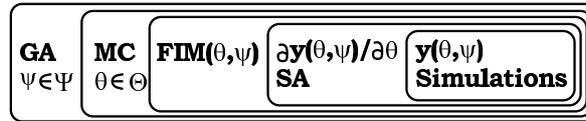


Figure 8.5: Schematic overview of the implemented expected value approach algorithm.

Conventional Monte Carlo would require a very large number of random samples from the parameter space in order to get an accurate approximation of the distribution of the criterion. Therefore, Latin Hypercube sampling (McKay et al., 1979) is used to limit the number of required shots. Latin Hypercube sampling divides each parameter probability distribution in s intervals (equal to the number of required shots) of equal probability. A value is selected randomly from each interval and the s values obtained for each parameter are combined randomly with the values of the other parameters resulting in s shots. This method ensures a full coverage of the range of each parameter by maximally stratifying each distribution. The question can then be raised of how many shots are required. This was experimentally determined by calculating the average D-optimal criterion for different numbers of shots, ranging from 1 to 150. These calculations were performed using the parameter bounds listed in Table 8.3 and assuming a uniform prior probability distribution on the parameters. A design with 10 uniformly distributed measurement points for each measurement variable was used to calculate the D-criterion values. The result of this investigation is shown in Figure 8.6. As can be seen from the figure, the average D-optimal value becomes stable for approximately 50 shots and higher. In order to account for the influence of the design on the required number of shots, a conservative number of 100 shots will be used.

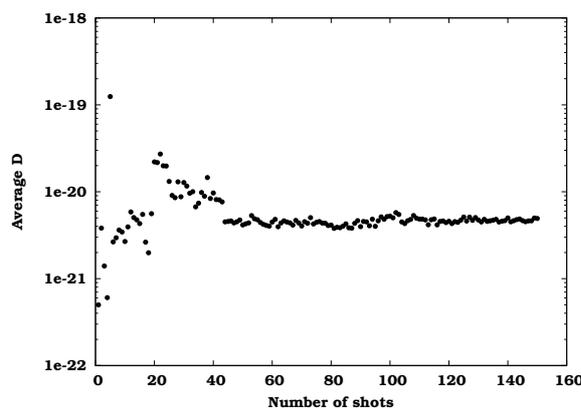


Figure 8.6: Average D-optimal (ED-optimal design) criterion value as a function of the number of Latin Hypercube samples.

Once the expected value of the design criterion can be calculated, it can be embedded in the optimization of the experimental degrees of freedom. In this study, the 20 measurement times are optimized using a GA with settings listed in Table 8.2. For each of the generated GA individuals (different designs), the expected value is calculated. This calculation involves the generation of 100 Monte Carlo shots from the parameter space. For each of these shots (parameter combinations) the D-optimal criterion is calculated, each requiring a new sensitivity analysis. These D-optimal criterion values are then used to calculate the average of the *ED*, *EID* or *API* criteria (depending on the objective of the optimization) which is then used as an estimate of the expectation value.

8.3.4.2 Maximin approach

The maximin approach to robust optimal experimental design (Section 8.2.4) was also implemented in EAST and applied to the fed-batch model. Solving this design problem required two nested GAs. A schematic overview of the implemented algorithm is shown in Figure 8.7.

The inner GA is used to find the parameter set which minimizes the D-optimal criterion (for a certain design). The parameters were encoded as continuous variables in the GA and the lower and upper bounds listed in Table 8.3 were used to constrain the optimization. The optimization settings of the GA listed in Table 8.2 were used, except for the population size which was set to 40. Each of the generated GA individuals (representing different parameter combinations) requires the calculation of a D-optimal criterion value and thus also a new sensitivity analysis.

The minimal D-optimal criterion value which is found using the inner GA optimization can then be used as the objective for the outer GA which optimizes the experimental degrees of freedom (20 measurement times). In this way, the outer optimization loop finds the design which maximizes the worst performing experiment. The GA settings of the outer loop optimization are listed in Table 8.2.

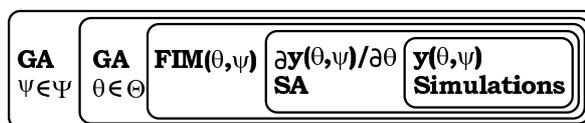


Figure 8.7: Schematic overview of the implemented maximin approach algorithm.

8.3.4.3 Sensitivity index approach

Basic criterion definition

In this section, a new approach to robust experimental design will be presented, the sensitivity index approach. The criterion used for this approach is defined in its most general form by Equation 8.8.

$$SI = \sum_{i=1}^p \sigma_{\theta_i}^2 \left(\frac{\partial J_{FIM_{crit}}}{\partial \theta_i} \right)^2 \quad (8.8)$$

This criterion is based on two components. Firstly, the (local) sensitivity of the FIM criterion ($J_{FIM_{crit}}$) with respect to the considered parameter. This expresses how sensitive the FIM criterion value is to parameter changes. Low sensitivities indicate designs for which the information content is not very sensitivity to changes in the parameter values, a desired property for a robust design. The second component of the criterion is the parameter estimation variance (obtained from a previous parameter estimation). This is used to amplify or tone down the FIM criterion sensitivity. For a parameter which has already been estimated accurately (small variance) a larger sensitivity can be allowed compared to a parameter which has not been estimated accurately (large variance). Generally speaking, the SI criterion should be as small as possible in order for the FIM criterion to be robust to parameter changes. A similar approach was also used by several authors in order to investigate if parameter changes would influence the FIM criterion values (Baltes et al., 1994; Petersen, 2000; Dette et al., 2003). However, these authors did not perform a robust experimental design, they only pointed out that indeed prior parameters choices influence the calculated designs.

Since the main goal of this study is to minimize the volume of the parameter estimation confidence region (D-optimal design), $J_{FIM_{crit}}$ from Equation 8.8 is replaced by J_D , leading to the SID criterion. This criterion can now be used in two ways: locally or globally.

LSID criterion

For the local approach (with respect to the parameters) a multi-objective optimization is performed at the point in parameter space of the initial calibration. The Pareto-optimal front, with respect to the D-optimal and SID criteria, is searched for by optimizing the experimental degrees of freedom (20 measurement times). In other words, which experimental designs maximize the D-optimal criterion and minimize the SID criterion. For this optimization, the NSGA-II algorithm (Section 6.4.3) was used with a population of size 100 and a total number of 1000 generations. Other GA settings are listed in Table 8.2. An overview of the implemented algorithm is shown in Figure 8.8.

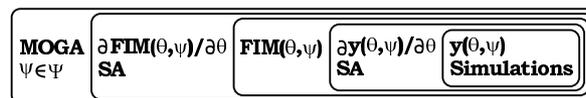


Figure 8.8: Schematic overview of the implemented LSID algorithm.

GSID criterion

Another approach (global with respect to the parameter domain, GSID-

optimal) would be to embed the SID calculation in two nested GA optimizations (similar to the maximin approach). The inner GA is again used to find the parameter set which maximizes the SID criterion (for a certain design). This corresponds to the parameter set for which the D-optimal criterion is most sensitive. Similar to the maximin approach, the parameters were encoded as continuous variables and the lower and upper bounds listed in Table 8.3 were used to constrain the optimization. The optimization settings listed in Table 8.2 were used, except for the population size which was set to 40. Each of the generated GA individuals (representing different parameter combinations) required the calculation of the sensitivity of the D-optimal criterion value to each of the parameters.

The maximal SID criterion value which is found using the inner GA optimization can then be used as the objective for the outer GA which optimizes the experimental degrees of freedom. In this way, the outer optimization loop finds the design which minimizes the highest sensitivity of the D-optimal criterion with respect to the parameters. The GA settings of the outer loop optimization are listed in Table 8.2. Similar to Equation 8.7, this robust experimental design can be written as:

$$\psi_{GSID}(\theta) = \arg \left\{ \min_{\psi \in \Psi} \left[\max_{\theta \in \Theta} [SID(\theta, \psi)] \right] \right\} \quad (8.9)$$

The calculation of the SID criterion requires the calculation of the local sensitivity of the FIM criterion to the parameters. In EAST, this was implemented using two approaches: a finite difference approach with fixed perturbation factors and a finite difference approach using Richardson's extrapolation. For the first approach, the sensitivity is calculated using central differences (Equation 3.4) requiring two FIM evaluations for each parameter and thus also two FIM sensitivity analyses. The Richardson's extrapolation approach is implemented as described in Section 4.2.3 and requires more FIM evaluations but does not require the perturbation factors to be specified. A schematic overview of the implemented algorithm is shown in Figure 8.9.

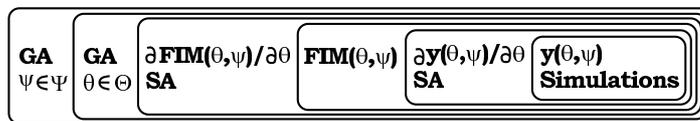


Figure 8.9: Schematic overview of the implemented GSID algorithm.

8.3.5 Results and discussion

Sequential design

As a first attempt to robust experimental design, another classical D-optimal design was constructed based on the initially calibrated model. This is an

example of sequential design, the most frequently used technique to tackle the parameter uncertainty in experimental design (Section 8.2.1).

The D-optimal measurement times which were found are illustrated in Figures 8.10 and 8.11 for substrate and biomass respectively together with the model simulations using the “true” parameter values. These figures give an overview of the optimal measurement designs for all design criteria tested in this simulation study. The D-optimal criterion value corresponding with this design was $1.76+24$, much higher compared to that of the initial design ($1.07+21$). The difference between the two D-optimal criterion values and the differences between the locations of the optimal measurement times for the initial design compared to this optimal design again illustrate that the parameter values of the model have a considerable influence on the constructed design. However, both designs show a similar property: all measurements are concentrated in only a few regions.

Based on the optimal measurement times, data was sub-sampled from the virtual data (Figures 8.1 and 8.2) and the model recalibrated. The results of the calibration are shown in Table 8.4. This table gives an overview of the model calibration results based on the optimal measurement times found for each design criterion. For each parameter, the estimated value and the estimation error (95 % confidence interval divided by the parameter value, multiplied by 100) are given. From these results it can be concluded that the parameter estimates are much closer to the true parameter values compared to the initial calibration (Table 8.1) and the associated estimation errors have decreased significantly.

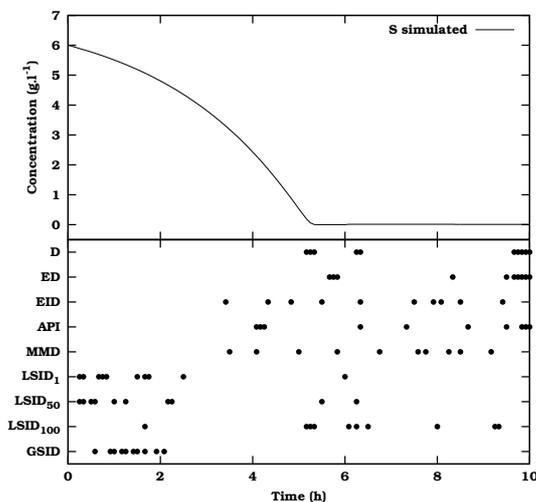


Figure 8.10: Overview of the optimal measurement times for substrate for all tested design criteria including the model simulations using the “true” values.

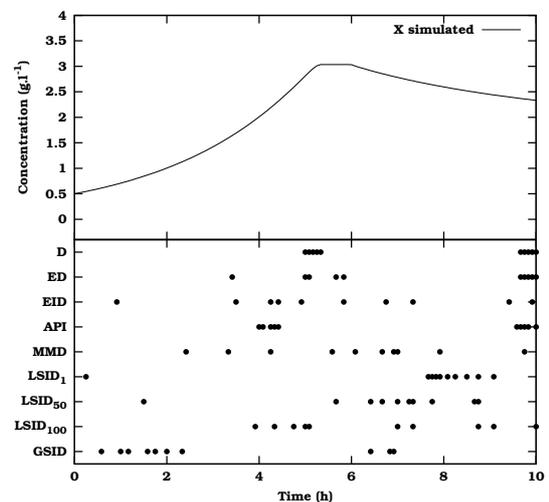


Figure 8.11: Overview of the optimal measurement times for biomass for all tested design criteria including the model simulations using the “true” values.

Table 8.4: Overview of the model calibration results based on the optimal measurement times found for each design criterion. For each parameter, the estimated value and estimation error (95 % confidence interval divided by the parameter value, multiplied by 100) are given.

	μ_{max}		K_S		Y		K_d	
	Value	Error	Value	Error	Value	Error	Value	Error
True	0.39		0.06		0.54		0.037	
Initial*	0.3537		0.0627		0.4240		0.0011	
D	0.3954	2.18	0.0573	3.41	0.5486	3.14	0.0416	15.79
ED	0.3849	0.08	0.0592	1.80	0.5283	1.03	0.0346	8.26
EID	0.3897	3.50	0.0600	4.39	0.5399	4.59	0.0376	29.51
API	0.3843	3.94	0.0578	5.22	0.5309	7.54	0.0357	37.82
MMD	0.3857	0.65	0.0597	4.80	0.5327	1.02	0.0336	10.69
LSID ₁	0.3970	2.09	0.0710	4.84	0.5436	3.29	0.0374	19.98
LSID ₅₀	0.3921	4.82	0.0572	8.29	0.5467	6.24	0.0389	38.38
LSID ₁₀₀	0.3936	3.28	0.0609	4.65	0.5453	4.44	0.0395	26.14
GSID	0.3467	56.28	0.0474	3721.32	0.4681	69.29	0.0005	31474.52

*: robust optimal experimental design starting point

Expected value approach

Next, robust experimental design in the average sense (expected value approach) was performed for three robust optimality criteria: ED, EID and API. The optimal measurement times for each of these criteria are shown in Figures 8.10 and 8.11. From the figures it can be seen that the ED- and API-optimal design show a similar measurement scheme in which measurements are concentrated in several regions, however not as pronounced as is the case for the D-optimal design. The EID-optimal design differs significantly from the two other criteria in the sense that the measurements are more evenly spread out.

For each of the design criteria, the optimal measurement times were used to sub-sample data from the virtual data (Figures 8.1 and 8.2) and recalibrate the model. The results of the calibration are listed in Table 8.4. Analysing the estimated parameter values shows that the EID-optimal design led to parameter estimates which were very close to the true values. For the other two criteria larger deviations were detected. However, the estimation errors for the ED-optimal design were found to be the smallest, followed by the EID-optimal design and the API-optimal design.

Maximin approach

The optimal measurement times found for the MMD-optimal design are shown in Figures 8.10 and 8.11. It can immediately be seen that the MMD-optimal design and the EID-optimal design show similar properties. This was also acknowledged by Walter and Pronzato (1990). For both criteria the measurements are spread out in the same regions.

The results of the calibration based on the optimal measurement times are shown in Table 8.4. This shows that the calibrated parameter values for μ_{max} , K_s and Y are close to the true values and that the estimation errors are quite small. For K_d , a larger deviation and associated error is observed. However, this is a general trend observed for all the designs.

Local sensitivity index approach

Before using the SID criterion globally (GSID), it will first be used in a local (LSID) multi-objective optimization together with the D-optimal criterion. The sensitivities of the FIM D-optimal criterion were calculated locally around the parameter set obtained from the initial calibration. At this point in parameter space, the experimental degrees of freedom were optimized in order to find the Pareto optimal front minimizing the LSID criterion (local SID) and maximizing the D-optimal criterion. The resulting Pareto front is shown in Figure 8.12. From an analysis of the front, it can be seen that there is a strong correlation between values for the LSID- and the D-optimal criterion. Low values for the LSID criterion correspond to robust designs for which the D-optimal criterion is not sensitive to parameter changes. These designs are also associated with low D-optimal criterion values indicating a low information content.

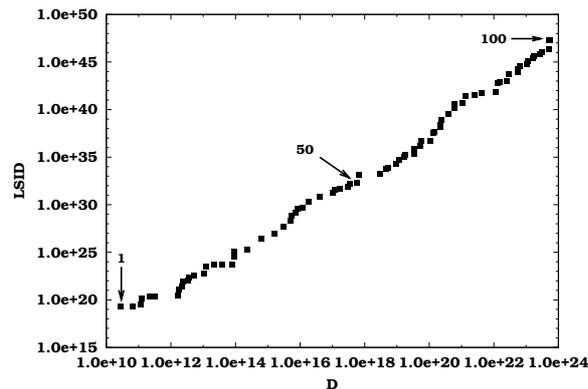


Figure 8.12: Pareto optimal front for a multi-objective optimization of the LSID-criterion and the D-optimal criterion. Three selected individuals are indicated by arrows.

In Figures 8.13 and 8.14 the optimal measurement times for each of the individuals of the Pareto front are shown, starting from the individual with the lowest D-optimal and LSID-criterion value (individual 1) to the individual with the highest D-optimal and LSID-criterion value (individual 100). These figures show which measurement regions are associated with D-optimal criterion values with low sensitivity towards parameter changes and how these regions shift with increasing D-optimal and LSID-criterion values. From Figure 8.13 it can also be concluded that the substrate measurement point just after the start of the additional feeding ($t = 6 h$) is associated with all designs corresponding to low LSID values. This can be explained by the fact

that the contribution of this measurement point to the (low) information content for these designs cannot be influenced by parameter changes (low LSID values) since the substrate concentration at this point is completely determined by the experimental setup (i.e. t_f , S_f and Q_f). On the other hand, it can be seen that for designs with a high information content (high D-optimal criterion value) the measurement point just before the depletion of substrate ($t = 5.3$ h) is also always present. The reason why this point has a high information content should be attributed to the use of relative errors which makes the weight at this point extremely high due to the low substrate concentration (see Figure 1.16). The exact point of substrate depletion, however, is very sensitive to parameter changes (high LSID value) since these changes would influence the substrate concentration profile and thus also the time instant of substrate depletion.

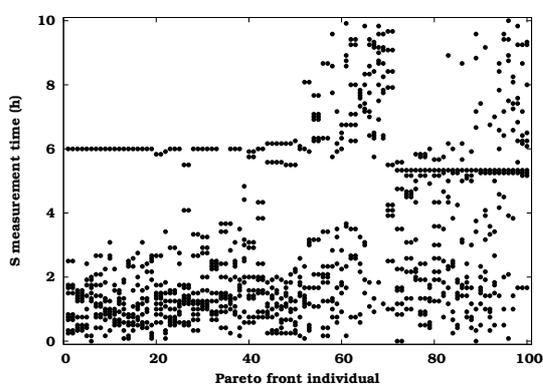


Figure 8.13: Optimal substrate measurement times associated with individuals of the Pareto optimal front for a multi-objective optimization of the D-optimal criterion and the LSID-criterion.

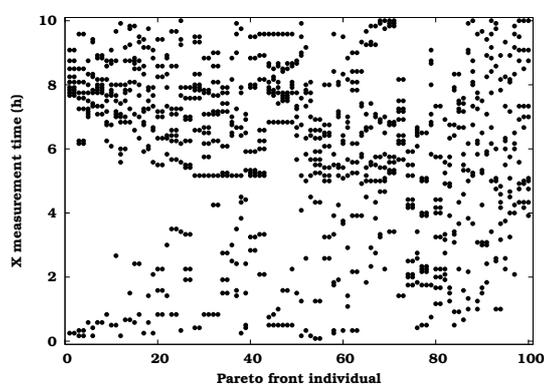


Figure 8.14: Optimal biomass measurement times associated with individuals of the Pareto optimal front for a multi-objective optimization of the D-optimal criterion and the LSID-criterion.

From the 100 individuals of the Pareto front, 3 individuals were selected for a more detailed analysis: individuals 1, 50 and 100. The designs associated with these individuals are also shown in Figures 8.10 and 8.11. Notice that designs associated with individuals 1 and 50 are unlike any of the other designs. Substrate and biomass is primarily measured in regions where no measurements are placed for the other designs. The design associated with individual 100 obviously shows similar characteristics as the classical D-optimal design. Looking at the result of the model calibrations based on the optimal measurement times of the selected individuals (Table 8.4), it can be seen that the parameters are estimated reasonably well. Even individual 1 resulted in accurate parameter estimates despite its low D-optimal criterion value. An explanation for this could be the fact that the predicted low information content is compensated by the low sensitivity of the D-optimal criterion values to parameter changes.

Global sensitivity index approach

According to the algorithm described in Section 8.3.4.3, a GSID-optimal experiment was determined. The optimal measurement times associated with this design are shown in Figures 8.10 and 8.11. This design is unlike the other robust designs in the sense that measurements appear to be desired in regions where no measurements are placed for the other designs. Using this design, the model was recalibrated, which did not lead to accurate parameter estimates. The obtained design corresponds with an extremely low information content (D-optimal criterion value: $1.00+11$) which is coupled with a low sensitivity to parameter changes.

Computational comparison

So far, nothing has been mentioned regarding the computational requirements for the different robust experimental design approaches. Each of the robust experimental design approaches will now be considered with respect to the amount of sensitivity analyses that need to be performed. An overview of these results is shown in Table 8.5. Performing a sensitivity analysis is probably the most important factor influencing the speed of the experimental design. For the non-linear dynamic models discussed in this thesis, the sensitivity analysis has to be approximated numerically (see also Chapter 3) and can take a long time to calculate especially when a large number of parameters are involved.

The classical D-optimal design calculated in this chapter required only 1 sensitivity analysis to be performed, which is then used to find the optimal measurement times (inner loop optimization). When dealing with robust experimental design, generally different sets of model parameters are to be considered causing the number of required sensitivity analyses to increase drastically.

For the expected value approach, a genetic algorithm is responsible for the optimization of the measurement degrees of freedom. For this case study, the algorithm converged after evaluating approximately 5000 individuals (different designs). Each of these designs required 100 FIM evaluations with each evaluation corresponding to a new parameter set (Monte Carlo shot) requiring 1 sensitivity analysis.

For MMD-optimal design, two nested GA optimizations are used. The outer loop required the evaluation of approximately 3000 individuals (different designs) before it converged. For each of the proposed individuals another GA was used to optimize the model parameters. This required on average 1000 FIM evaluations (with different parameter sets) each requiring 1 sensitivity analysis.

The multi-objective GA used for the local sensitivity index approach required the evaluation of 80000 individuals (different designs). Each of these individuals required a local sensitivity analysis of the D-optimal criterion value, requiring approximately 28 FIM evaluations (approximately 7 perturbation factor trials for each of the 4 parameters for the Richardson's extrapolation technique).

Finally, the global sensitivity index approach also used two nested GAs. The outer GA converged after 4000 evaluations (different designs) whereas the inner GA converged for each of these individuals after approximately 500 sensitivity index evaluations (for different parameter sets). Each of the sensitivity index evaluations required 2 FIM evaluations and thus also 2 sensitivity analyses (central difference with fixed perturbation factors). The correct perturbation factors were determined from a manual search in different locations of the parameter domain. Richardson's extrapolation was not used in this case due to computational limitations.

As can be seen from the above analysis of Table 8.5, robust experimental design requires an enormous number of FIM evaluations. Therefore, applications of these techniques will remain limited to relatively simple models until the sensitivity analysis speed can be improved and more efficient global optimization techniques are used or developed. Even the use of local optimizers can be considered. However, the known problems related to these optimizers would certainly influence the results of the robust design. Another solution to this problem might be the use of some form of distributed computing where all required calculations are split up over several machines and performed in parallel, dramatically reducing the required calculation time. An example of such system is WDVE (WEST Distributed Virtual Experimentation) (Claeys et al., 2004).

Table 8.5: Overview of the number of sensitivity analyses required for each of the robust experimental design approaches.

Criterion	Required number of sensitivity analyses
D	1.00+00
ED	5.52+05
EID	4.17+05
API	4.69+05
MMD	2.95+06
LSID ₁₋₁₀₀	2.32+06
GSID	4.06+06

8.4 Conclusions

The main topic of this chapter was a well-known problem of optimal experimental design for non-linear models: uncertainty related to the model parameters. An outline of existing robust experimental design techniques was given followed by a practical application of these techniques to a fed-batch bio-reactor simulation study. Beside the already known robust design techniques, a new technique was proposed based on the sensitivity of the FIM design criteria with respect to parameter changes. From the case

study it was found that EID- and MMD-optimal designs produced the best results. The newly suggested approach unfortunately did not result in useful results. This was due to the strong correlation between the sensitivity index criterion and the information content (D-optimal design). However, it should be studied if this would also be the case for other FIM design criteria. It was also found that robust experimental design is very computationally demanding. From all presented techniques (not considering discrete and sequential design) the expected value approach required the least amount of calculations. This computational requirement limits the possibility of applying these techniques to larger, more complex models (DAE or ODE models) unless the most computationally demanding parts of these algorithms can be made faster. Firstly, the time required to perform a sensitivity analysis should be reduced as much as possible. Secondly, more efficient global optimizers might be used. Even the use of local optimizers could be considered.

From the schematic overviews of all implemented algorithms it is clear that the optimization of the experimental degrees of freedom is always performed in the outer optimization loop. Therefore, extending the number of experimental degrees of freedom to be optimized should have considerably less impact on the computational demand than when more model parameters would be considered in the design.

Chapter 9

General conclusions and future perspectives

9.1 Goal of this chapter

The introductory chapter of this PhD thesis (Chapter 1) showed that optimal experimental design is a valuable technique to improve the parameter estimation quality of bioprocess models. However, the non-linear and dynamic nature of these models makes the application of optimal experimental design techniques far from straightforward. Based on the known issues related to non-linear dynamic optimal experimental design, a problem statement was formulated in Section 1.4. The remainder of this chapter will therefore deal with each of the identified problems and how a solution to these problems was developed. Beside discussing the main conclusions, much attention will also be paid to future perspectives on how the proposed techniques may be improved or extended further in order to increase the usefulness and applicability of optimal experimental design to non-linear dynamic models.

9.2 Problem 1: Software availability

Problem description

Optimal experimental design involves a series of complicated steps including parameter estimation, sensitivity analysis, non-linear optimization. However, no simulation package exists which combines all these methods into a single module. Beside this, most applications of optimal experimental

design up till now have been case specific, frequently relying on analytical solutions, resulting in a loss of generality.

Solutions

A solution to these problems was developed by programming extensions to an existing modelling and simulation package, WEST (Vanhooren et al., 2003). This modified simulation software, called EAST, is able to solve optimal experimental design problems in a general way and is applicable to every model available within WEST. In order to maintain generality, much emphasis was put on numerical techniques rather than on analytical techniques. The newly developed software has proven to be very flexible and is able to tackle many different types of optimal experimental design problems like single- and multi-objective optimal experimental design, but also more complicated tasks such as automatic and robust experimental design. Specific problems and perspectives related to each of these methods will be discussed in separate sections.

Perspectives

The modular structure of EAST allows to create extensions by reusing existing components. This reduces the effort to develop new extensions and allows for a rapid implementation of new ideas. Potential new developments are situated in the field of optimal experimental design for model selection and the combined optimal experimental design problem for model selection and parameter estimation. Beside this, development or implementation of more performant numerical solvers (e.g. optimizers) should also be considered.

9.3 Problem 2: Local sensitivity analysis

Problem description

Local sensitivity functions are one of the two components of the Fisher Information Matrix. Many methods exist to calculate these local sensitivity functions but only a few are applicable when the model equations are not directly accessible. In this case the finite difference technique is used most often. However, the practical application of this technique to a non-linear dynamic model poses a problem: a correct perturbation factor needs to be chosen in order to prevent numerical errors or errors related to the non-linearity of the model to influence the sensitivity analysis calculations.

Solutions

In Chapter 3 a semi-automatic method to detect wrongly calculated sensitivity functions was developed. This method is based on the quantification of the difference between two sensitivity functions calculated with opposite

perturbation factors. Several criteria were proposed to quantify this difference from which the SRE (sum of relative errors) criterion was found to be most useful. An important conclusion of this research was that perturbation factors are case dependent: they depend on the parameters and to a lesser extent on the variables. It was also found that the settings of the numerical integrator used to solve the model equations has a large influence on the optimal perturbation factors.

In order to eliminate the choice of the perturbation factors a technique based on complex-number calculations (the complex-step derivative approximation method) was also applied. It only requires slight modifications of the original model equation code. Using this technique, the user is no longer required to specify perturbation factors.

Perspectives

The development of the method to detect incorrect sensitivity functions is certainly a big step forward. It provides the user with a quantitative tool to evaluate the quality of the calculated sensitivity functions. However, automating this technique completely (removing all user interaction) would require it to be incorporated into an optimization algorithm. Although feasible, this approach would increase the number of required simulations significantly. Therefore, techniques like the complex-step derivative approximation method appear very valuable. However, a computational comparison between the finite difference method and the complex-step derivative approximation has shown that the latter was much more computationally demanding than expected, especially for large models. The reason for this has not yet been pinpointed and further research is certainly required in order to make this technique competitive with other sensitivity analysis techniques. A reason for this slow-down could be that potentially inefficient C++ complex algebra was used in this case whereas usually Fortran implementations of this technique are encountered in literature.

Other techniques to calculate sensitivity functions, like automatic differentiation have emerged in the last years and are certainly candidates for further research too, particularly because these techniques have been developed with automation in mind.

9.4 Problem 3: Parameter identifiability

Problem description

In order to quantify the quality of a parameter estimation exercise, the parameter estimation error covariance matrix should be available. This can be calculated based on the Hessian matrix or the Fisher Information Matrix (FIM). Calculating the Hessian matrix requires the evaluation of the second derivatives of the objective function with respect to the model parameters. These calculations have to be performed numerically and are quite computationally intensive.

Performing optimal experimental design for complex models containing many parameters requires an a priori choice to be made about the number of parameters that will be taken into account for the design. Finding optimal experiments that would result in data of such high quality that all parameters can be estimated is utopian, especially because some parameter combinations might not even be structurally identifiable. Therefore, techniques are needed to select identifiable parameter sets based on existing data.

Solutions

Several numerical techniques to calculate the Hessian matrix of a parameter estimation problem were studied and compared in Chapter 4. Especially the finite difference approximation using Richardson's extrapolation provided good results although it was found to be very computationally demanding. In order to improve the calculation speed of this technique, the complex-step approximation method was utilized to calculate the Jacobian matrix (first derivatives of the objective function with respect to the parameters).

An analysis of the relationship between the Hessian matrix and the FIM revealed that the difference between both matrices can conveniently be used as an indicator for model inadequacies, local minima and/or non-normally distributed residuals.

Making use of the relationship between the FIM and the Hessian, a new method, based on FIM related properties, was proposed to select identifiable parameter subsets. The newly proposed technique only requires the user to specify the maximum allowable correlation between parameters. Other than that, no user-interaction is needed. Although the use of the FIM D-optimal criterion value to compare parameter sets of different dimensions may be questioned, it did prove useful for the presented case study.

Perspectives

A topic which still requires investigation is the selection of identifiable parameter sets based on an available data set. A general technique which does not require user interaction and is applicable to large models is still not available yet. The application of the current techniques is restricted to the evaluation of the identifiability of approximately 20 to 30 parameters in order to avoid a combinatorial explosion. If the model consists of more parameters, a pre-selection needs to be made by the user, potentially removing some identifiable parameters. In order to avoid this pre-selection, it should be possible to use knowledge about the identifiability of small parameter sets to better direct the search for parameter subsets of larger size without having to analyse all possible combinations.

Selecting the parameter subset to focus upon in the experimental design procedure is also a topic for further research. An optimal experiment designed to improve the estimation quality of a certain set of parameters might also be useful to estimate more than these selected parameters. Surely,

experiments which provide data with a higher information content will increase the probability that other parameters will become identifiable. Therefore, it might be useful to include a search for the identifiable parameter subsets at the start of each iteration of the experimental design procedure. Along these lines, it should also be possible to include a model selection procedure in the optimal experimental design scheme. An extended optimal experimental design procedure illustrating this idea is shown in Figure 9.1.

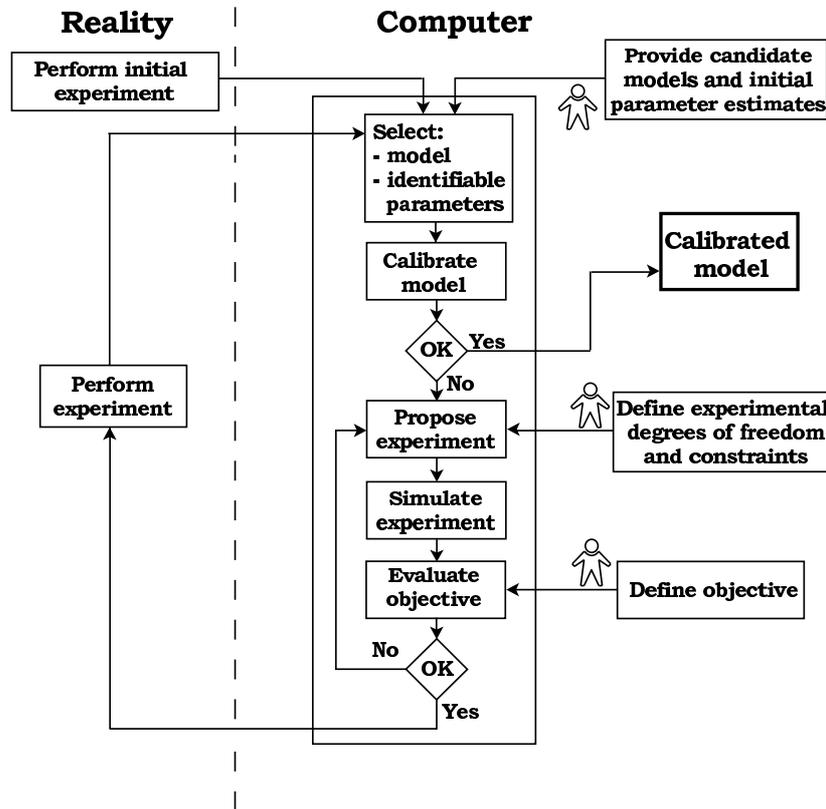


Figure 9.1: Schematic overview of the extended optimal experimental design procedure. Places where user interaction is required are indicated by human symbols (adapted from Dochain and Vanrolleghem, 2001).

If these two additional features could be added to the experimental design procedure it would free the user from the often difficult task of providing the most applicable model for the system under study and an identifiable parameter subset of this model. Instead, the user would then only need to provide candidate models and a design would be constructed which would be useful for selecting the most appropriate model and calibrating the parameters of this model (probably going through several iterations).

9.5 Problem 4: Global optimization in optimal experimental design

Problem description

In Section 1.3 it was illustrated that the optimization problem related to optimal experimental design can be very complex, especially when many experimental degrees of freedom and constraints are considered. The FIM criteria objective surfaces for a simple case of two experimental degrees of freedom already showed that many local minima can be present (Figures 1.20, 1.21, 1.22 and 1.23). When using classical non-linear optimization algorithms there is a high probability that only a local minimum will be detected. Therefore, a more global approach to these optimization problems is required.

Solutions

By analysing the optimal experimental design optimization problem in detail, it was found that the optimization could be split up into two parts: an inner and an outer loop for the optimization of measurement experimental degrees of freedom and manipulation experimental degrees of freedom respectively. In this way the complex experimental design optimization problem can be solved faster in many situations since no repeated (time consuming) sensitivity analysis is needed to optimize the measurement experimental degrees of freedom.

In order to increase the probability of finding the global optimum for an optimal experimental design problem, the use of a real-coded genetic algorithm (GA) was investigated. It was clearly shown that the GA outperformed classical optimization techniques and was able to locate the global optimum in a consistent way. However, the drawback of using these algorithms was that approximately 100 times more objective function evaluations were needed compared to the classical optimization techniques.

By using a real-coded GA it became possible to solve complex optimization problems: large search spaces, many optimization variables and requiring high precision results. Beside this, real-coded GAs also allow to solve optimization problems characterized by combinations of continuous and discrete optimization variables (the latter only taking discrete values for the optimization variable) without having to rely on different genetic operators for each of the types of optimization variables.

Perspectives

The large number of required objective function evaluations that are needed to find the global minimum can probably be reduced by better tuning the GA settings and operators. Using adaptive methods to control these GA parameters during the course of the optimization would be a valuable addition to the performed research. The use of hybrid optimization algorithms, i.e. a combination of a global and a local optimizer would also be useful in this

context. The GA would then only be used to find regions of interest within which a faster converging classical local optimization algorithm is used to quickly determine the minimum in that region and thus more quickly find the global minimum.

Another future extension which would make the results of the optimal experimental design procedure more practically useful would be a more advanced constraint handling. Currently, constraints are only defined as a lower and upper bound on each experimental degree of freedom. More complex situations like dependencies between experimental degrees of freedom and constraints based on certain system variables would make the experimental design procedure, as it is implemented in EAST, more flexible and allow more realistic and practically useful experiments to be designed.

9.6 Problem 5: Multi-objective optimization in experimental design

Problem description

During the illustration of optimal experimental design in Chapter 1, it became apparent that the FIM optimal design criteria are often conflicting and that an experiment which is optimal for a certain design criterion is far from optimal for another design criterion.

Solution

Finding experimental designs which are optimal with respect to several criteria can be accomplished by performing a multi-objective optimization of the experimental degrees of freedom. In Chapter 6 a multi-objective GA was used to solve optimal experimental design problems using several FIM criteria and also incorporating experimental costs. In contrast to other multi-objective techniques, these types of algorithms do not require weights or other steering parameters to be specified that are needed to turn the multi-objective problem into a single-objective problem. Using the obtained Pareto front (a set of solutions for which it cannot be said that one solution is better in all objectives compared to the other solutions) a user can make a sound choice with respect to which experiment will be performed based on a trade-off between the different objectives.

Perspectives

The use of multi-objective GAs has great potential for designing realistic measurement campaigns. Incorporating more objectives like system performance, model prediction uncertainty and other more practice oriented objectives could also prove beneficial. Another application of multi-objective experimental design might be the design of experiments for simultaneous model selection and parameter estimation without having to reduce this problem to a combined-objective optimization problem.

9.7 Problem 6: Automation of experimental design

Problem description

The classical iterative optimal experimental design procedure presented in Section 1.3.3 involves different, rather complicated mathematical and practical steps in order to obtain the calibrated model. These steps require much user interaction and expert knowledge. This is one of the reasons why optimal experimental design has not been used so frequently for bioprocess models.

Solutions

In Chapter 7 an automatic optimal experimental design procedure was proposed. In this procedure, user interaction and expert knowledge is only required at the beginning and all subsequent steps of the iterative search for the best parameter estimates can be performed automatically. These steps include (1) finding the optimal experiment, (2) performing the experiments in practice and (3) recalibrating the model. The proposed procedure was successfully applied to calibrate a one-step nitrification model using a respirometric experiment.

Perspectives

With these developments, the dream of starting an automated optimal experimental design procedure in the evening and having a calibrated model available the next morning has become a bit more reality. Future perspectives in this field should certainly be oriented towards applications on other experimental setups and potentially also on full-scale installations. In order for this to succeed, integration of model selection and an automated identifiability analysis in the proposed procedure should certainly be considered since this would make the whole procedure more robust.

Attention should also be focused on the development of a systematic approach for automatic calibration of complex bioprocess models since at the moment these are still mainly calibrated using expert knowledge and trial-and-error.

9.8 Problem 7: Robust experimental design

Problem description

An important aspect of optimal experimental design for parameter estimation for non-linear models is the dependency of the design on the model parameters. This is caused by the fact that the FIM is calculated from sensitivity functions (partial derivatives of the model variables to the parameters) which, for non-linear models, are still function of the model parameters. This causes designs based on the FIM to be only locally valid (for

the particular values of the parameters for which the experimental design is performed) which might lead to unsatisfactory results when the used model parameters are too different from the “real” parameter values.

Solutions

Several known techniques for robust experimental design were studied and implemented in EAST together with a newly proposed technique based on the sensitivity of the FIM criteria to parameter changes. It was shown that the designs based on known robust experimental design criteria produced the best results. It was also found that robust experimental design is very computationally demanding. For the “simple” case presented in Chapter 8, where 4 parameters and 20 experimental degrees of freedom were considered, approximately 4 million sensitivity analyses were required, representing several days of calculations on a P4 3 GHz system. This computational demand is caused by the complex structure of some of the algorithms. For example, the newly proposed robust design criterion requires a GA optimization to be performed in the domain of the experimental degrees of freedom. Each of the individuals from this optimization is then used as a basis for a GA optimization in the parameter domain for which each individual requires a calculation of a sensitivity of a design criterion (using Richardson’s extrapolation). The design criterion in turn is calculated based on the evaluation of a FIM which requires a sensitivity analysis of certain model variables with respect to a selected number of parameters. For each of these sensitivity analyses a number of model simulations are required in case a finite difference approximation is used.

Perspectives

At this moment, the computational requirements limit the application of these techniques to small, simple models with a limited set of experimental degrees of freedom. This could be solved by making the computationally demanding parts of these algorithms run faster or more efficient. Firstly, the time required to perform a sensitivity analysis should be reduced as much as possible. Secondly, more efficient global optimizers might be used and/or developed.

9.9 Final thoughts

Without question, optimal experimental design is a very useful and needed mathematical tool if experimental data with a high information content is to be obtained from complex systems like bioprocesses. However this valuable data comes at a certain price.

First of all, it has to be mentioned that experimental design involves a lot of complicated mathematical steps involving many numerical techniques.

It is often difficult to find and stay on the border between numerical inaccuracies and errors related to the non-linear characteristics of the model. Therefore, it will always be needed to verify the accuracy of each intermediate step of the experimental design procedure. Without this quality check, errors would propagate and be magnified into the final solution.

Secondly, it should never be forgotten what the assumptions are that lie at the basis of the optimal experimental design procedure. Examples of these assumptions could be the model that is used as the basis for the experimental design or the error characteristics of the measurements (which are mostly assumed to be normally distributed). Without verifying these assumptions, it cannot be expected that an optimal experimental design will indeed provide data with the required information content.

Thirdly, computational power is a finite property. This work has shown that the application of optimal experimental design to complex bioprocess models can be very computationally demanding, especially when global optimization and robust design is considered. However, Moore's law (Moore, 1965) which predicts the doubling of processor transistors every couple of years, still holds today. Computational power is (ever) increasing and methods which seem unfeasible today might become standard tomorrow. However, model complexity also increases steadily. Therefore, techniques that would improve calculation speed like distributed computing (Claeys et al., 2004) or grid computing, which distribute the required computations over several machines in parallel, will always be useful and needed, especially since the problem at hand is perfectly suited to be split up into several smaller computational entities. For example, the different simulations required to perform a finite difference sensitivity analysis can be executed on different machines and so are the objective function evaluations of different individuals of a GA population.

Some of the methods and software developed in this work would also be of use in fields other than optimal experimental design.

Sensitivity analysis is certainly not only useful in the field of optimal experimental design. Many derivative-based optimization algorithms also need to compute the first derivatives of the objective function. When these derivatives cannot be supplied by the user, numerical techniques need to be used. Very often finite difference approximations are used with fixed perturbation factors. More intelligent or automatic perturbation factor selection techniques would certainly be useful in this context.

Solving systems of ordinary differential equations (ODEs) is another field where finite differences (with fixed perturbation factors) are still often used. Many solvers use the first derivatives of the state variables (Jacobian) in order to integrate the differential equations. Again, these derivatives are often numerically approximated by finite differences when they are not provided analytically, potentially introducing errors by a non-optimal choice of the perturbation factors.

The application of a GA as a global optimizer has already been successfully applied in other domains. However, for the calibration of non-linear dy-

dynamic bioprocess models it is not yet often used even though it is an ideal candidate to solve the difficult (many local minima) optimization problems that often present themselves in this field.

Finally, it can be concluded that optimal experimental design is an essential tool in order to develop models for better understanding and estimation of complex dynamic non-linear systems like bioprocesses.

Bibliography

- Abusam, A., Keesman, K.J., van Straten, G., Spanjers, H. and Meinema, K. (2001), Parameter estimation procedure for complex non-linear systems: calibration of ASM No. 1 for N-removal in a full-scale oxidation ditch, *Water Science and Technology*, 43(7), 357–365.
- Asprey, S.P. and Macchietto, S. (2002), Designing robust optimal dynamic experiments, *Journal of Process Control*, 12(4), 545–556.
- Atherton, R.W., Schainker, R.B. and Ducot, E.R. (1975), Statistical sensitivity analysis of models for chemical-kinetics, *AIChE Journal*, 21(3), 441–448.
- Atkinson, A.C. (2003), Horwitz's rule, transforming both sides and the design of experiments for mechanistic models, *Journal Of The Royal Statistical Society Series C - Applied Statistics*, 52(3), 261–278.
- Atkinson, A.C. and Bogacka, B. (2002), Compound and other optimum designs for systems of nonlinear differential equations arising in chemical kinetics, *Chemometrics and Intelligent Laboratory Systems*, 61(1-2), 17–33.
- Baetens, D., Hosten, L.H., Petersen, B., Van Volssem, S. and Vanrolleghem, P.A. (2000), Optimal experimental design for the calibration of models of phosphorous removing activated sludge systems, *Journal A*, 41(3), 65–73.
- Baltes, M., Schneider, R., Sturm, C. and Reuss, M. (1994), Optimal experimental design for parameter estimation in unstructured growth models, *Biotechnology Progress*, 10(5), 480–488.
- Bastin, G. and Dochain, D. (1990), *On-line estimation and adaptive control of bioreactors*, Elsevier, Amsterdam, The Netherlands.
- Batstone, D.J., Keller, J., Angelidaki, I., Kalyuzhnyi, S.V., Pavlostathis, S.G., Rozzi, A., Sanders, W.T.M., Siegrist, H. and Vavilin, V.A. (2002),

- Anaerobic Digestion Model No. 1 (ADM1)*, IWA scientific and technical report no. 13, IWA Publishing, London, UK.
- Bauer, I., Bock, H.G., Korkel, S. and Schloder, J.P. (2000), Numerical methods for optimum experimental design in dae systems, *Journal of Computational and Applied Mathematics*, 120(1-2), 1–25.
- Beck, M.B. (1987), Water quality modelling: a review of the analysis of uncertainty, *Water Resources Research*, 23(8), 1393–1442.
- Berkholz, R., Rohlig, D. and Guthke, R. (2000), Data and knowledge based experimental design for fermentation process optimization, *Enzyme and Microbial Technology*, 27(10), 784–788.
- Bernaerts, K., Servaes, R.D., Kooyman, S. and Van Impe, J.F. (2000a), Iterative optimal experimental design for estimation of microbial growth kinetics as function of temperature., in: Dochain, D. and Perrier, M., editors, *International Conference on Computer Applications in Biotechnology (CAB8)*, 19–24.
- Bernaerts, K., Versyck, K.J. and Van Impe, J.F. (2000b), On the design of optimal dynamic experiments for parameter estimation of a ratkowsky-type growth kinetics at suboptimal temperatures, *International Journal of Food Microbiology*, 54(1-2), 27–38.
- Bernard, O., Hadj-Sadok, Z., Dochain, D., Genovesi, A. and Steyer, J.P. (2001), Dynamical model development and parameter identification for an anaerobic wastewater treatment process, *Biotechnology and Bioengineering*, 75(4), 424–438.
- Beven, K. and Binley, A. (1992), The future of distributed models: model calibration and uncertainty prediction, *Hydrological Processes*, 6(3), 279–298.
- Blanco, A., Delgado, M. and Pegalajar, M.C. (2001), A real-coded genetic algorithm for training recurrent neural networks, *Neural Networks*, 14(1), 93–105.
- Box, G.E.P. and Lucas, H.L. (1959), Design of experiments in non-linear situations, *Biometrika*, 46, 77–90.
- Boyer, D.O., Martinez, C.H. and Perez, J.M. (2001), Genetic algorithm with crossover based on confidence intervals as an alternative to least squares estimation for nonlinear models, in: *Proceedings of the 4th Metaheuristic International Conference*, 343–347.
- Brent, R.P. (1973), *Algorithms for Minimization without Derivatives*, Prentice-Hall, Englewood Cliffs, USA.

- Brun, R., Kuhni, M., Siegrist, H., Gujer, W. and Reichert, P. (2002), Practical identifiability of asm2d parameters - systematic selection and tuning of parameter subsets, *Water Research*, 36(16), 4113–4127.
- Brun, R., Reichert, P. and Kunsch, H.R. (2001), Practical identifiability analysis of large environmental simulation models, *Water Resources Research*, 37(4), 1015–1030.
- Burg, C.O.E. and Newman III, J.C. (2003), Computationally efficient, numerically exact design space derivatives via the complex Taylor's series expansion method, *Computers and Fluids*, 32(3), 373–383.
- Butuk, N. and Pemba, J.P. (2003), Computing CHEMKIN sensitivities using complex variable, *Journal of Engineering for Gas Turbine and Power - Transactions of the ASME*, 125(3), 854–858.
- Chen, B.H. and Asprey, S.P. (2003), On the design of optimally informative dynamic experiments for model discrimination in multiresponse nonlinear situations, *Industrial and Engineering Chemistry Research*, 42(7), 1379–1390.
- Chung, S.H., Ma, D.L. and Braatz, R.D. (2000), Optimal model-based experimental design in batch crystallization, *Chemometrics and Intelligent Laboratory Systems*, 50(1), 83–90.
- Claeys, F., Chtepen, M., Benedetti, L., Dhoedt, B. and Vanrolleghem, P.A. (2004), Distributed virtual experiments in water quality management, in: *Proceedings of the 6th International Symposium on System Analysis and Integrated Assessment in Water Management (WATERMATEX 2004)*, 485–493.
- Cochran, W.G. (1973), Experiments for nonlinear functions, *Journal of the American Statistical Association*, 68(344), 771–781.
- Coello, C.A. (2000), An updated survey of ga-based multiobjective optimization techniques, *ACM Computing Surveys*, 32(2), 109 – 143.
- Copp, J.B. (2001), *The COST Simulation Benchmark: Description and Simulator Manual*, Office for Official Publications of the European Community, Luxembourg, Luxembourg.
- Cunha, L.M. and Oliveira, F.A.R. (2000), Optimal experimental design for estimating the kinetic parameters of processes described by the first-order arrhenius model under linearly increasing temperature profiles, *Journal of Food Engineering*, 46(1), 53–60.
- Cunha, L.M., Oliveira, F.A.R., Brandao, T.R.S. and Oliveira, J.C. (1997), Optimal experimental design for estimating the kinetic parameters of the bigelow model, *Journal of Food Engineering*, 33(1-2), 111–128.

- Davis, L. (1991), *Handbook on Genetic Algorithms*, Van Nostrand Reinhold, New York, USA.
- Deb, K. (2001), *Multi-objective optimization using evolutionary algorithms*, Wiley, Chichester, UK.
- Deksissa, T., De Pauw, D. and Vanrolleghem, P.A. (2004), Dynamic in-stream fate modelling of xenobiotic organic compounds: A case study of Linear Alkylbenzene Sulfonates in the Lambro river, Italy, *Environmental Toxicology and Chemistry*, 23(9), 2267–2278.
- Dette, H., Melas, V.B., Pepelyshev, A. and Strigul, N. (2003), Efficient design of experiments in the monod model, *Journal of the Royal Statistical Society Series B - Statistical Methodology*, 65, 725–742.
- Dickinson, R.P. and Gelinas, R.J. (1976), Sensitivity analysis of ordinary differential equation systems - a direct method, *Journal of Computational Physics*, 21(2), 123–143.
- Diez, M.D., Fjeld, M., Andersen, E. and Lie, B. (2004), Validation of a compartmental population balance model of an industrial leaching process: the Silgrain process, in: *Proceedings of the 2nd International Conference on Population Balance Modelling*, on CDROM.
- Dochain, D. and Vanrolleghem, P.A. (2001), *Dynamical Modelling and Estimation in Wastewater Treatment Processes*, IWA Publishing, London, UK.
- Dougherty, E.P., Hwang, J.T. and Rabitz, H. (1979), Further developments and applications of the green's function method of sensitivity analysis in chemical kinetics, *Journal of Chemical Physics*, 71(4), 1794–1808.
- Dougherty, E.P. and Rabitz, H. (1980), Computational kinetics and sensitivity analysis of hydrogen-oxygen combustion, *Journal of Chemical Physics*, 72(12), 6571–6586.
- Dunker, A.M. (1981), Efficient calculation of sensitivity coefficients for complex atmospheric models, *Atmospheric Environment*, 15(7), 1155–1161.
- Dunker, A.M. (1984), The decoupled direct method for calculating sensitivity coefficients in chemical kinetics, *Journal of Chemical Physics*, 81(5), 2385–2393.
- Emery, A.F. and Fadale, T.D. (1996), Design of experiments using uncertainty information, *Journal of Heat Transfer-Transactions of the ASME*, 118(3), 532–538.
- Eshelman, L.J. and Schaffer, J.D. (1993), *Foundation of Genetic Algorithms 2*, chapter Real-coded genetic algorithms and interval schemata, 187–202, Morgan Kaufmann, San Mateo, USA.

- Espie, D. and Macchietto, S. (1989), The optimal design of dynamic experiments, *AIChE Journal*, 35(2), 223–229.
- Fadale, T.D., Nenarokomov, A.V. and Emery, A.F. (1995), Two approaches to optimal sensor locations, *Journal of Heat Transfer-Transactions of the ASME*, 117(2), 373–379.
- Faller, D., Klingmuller, U. and Timmer, J. (2003), Simulation methods for optimal experimental design in systems biology, *Simulation*, 79(12), 717–725.
- Fisher, R.A. (1949), *The Design of Experiments*, Hafner Publishing, New York, USA, 5th edition.
- Fletcher, R. (1987), *Practical Methods of Optimization*, John Wiley & Sons, Chichester, UK.
- Fonseca, C.M. and Fleming, P.J. (1995), An overview of evolutionary algorithms in multiobjective optimization, *Evolutionary Computation*, 3(1), 1–16.
- Foracchia, M., Hooker, A., Vicini, P. and Ruggeri, A. (2004), POPED, a software for optimal experiment design in population kinetics, *Computer Methods and Programs in Biomedicine*, 74(1), 26–46.
- Ford, I., Titterington, D.M. and Kitsos, C.P. (1989), Recent advances in non-linear experimental design, *Technometrics*, 31(1), 49–60.
- Foubert, I., Dewettinck, K. and Vanrolleghem, P.A. (2003), Modelling of the crystallization kinetics of fats, *Trends in Food Science and Technology*, 14(3), 79–92.
- Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Massachusetts, USA.
- Goldberg, D.E. (1991), Real-coded genetic algorithms, virtual alphabets, and blocking, *Complex Systems*, 5, 139–167.
- Goldberg, D.E. and Richardson, J. (1987), Genetic algorithms with sharing for multimodal function optimization, in: *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, 41–49.
- Grady, C.P.L., Smets, B.F. and Barbeau, D.S. (1996), Variability in kinetic parameter estimates: A review of possible causes and a proposed terminology, *Water Research*, 30(3), 742–748.
- Grijspeerdt, K. and Vanrolleghem, P. (1999), Estimating the parameters of the baranyi model for bacterial growth, *Food Microbiology*, 16(6), 593–605.
- Hao, X.D., Heijnen, J.J. and Van Loosdrecht, M.C.M. (2002), Model-based evaluation of temperature and inflow variations on a partial nitrification-ANAMMOX biofilm process, *Water Research*, 36(19), 4839–4849.

- Hariri, A., Boivin, P., Fick, M. and Fournier, F. (2004), Optimisation of steeping program in malt industry, in: *Proceedings of the 2nd International IWA Conference on Automation in Water Quality Monitoring (AutMoNet 2004)*, on CDROM, Vienna, Austria.
- Henze, M., Grady, C.P.L., Gujer, W., Marais, G.v.R. and Matsuo, T. (1986), *Activated sludge model no.1, IWA scientific and technical report no.1*, IWA Publishing, London, UK.
- Henze, M., Gujer, W., Matsuo, T and Van Loosdrecht, M. (2000), *Activated sludge models ASM1, ASM2, ASM2d and ASM3, IWA scientific and technical report no.9*, IWA publishing, London, UK.
- Henze, M., Gujer, W., Mino, T., Matsuo, T., Wentzel, M.C., Marais, G.V.R. and Van Loosdrecht, M.C.M. (1999), Activated sludge model no. 2d, *Water Science and Technology*, 39(1), 165–182.
- Herrera, F., Lozano, M. and Sanchez, A.M. (2003), A taxonomy for the crossover operator for real-coded genetic algorithms: an experimental study, *International Journal of Intelligent Systems*, 18(3), 309–338.
- Herrera, F., Lozano, M. and Verdegay, J.L. (1994), The use of fuzzy connectives to design real-coded genetic algorithms, *Mathware and Soft Computing*, 1(3), 239–251.
- Herrera, F., Lozano, M. and Verdegay, J.L. (1998), Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis, *Artificial Intelligence Review*, 12(4), 265–319.
- Hindmarsh, A.C. and Petzold, L.R. (1995), Algorithms and software for ordinary differential equations and differential-algebraic equations, *Computers in Physics*, 9, 148–155.
- Hoffmann, F., Posten, C. and Rinas, U. (2001), Kinetic model of in vivo folding and inclusion body formation in recombinant escherichia coli, *Biotechnology and Bioengineering*, 72(3), 315–322.
- Holland, J.H. (1975), *Adaption in Natural and Artificial Systems*, MIT Press, Michigan, USA.
- Holmberg, A. (1982), On the practical identifiability of microbial growth models incorporating Michaelis-Menten type nonlinearities, *Mathematical Biosciences*, 62(1), 23–43.
- Hwang, D., Byun, D.W. and Odman, M.T. (1997), An automatic differentiation technique for sensitivity analysis of numerical advection schemes in air quality models, *Atmospheric Environment*, 31(6), 879–888.
- Hwang, J.T. (1983), Sensitivity analysis in chemical kinetics by the method of polynomial approximations, *International Journal of Chemical Kinetics*, 15, 959–987.

-
- Hwang, J.T. (1985), A computational algorithm for the polynomial approximation method of sensitivity analysis in chemical kinetics, *Journal of the Chinese Chemical Society*, 32(3), 253–261.
- Hwang, J.T., Dougherty, E.P., Rabitz, S. and Rabitz, H. (1978), The Green's function method of sensitivity analysis in chemical kinetics, *Journal of Chemical Physics*, 69(11), 5180–5191.
- Ioslovich, I., Gutman, P.O. and Seginer, I. (2004), Dominant parameter selection in the marginally identifiable case, *Mathematics and Computers in Simulation*, 65(1-2), 127–136.
- Jeppsson, U. and Pons, M.N. (2004), The COST benchmark simulation model - current state and future perspective, *Control Engineering Practice*, 12(3), 299–304.
- Keesman, K.J. (1990), Membership-set estimation using random scanning and principal component analysis, *Mathematics and Computers in Simulation*, 32(5-6), 535–543.
- King, J. and Wong, W.-K. (1998), Optimal minimax designs for prediction in heteroscedastic models, *Journal of Statistical Planning and Inference*, 69(2), 371–383.
- Kramer, M.A., Calo, J.M. and Rabitz, H. (1981), An improved computational method for sensitivity analysis: Green's function method with 'aim', *Applied Mathematical Modelling*, 5, 432–441.
- Kramer, M.A., Rabitz, H., Calo, J.M. and Kee, R.J. (1984), Sensitivity analysis in chemical kinetics: Recent developments and computational comparisons, *International Journal of Chemical Kinetics*, 16, 559–578.
- Li, R., Henson, M.A. and Kurtz, M.J. (2004), Selection of model parameters for off-line parameter estimation, *IEEE on Control Systems Technology*, 12(3), 402–412.
- Ljung, L. (1999), *System Identification: Theory for the User*, Prentice Hall, Englewood Cliffs, New Jersey, USA, second edition.
- Lobry, J.R., Rosso, L. and Flandrois, J.P. (1991), A FORTRAN subroutine for the determination of parameter confidence limits in nonlinear models, *Binary*, 3, 86–93.
- Lyness, J.N. and Moler, C.B. (1967), Numerical differentiation of analytic functions, *SIAM Journal on Numerical Analysis*, 4(2), 202–210.
- Marsili-Libelli, S., Guerrizio, S. and Checchi, N. (2003), Confidence regions of estimated parameters for ecological systems, *Ecological Modelling*, 165(2-3), 127–146.

- Marsili-Libelli, S., Ratini, P., Spagni, A. and Bortone, G. (2001), Implementation, study and calibration of a modified asm2d for the simulation of sbr processes, *Water Science and Technology*, 43(3), 69–76.
- Marsili-Libelli, S. and Tabani, F. (2002), Accuracy analysis of a respirometer for activated sludge dynamic modelling, *Water Research*, 36(5), 1181–1192.
- Martins, J. R. R. A., Sturdza, P. and Alonso, J.J. (2001), The connection between the complex-step derivative approximation and algorithmic differentiation, in: *Proceedings of the 39th Aerospace Sciences Meeting*, Reno, USA.
- Martins, J.R.R.A., Kroo, I.M. and Alonso, J.J. (2000), An automated method for sensitivity analysis using complex variables, in: *Proceedings of the 38th Aerospace Sciences Meeting*, Reno, USA.
- Martins, J.R.R.A., Sturdza, P. and Alonso, J.J. (2003), The complex-step derivative approximation, *ACM Transactions on Mathematical Software*, 29(3), 245–262.
- McKay, M.D., Beckman, R.J. and Conover, W.J. (1979), A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, *Technometrics*, 21(2), 239–245.
- Metcalf and Eddy (1991), *Wastewater Engineering: Treatment, Disposal and Reuse*, McGraw-Hill, New York, USA, 3th edition.
- Michalewicz, Z. (1992), *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Berlin, Germany.
- Monod, J. (1949), The growth of bacterial cultures, *Annual Review of Microbiology*, 3, 371–394.
- Moore, G.E. (1965), Cramping more components on integrated circuits, *Electronics*, 38(8), 114–117.
- Muhlenbein, H. and Schlierkamp-Voosen, D. (1993), Predictive models for the breeder genetic algorithm I. Continuous parameter optimization, *Evolutionary Computation*, 1(1), 25–49.
- Munack, A. (1991), *Measuring, modelling and control*, volume 4, chapter Optimization of sampling, 252–264, Weinheim.
- Nahor, H.B., Scheerlinck, N., Van Impe, J.F. and Nicolai, B.M. (2003), Optimization of the temperature sensor position in a hot wire probe set up for estimation of the thermal properties of foods using optimal experimental design, *Journal of Food Engineering*, 57(1), 103–110.
- Nelder, J.A. and Mead, R. (1965), A simplex method for function minimization, *Computer Journal*, 7, 308–313.

- Omlin, M., Brun, R. and Reichert, P. (2001), Biogeochemical model of lake Zurich: sensitivity, identifiability and uncertainty analysis, *Ecological Modelling*, 141(1-3), 105–123.
- Omlin, M. and Reichert, P. (1999), A comparison of techniques for the estimation of model prediction uncertainty, *Ecological Modelling*, 115(1), 45–59.
- Ono, I., Kita, H. and Kobayashi, S. (1999), A robust real-coded genetic algorithm using unimodal normal distribution crossover augmented by uniform crossover : Effects of self-adaptation of crossover probabilities, in: *GECCO 1999: Proceedings of the Genetic and Evolutionary Computation Conference*, 496–503.
- Ono, I., Kobayashi, S. and Yoshida, K. (2000), Optimal lens design by real-coded genetic algorithms using undx, *Computer Methods in Applied Mechanics and Engineering*, 186(2-4), 483–497.
- Ossenbruggen, P.J., Spanjers, H. and Klapwijk, A. (1996), Assessment of a two-step nitrification model for activated sludge, *Water Research*, 30(4), 939–953.
- Oyama, A., Obayashi, S. and Nakamura, T. (2001), Real-coded adaptive range genetic algorithm applied to transonic wing optimization, *Applied Soft Computing*, 1(3), 179–187.
- Perez-Foguet, A., Rodriguez-Ferran, A. and Huerta, A. (2000), Numerical differentiation for local and global tangent operators in computational plasticity, *Computer Methods in Applied Mechanics and Engineering*, 189(1), 277–296.
- Petersen, B. (2000), *Calibration, identifiability and optimal experimental design of activated sludge models*, Ph.D. thesis, Ghent University, Ghent, Belgium.
- Petersen, B., Gernaey, K., Devisscher, M., Dochain, D. and Vanrolleghem, P.A. (2003), A simplified method to assess structurally identifiable parameters in Monod-based activated sludge models, *Water Research*, 37(12), 2893–2904.
- Petersen, B., Gernaey, K. and Vanrolleghem, P.A. (2001), Practical identifiability of model parameters by combined respirometric-titrimetric measurements, *Water Science and Technology*, 43(7), 347–355.
- Poland, J., Mitterer, A., Knödler, K. and Zell, A. (2001), Genetic algorithms can improve the construction of D-optimal experimental designs, in: Mastorakis, N., editor, *Advances in Fuzzy Systems and Evolutionary Computation*, 227–231.

- Powell, M.J.D. (1964), An efficient method for finding the minimum of a function of several variables without calculating derivatives, *The Computer Journal*, 7(2), 155–162.
- Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1992), *Numerical recipes in C*, Cambridge University Press, Cambridge, UK.
- Pronzato, L. and Walter, E. (1988), Robust experiment design via maximin optimization, *Mathematical Biosciences*, 89(2), 161–176.
- Rabitz, H., Kramer, M. and Dacol, D. (1983), Sensitivity analysis in chemical kinetics, *Annual Review of Physical Chemistry*, 34, 419–461.
- Radcliffe, N.J. (1991), Equivalence class analysis of genetic algorithms, *Complex Systems*, 5(2), 183–205.
- Radcliffe, N.J. (1994), The algebra of genetic algorithms, *Annals of Mathematics and Artificial Intelligence*, 10(4), 339–384.
- Rademaker, M. (2004), *Gebruik van genetische algoritmen voor experimenteel ontwerp*, Master's thesis, Ghent University, Ghent, Belgium.
- Rasheed, K. (1999), Guided crossover: A new operator for genetic algorithm based optimization, in: Angeline, P.J., Michalewicz, Z., Schoenauer, M., Yao, X. and Zalzala, A., editors, *Proceedings of the Congress of Evolutionary Computation*, 1535–1541, IEEE Press, Mayflower Hotel, Washington D.C., USA.
- Reichert, P. (1994), Aquasim - a tool for simulation and data analysis of aquatic systems, *Water Science and Technology*, 30(2), 21–30.
- Reichert, P. and Vanrolleghem, P. (2001), Identifiability and uncertainty analysis of river water quality model no. 1 (RWQM1), *Water Science and Technology*, 43(7), 329–338.
- Reichert, P., von Schulthess, R. and Wild, D. (1995), The use of aquasim for estimating parameters of activated sludge models, *Water Science and Technology*, 31(2), 135–147.
- Saltelli, A., Chan, K. and Scott, E.M. (2000), *Sensitivity Analysis*, Wiley, New York, USA.
- Sarkar, D. and Modak, J.M. (2003), Optimisation of fed-batch bioreactors using genetic algorithms, *Chemical Engineering Science*, 58, 2283–2296.
- Schlierkamp-Voosen, D. (1994), Strategy adaption by competition, in: *Proceedings of the Second European Congress on Intelligent Techniques and Soft Computing*, 1270–1274.
- Schraudolph, N.N. and Belew, R.K. (1992), Dynamic parameter encoding for genetic algorithms, *Machine Learning*, 9, 9–21.

- Seagren, E.A., Kim, H. and Smets, B.F. (2003), Identifiability and retrievability of unique parameters describing intrinsic Andrews kinetics, *Applied Microbiology and Biotechnology*, 61(4), 314–322.
- Seber, G.A.F. and Wild, C.J. (1989), *Nonlinear regression*, Wiley, New York, USA.
- Sedrati, Y., Cabassud, M., Le Lann, M.V. and Casamatta, G. (1999), Sequential experimental design strategy for kinetic parameters estimation, *Computers and Chemical Engineering*, 23, S427–S430.
- Shaffer, J.D. (1985), Multiple objective optimization with vector evaluated genetic algorithms, in: Grefenstette, J.J., editor, *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, 93–100, New Jersey, USA.
- Sin, G., Insel, G., Lee, D.S. and Vanrolleghem, P.A. (2004a), Optimal but robust N and P removal in SBRs: A systematic study of operating scenarios., *Water Science and Technology*, 50(10), 97–105.
- Sin, G., Insel, G., Villez, K., Yoo, C., Lee, D. and Vanrolleghem, P.A. (2004b), Modelling and calibration of nitrogen and phosphorous removal in the BIOMATH SBR, Technical report, BIOMATH: Department of Applied Mathematics, Biometrics and Process Control. Ghent University, Ghent, Belgium.
- Sin, G., Malisse, K. and Vanrolleghem, P.A. (2003), An integrated sensor for the monitoring of aerobic and anoxic activated sludge activities in biological nitrogen removal plants, *Water Science and Technology*, 47(2), 141–148.
- Spanjers, H., Vanrolleghem, P.A., Nguyen, K., Vanhooren, H. and Patry, GG. (1998), Towards a simulation-benchmark for evaluating respirometry-based control strategies, *Water Science and Technology*, 37(12), 219–226.
- Spear, R.C. and Hornberger, G.M. (1980), Eutrophication in Peel Inlet: II. Identification of critical uncertainties via generalized sensitivity analysis, *Water Research*, 14(1), 43–49.
- Spendley, W., Hext, G.R. and Himsworth, F.R. (1962), Sequential application of simplex designs in optimisation and evolutionary operation, *Technometrics*, 4(4), 441–461.
- Squire, W. and Trapp, G. (1998), Using complex variables to estimate derivatives of real functions, *SIAM Review*, 40(1), 110–112.
- Srinivas, N. and Deb, K. (1994), Multi-objective function optimization using non-dominated sorting genetic algorithms, *Evolutionary Computation Journal*, 2(3), 221–248.

- Steppe, K. (2004), *Diurnal dynamics of water flow through trees: design and validation of a mathematical flow and storage model*, Ph.D. thesis, Ghent University, Ghent, Belgium.
- Syddall, M.T., Paul, G.C. and Kent, C.A. (1998), Improving the estimation of parameters of penicillin fermentation models., in: *7th International Conference on Computer Applications in Biotechnology*, 23–28, Osaka, Japan.
- Takacs, I., Patry, G.G. and Nolasco, D. (1991), A dynamic model of the clarification thickening process, *Water Research*, 25(10), 1263–1271.
- Tan, K.C., Lee, T.H. and Khor, E.F. (2002), Evolutionary algorithms for multi-objective optimization: performance assessment and comparisons, *Artificial Intelligence Review*, 17(4), 253–290.
- Tod, M. and Rocchisani, J.-M. (1996), Implementation of OSPOP, an algorithm for the estimation of optimal sampling times in pharmacokinetics by the ED, EID and API criteria, *Computer Methods and Programs in Biomedicine*, 50(1), 13–22.
- Trolltech (1994), *Qt, a cross-platform C++ GUI/API*, <http://www.trolltech.com>.
- Tsutsui, S., Yamamura, M. and Higuchi, T. (1999), Multi-parent recombination with simplex crossover in real coded genetic algorithms, in: *Proceedings of the 1999 Genetic and Evolutionary Computation Conference (GECCO'99)*, 657–664.
- Turanyi, T. (1990), Sensitivity analysis of complex kinetic systems, tools and applications, *Journal of Mathematical Chemistry*, 5, 203–248.
- Valko, P. and Vajda, S. (1984), An extended ODE solver for sensitivity calculations, *Computers and Chemistry*, 8(4), 255–271.
- Van Veldhuizen, D.A. and Lamont, G.B. (2000), Multiobjective evolutionary algorithms: analyzing the state-of-the-art, *Evolutionary Computation*, 8(2), 125–147.
- Van Vooren, L. (2000), *Buffer capacity based multipurpose hard- and software sensor for environmental applications*, Ph.D. thesis, Ghent University, Ghent, Belgium.
- Vanhooren, H., Meirlaen, J., Amerlinck, Y., Claeys, F., Vangheluwe, H. and Vanrolleghem, P.A. (2003), WEST: modelling biological wastewater treatment, *Journal of Hydroinformatics*, 5(1), 27–50.
- Vanrolleghem, P.A. and Keesman, K.J. (1996), Identification of biodegradation models under model and data uncertainty, *Water Science and Technology*, 33(2), 91–105.

- Vanrolleghem, P.A., Sin, G. and Gernaey, K.V. (2004), Transient response of aerobic and anoxic activated sludge activities to sudden substrate concentration changes, *Biotechnology and Bioengineering*, 86(3), 277–290.
- Vanrolleghem, P.A., Van Daele, M. and Dochain, D. (1995), Practical identifiability of a biokinetic model of activated sludge respiration, *Water Research*, 29(11), 2561–2570.
- Vatsa, V.N. (2000), Computation of sensitivity derivatives of Navier-Stokes equations using complex variables, *Advances in Engineering Software*, 31(8-9), 655–659.
- Versyck, K.J., Bernaerts, K., Geeraerd, A.H. and Van Impe, J.F. (1999), Introducing optimal experimental design in predictive modeling: A motivating example, *International Journal of Food Microbiology*, 51(1), 39–51.
- Versyck, K.J., Claes, J.E. and Van Impe, J.F. (1997), Practical identification of unstructured growth kinetics by application of optimal experimental design, *Biotechnology Progress*, 13(5), 524–531.
- Versyck, K.J. and Van Impe, J.F. (2000a), Optimal design of system identification experiments for bioprocesses, *Journal A*, 41(2), 25–34.
- Versyck, K.J.E. and Van Impe, J.F.M. (2000b), Robustness of the iterative optimal experiment design scheme for parameter estimation: identification of microbial heat resistance parameters, in: Dochain, D. and Perrier, M., editors, *International Conference on Computer Applications in Biotechnology (CAB8)*, 31–36.
- Voight, H.M. and Anheyer, T. (1994), Modal mutations in evolutionary algorithms, in: *Proceedings of the First IEEE Conference on Evolutionary Computation*, 88–92.
- von Sperling, M. (1993), Parameter estimation and sensitivity analysis of an activated sludge model using monte carlo simulation and the analyst's involvement, *Water Science and Technology*, 28(11-12), 219–229.
- Wall, M. (1995), *GAlib, a C++ library of genetic algorithm components*, <http://lancet.mit.edu/ga/GAlib.html>.
- Walter, E. and Pronzato, L. (1987), Optimal experiment design for nonlinear models subject to prior uncertainties, *American Journal of Physiology*, 253(3), R530–R534.
- Walter, E. and Pronzato, L. (1990), Qualitative and quantitative experiment design for phenomenological models - a survey, *Automatica*, 26(2), 195–213.
- Walter, E. and Pronzato, L. (1997), *Identification of Parametric Models: From Experimental Data*, Springer Verlag, London, UK.

- Weijers, S.R., Kok, J.J., Preisig, H.A., Buunen, A. and Wouda, T.W.M. (1996), Parameter identifiability in the IAWQ model no 1 for modelling activated sludge plants for enhanced nitrogen removal, *Computer and Chemical Engineering*, 40, S1455–S1460.
- Weijers, S.R. and Vanrolleghem, P.A. (1997), A procedure for selecting best identifiable parameters in calibrating activated sludge model no.1 to full-scale plant data, *Water Science and Technology*, 36(5), 69–79.
- Welch, B.B. (2000), *Practical Programming in Tcl and Tk*, Prentice Hall, New Jersey, USA, 3th edition.
- Whitley, L.D., Mathias, K.E. and Fitzhorn, P.A. (1991), Delta coding: An iterative search strategy for genetic algorithms, in: Belew, R.K. and Booker, L.B., editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, 77–84, Morgan Kaufmann.
- Wilderer, P.A., Irvine, R.L. and Goronszy, MC (2001), *Sequencing Batch Reactor Technology*, Scientific and Technical Report No. 10, IWA publishing, London, UK.
- Wright, A. (1990), Genetic algorithms for real parameter optimization, in: Rawlin, G.J.E., editor, *Proceedings of the First Workshop on the Foundations of Genetic Algorithms and Classifier Systems*, 205–218, Morgan Kaufmann, San Mateo, USA.
- Wright, A.H. (1991), *Foundations of Genetic Algorithms*, chapter Genetic algorithms for real parameter optimization, 205–218, Morgan Kaufman, San Mateo.
- Zitzler, E., Deb, K. and Thiele, L. (2000), Comparison of multiobjective evolutionary algorithms: empirical results, *Evolutionary Computation*, 8(2), 173–195.

Summary

Successful calibration of bioprocess models can only be achieved when information rich data is available. Therefore, it is desired to design experiments in such a way that the data which will be collected meet this objective. In order to design an optimal experiment, different choices have to be made. It has to be decided whether, where and how the system under study will be manipulated and where, how and when measurements will be performed on this system.

The non-linear and dynamic nature of bioprocess models makes the application of optimal experimental design techniques far from straightforward. During this PhD thesis, several issues related to non-linear dynamic optimal experimental design were identified and solutions proposed.

Optimal experimental design involves a series of complicated steps including parameter estimation, sensitivity analysis, non-linear optimization. However, no simulation package exists which combines all these methods. Hence, extensions were programmed to an existing modelling and simulation package, WEST. This modified simulation software, called EAST, is able to solve optimal experimental design problems in a general way and is applicable to every model available within WEST. In order to maintain generality, much emphasis was put on numerical rather than on analytical techniques.

Local sensitivity functions are an important component of the Fisher Information Matrix (FIM) which is used as the basis for optimal experimental design for model calibration. A much used method to calculate these local sensitivity functions is the finite difference technique. However, the practical application of this technique to a non-linear dynamic model poses a problem: a correct perturbation factor needs to be chosen in order to prevent numerical errors or errors related to the non-linearity of the model to influence the sensitivity analysis results. Therefore, a semi-automatic method to detect wrongly calculated sensitivity functions was developed which is based on the quantification of the difference between two sensitivity functions calculated with opposite perturbation factors. In order

to eliminate the error prone and laborious choice of the perturbation factors, a technique based on complex-number calculations (the complex-step derivative approximation method) was also investigated. Using this technique, the user is no longer required to specify perturbation factors and completely reliable results are obtained. However, a significant simulation speed decrease could be observed when using this technique.

In order to quantify the quality of a parameter estimation exercise, the parameter estimation error covariance matrix should be available. This can be calculated based on the Hessian matrix or the FIM. Calculating the Hessian matrix requires the numerical and computationally intensive evaluation of the second derivatives of the objective function with respect to the model parameters. The Richardson's extrapolation technique proved very useful in calculating these derivatives in a correct and automatic way.

Performing optimal experimental design for complex models containing many parameters requires an a priori choice to be made about the number of parameters that will be taken into account for the design. Therefore, techniques are needed to select identifiable parameter sets based on existing data. Making use of the relationship between the FIM and the Hessian, a new method, based on FIM related properties, was proposed to select identifiable parameter subsets which requires only limited user interaction.

The optimization problem related to optimal experimental design for model calibration can be very complex, especially when many experimental degrees of freedom and constraints are considered. The objective surfaces of the FIM criteria typically show a large number of local minima. In order to increase the probability of finding the global optimum, real-coded genetic algorithms (GAs) were successfully applied to the design of measurement campaigns for sequencing batch reactors.

It was also shown that GAs can be used to solve optimization problems which involve combinations of continuous and discrete optimization variables (experimental degrees of freedom).

In order to decrease the optimization computational demand, the experimental design problem was split up into parts: an inner and an outer loop for the optimization of measurement experimental degrees of freedom and manipulation experimental degrees of freedom respectively.

Another well-known problem of experimental design based on FIM properties is that the FIM optimal design criteria are often conflicting and that an experiment which is optimal for a certain design criterion is far from optimal for another design criterion. Finding experimental designs which are optimal with respect to several criteria can be accomplished by performing a multi-objective optimization of the experimental degrees of freedom. Multi-objective GAs proved ideal candidates for solving optimal experimental design problems dealing with several FIM criteria and also considering experimental costs.

The classical iterative optimal experimental design procedure (iterative experimentation, calibration and experimental design) involves different, rather complicated mathematical and practical steps in order to obtain the cali-

brated model. These steps require much user interaction and expert knowledge. As a solution, an automatic optimal experimental design procedure was developed in which user interaction and expert knowledge is only required at the beginning. All subsequent steps of the iterative search for the best parameter estimates can be performed automatically. These steps include (1) finding the optimal experiment, (2) performing the experiments in practice and (3) recalibrating the model. The proposed procedure was successfully applied to calibrate a one-step nitrification model using a respirometric experiment.

An important aspect of optimal experimental design for parameter estimation for non-linear models is the dependency of the design on the model parameters. This is caused by the fact that the FIM is calculated from sensitivity functions (partial derivatives of the model variables to the parameters) which, for non-linear models, are still function of the model parameters. This causes designs based on the FIM to be only locally optimal (for the particular values of the parameters for which the experimental design is performed) which may lead to unsatisfactory results when the used model parameters are too different from the “real” parameter values. Several known techniques for robust experimental design were studied and implemented in EAST together with a newly proposed technique based on the sensitivity of the FIM criteria to parameter changes. It was shown that the designs based on known robust experimental design criteria produced the best results. It was also found that robust experimental design is very computationally demanding. At this moment, the computational requirements limit the application of these techniques to small, simple models with a limited set of experimental degrees of freedom.

As a concluding remark, it can be said that optimal experimental design is an essential tool if high quality knowledge and models for complex dynamic non-linear systems like bioprocesses are to be obtained.

Samenvatting

Bioprocessmodellen kunnen enkel succesvol worden gekalibreerd wanneer data met een voldoende informatieinhoud beschikbaar zijn. Het is daarom aangewezen om experimenten zodanig te ontwerpen dat data verzameld kunnen worden die aan deze eis voldoen. Om een optimaal experiment te ontwerpen moeten verschillende keuzes gemaakt worden. Er moet beslist worden waar en hoe het bestudeerde systeem moet gemanipuleerd worden en waar, wanneer en hoe metingen uitgevoerd moeten worden.

Het dynamische en niet-lineaire karakter van bioprocessmodellen zorgt ervoor dat de toepassing van optimaal experimenteel ontwerp niet vanzelfsprekend is. Gedurende deze doctoraats thesis werden verschillende knelpunten met betrekking tot optimaal experimenteel ontwerp voor dynamische en niet-lineaire systemen geïdentificeerd en oplossingen voorgesteld.

Optimaal experimenteel ontwerp omhelst een reeks van ingewikkelde stappen zoals parameterschatting, sensitiviteitsanalyse en niet-lineaire optimalisatie. Tot op heden was echter geen softwarepakket beschikbaar dat al deze methoden combineerde in één geheel. Daarom werd een bestaand modellerings- en simulatiepakket, WEST, voorzien van verschillende uitbreidingen. Dit gemodificeerde software pakket, EAST genaamd, is in staat om optimaal experimenteel ontwerp problemen op te lossen op een algemene manier en is toepasbaar op alle binnen WEST geïmplementeerde modellen. Om deze algemeenheid te verzekeren werd vooral aandacht besteed aan numerieke technieken in plaats van aan analytische technieken.

Lokale sensitiviteitsfuncties vormen een belangrijke component van de Fisher Informatie Matrix (FIM) die de basis is voor optimaal experimenteel ontwerp voor modelkalibrering. Een veel gebruikte methode om deze lokale sensitiviteitsfuncties te berekenen is de eindige verschil methode. De toepassing van deze techniek op een niet-lineair model kan leiden tot problemen. Inderdaad, er dient een juiste perturbatiefactor gekozen te worden opdat numerische fouten en fouten als gevolg van de niet-lineariteit van het model de resultaten van de sensitiviteitsanalyse niet zouden beïnvloeden.

Daarom werd een semi-automatische techniek ontwikkeld om foutief berekende sensitiviteitsfuncties te detecteren op basis van de kwantificatie van verschillen tussen sensitiviteitsfuncties berekend met tegengestelde perturbatiefactoren. Om de arbeidsintensieve en foutgevoelige zoektocht naar optimale perturbatiefactoren te omzeilen werd ook een techniek bestudeerd op basis van berekeningen met complexe getallen (de complexe-stap afgeleide approximatie techniek). Dankzij deze techniek is het voor de gebruiker niet langer nodig om correcte perturbatiefactoren te specificeren en kunnen volledig betrouwbare resultaten bekomen worden. Daartegenover staat echter wel dat de invoering van complexe getallen een significante vertraging van de simulaties veroorzaakt.

Het kwantificeren van de kwaliteit van een parameterschattingsoefening kan gebeuren indien de parameterschattingfoutcovariantiematrix beschikbaar is. Deze matrix kan berekend worden op basis van de Hessiaanmatrix of de FIM. Het berekenen van de Hessiaanmatrix vereist de numerische en computationeel ingewikkelde evaluatie van de tweede afgeleiden van de parameterschattingsobjectieffunctie naar de modelparameters. De Richardson extrapolatietechniek werd succesvol toegepast om deze afgeleiden correct en automatisch te berekenen.

Het toepassen van optimaal experimenteel ontwerp voor modellen met veel parameters vereist een a priori keuze met betrekking tot het aantal parameters die gebruikt zullen worden in het ontwerp. Hiervoor zijn technieken waarmee identificeerbare parameters kunnen bepaald worden op basis van beschikbare data noodzakelijk. Gebruikmakend van de relatie tussen de FIM en de Hessiaanmatrix werd een nieuwe methode, gebaseerd op FIM-eigenschappen, voorgesteld om een set van identificeerbare parameters te bepalen. Deze techniek vereist slechts een beperkte interactie met de gebruiker.

Het optimalisatieprobleem gekoppeld aan optimaal experimenteel ontwerp kan zeer complex zijn, zeker wanneer veel experimentele vrijheidsgraden en beperkingen beschouwd worden. Typisch vertoont het objectiefoppervlak van de FIM-criteria verschillende lokale minima. Om de kans te vergroten het globale minimum te detecteren werden reële-getal genetische algoritmes (GAs) succesvol gebruikt, ondermeer voor het ontwerp van meetcampagnes voor sequentiële batchreactoren.

Er werd eveneens aangetoond dat GAs gebruikt kunnen worden om optimalisatieproblemen op te lossen die beschreven worden door combinaties van discrete en continue variabelen (experimentele vrijheidsgraden).

Om de computationele last te beperken werd het optimaal experimenteel ontwerpprobleem opgesplitst in twee delen: een buitenste en een binnenste lus waarbij de buitenste lus gebruikt werd voor de optimalisatie van de experimentele manipulaties en de binnenste lus voor de optimalisatie van de metingen.

Een welbekend probleem van optimaal experimenteel ontwerp gebaseerd op FIM-criteria is dat het gebruik van deze criteria vaak conflicterende “optimale” experimenten oplevert. Een experiment dat optimaal is voor een

bepaald criterium kan ver van optimaal zijn met betrekking tot een ander criterium. Experimenten ontwerpen die optimaal zijn met betrekking tot verschillende criteria kan gebeuren aan de hand van multi-objectief optimalisatie van de experimentele vrijheidsgraden. Er werd aangetoond dat multi-objectief GAs ideale kandidaten zijn om dit soort problemen op te lossen, waarbij naast FIM criteria ook experimentele kosten in rekening gebracht konden worden als additioneel objectief.

De klassieke iteratieve optimaal experimenteel ontwerpprocedure (opvolging van experimenteren, kalibreren en optimaal experimenteel ontwerp) vereist verschillende, relatief gecompliceerde, mathematische en praktische stappen om tot een goed gekalibreerd model te komen. Deze stappen vereisen heel wat expert kennis en interactie met de gebruiker. Als antwoord op dit probleem werd een automatische experimenteel ontwerpprocedure ontwikkeld waarbij expert kennis en interactie met de gebruiker enkel initieel noodzakelijk is. Alle verdere stappen van de iteratieve zoektocht naar de optimale parameters gebeuren automatisch. Deze stappen zijn ondermeer (1) het zoeken van het optimale experiment, (2) het uitvoeren van dit experiment in de praktijk en (3) de kalibrering van het model. De voorgestelde procedure werd succesvol toegepast om een één-stap nitrificatiemodel te kalibreren gebruikmakend van een respirometrisch experiment.

Optimaal experimenteel ontwerp voor de kalibrering van niet-lineaire modellen wordt gekarakteriseerd door de afhankelijkheid van het experimenteel ontwerp van de modelparameters geldend voor het bestudeerde systeem. Dit wordt veroorzaakt door het feit dat de FIM berekend wordt op basis van sensitiviteitsfuncties (partieel afgeleiden van de modelvariabelen naar de modelparameters) die voor niet-lineaire modellen nog steeds functie zijn van de modelparameters. Dit zorgt ervoor dat experimentele ontwerpen slechts lokaal geldig zijn (voor de specifieke waarden van de parameters waarmee het experimenteel ontwerp bepaald werd). Dit kan leiden tot ontgoochelende resultaten indien de gebruikte parameterwaarden te sterk afwijken van de werkelijke waarden. Daarom werden verschillende bestaande technieken voor robuust experimenteel ontwerp geïmplementeerd samen met een nieuwe techniek gebaseerd op de sensitiviteit van de FIM-criteria aan de parameterwaarden. Hierbij werd aangetoond dat de experimenten ontworpen op basis van de bekende technieken voor robuust ontwerp betere resultaten opleverden. Er kon ook worden opgemerkt dat robuust optimaal experimenteel ontwerp heel computationeel intensief is en momenteel enkel toepasbaar is voor relatief simpele modellen met een beperkte set van experimentele vrijheidsgraden.

Als conclusie kan gesteld worden dat optimaal experimenteel ontwerp een essentiële techniek is om informatie van een hoge kwaliteit te verkrijgen voor het modelleren van complexe, dynamische en niet-lineaire systemen zoals bioprocessen.

Curriculum vitae

Personalia

Naam Dirk Jos Willy De Pauw
Geboortedatum 28 maart 1977
Geboorteplaats Wilrijk
Nationaliteit Belg

Opleiding

- 1989-1995 **Secundair onderwijs:** Wiskunde (8 uur)-Wetenschappen, Sint-Norbertus Instituut, Antwerpen.
- 1995-2000 **Universitair onderwijs:** Bio-ingenieur in de milieutechnologie (onderscheiding).
Faculteit Landbouwkunde en Toegepaste Biologische Wetenschappen, Universiteit Gent.
Scriptie: "Modellering van biofilmsystemen in trickling filters".

Bijkomende cursussen

- 2001 **Naam:** Gegevensverwerking.
Plaats: Universiteit Gent, Gent, België.
Datum: 2de semester.
- 2002 **Naam:** Summer School on Sensitivity Analysis (SAMO2002).
Plaats: Venetië, Italië.
Datum: 12-15 juni 2002.

2003 **Naam:** Programmeren II.
Plaats: Universiteit Gent, Gent, België.
Datum: 1ste semester.

Congressen, symposia en studiedagen

2000 **Naam:** 5th International Symposium on Systems Analysis and Computing in Water Quality Management (WATERMATEX 2000).
Plaats: Gent, België.
Datum: 18-20 september 2000.

2001 **Naam:** 5th Kollekolle Seminar: Modelling of Activated Sludge Processes in Theory and Practice.
Plaats: Kopenhagen, Denemarken.
Datum: 10-12 september 2001.

2003 **Naam:** 4th IMCAS Symposium on Mathematical Modelling (MATHMOD 2003).
Plaats: Vienna University of Technology, Wenen, Oostenrijk.
Datum: 5-7 februari 2003.

Naam: World Wide West meeting (WWWest 2003).
Plaats: Het Pand, Gent, België.
Datum: 14-16 april 2003.

Naam: 9th PhD Symposium on Applied Biological Sciences.
Plaats: Leuven, België.
Datum: 16 oktober 2003.

2004 **Naam:** World Wide West meeting (WWWest 2004).
Plaats: Tsinghua University, Beijing, China.
Datum: 1-2 november 2004.

Naam: 6th International Symposium on Systems Analysis and Computing in Water Quality Management (WATERMATEX 2004).
Plaats: Beijing, China.
Datum: 3-5 november 2004.

2005 **Naam:** Physical versus Black-box Models: What's New in System Identification?
Plaats: KVIV Ingenieurshuis, Antwerpen, België.
Datum: 17 maart 2005.

Werkervaring

2000-2005 **Plaats:** Universiteit Gent, Faculteit Landbouwkundige en Toegepaste Biologische Wetenschappen, Vakgroep Toegepaste Wetkunde, Biometrie en Procesregeling.

Functie: Doctoraatsbursaal, Bijzonder Onderzoeksfonds Universiteit Gent (BOF).

Doctoraat: "Optimaal experimenteel ontwerp voor kalibrering van bioprocesmodellen: een gevalideerde software toolbox".

Publicaties

Publicaties in tijdschriften

- Vanhooren H., **De Pauw D.** and Vanrolleghem P.A. (2003). Induction of denitrification in a pilot-scale trickling filter by adding nitrate at high loading rate. *Water Science and Technology* **47**(11), 61-68.
- **De Pauw D.J.W.**, Sin G., Insel G., Van Hulle S.W.H, Vandenberghe V. and Vanrolleghem P.A. (2004). Discussion of "Assessing parameter identifiability of activated sludge model number 1" by Pedro Afonso Cunha and Maria Da Conceicao Cunha. *Journal Of Environmental Engineering-ASCE* **130**(1), 110-112.
- Deksissa T., **De Pauw D.** and Vanrolleghem P.A. (2004). Dynamic in-stream fate modeling of xenobiotic organic compounds: A case study of linear alkylbenzene sulfonates in the Lambro River, Italy. *Environmental Toxicology and Chemistry* **23**(9), 2267-2278.
- **De Pauw D.J.W.**, Vanrolleghem P.A. (2005). Practical aspects of sensitivity function approximation. *Mathematical and Computer Modelling of Dynamical Systems*, Revised version submitted.
- **De Pauw D.J.W.** and Vanrolleghem P.A. (2005). Designing and performing experiments for model calibration using an automated iterative procedure. *Water Science and Technology*, Revised version submitted.
- Sin G., Van Hulle S.W.H., **De Pauw D.J.W.**, van Griensven A. and Vanrolleghem, P.A. (2005). A critical comparison of systematic calibration protocols for activated sludge models: A SWOT analysis. *Water Research*, Revised version submitted.
- Steppe K., **De Pauw D.J.W.**, Lemeur R. and Vanrolleghem P.A. (2005). Comparison of two approaches to calculate the capacitance in a model linking sap flow dynamics in a tree to daily stem diameter fluctuation and stem growth. *Tree Physiology*, Submitted.

Publicaties in conferentie proceedings

- Vanhooren H., **De Pauw D.**, Gillot S., Demey D., Vannijvel I. and Vanrolleghem P.A. (2000). Adequate complexity for biofilms systems modelling. *Communications in Agricultural and Applied Biological Sciences Ghent University*, **65**(4), 39-44.
- **De Pauw D.J.W.**, Carvalho G., Nopens I., Verdonck F.A.M., Meirlaen J. and Vanrolleghem P.A. (2002). WEST, a general tool for dynamic modelling and simulation of biodegradation processes. In: Proceedings of the 12th Annual Meeting of SETAC Europe. Vienna, Austria, May 12-16, 2002.
- Vanhooren H., **De Pauw D.** and Vanrolleghem P.A. (2002) Induction of denitrification in a pilot-scale trickling filter by adding nitrate at high loading rate. In: Proceedings of the 3rd IWA World Water Congress. Melbourne, Australia, April 7-12 2002.
- Vanhooren H., Van Hulle S., **De Pauw D.** and Vanrolleghem P.A. (2002). Monitoring and modelling a pilot-scale trickling filter using on-line off-gas analysis. In: Proceedings of the International Specialised Conference on Biofilm Monitoring. Porto, Portugal, March 17-20, 2002. 260-263.
- **De Pauw D.J.W.** and Vanrolleghem P.A. (2003). Optimal experimental design for model calibration: general procedure. *Communications in Agricultural and Applied Biological Sciences Ghent University*, **68**(3), 95-98.
- **De Pauw D.J.W.** and Vanrolleghem P.A. (2003). Practical aspects of sensitivity analysis for dynamic models. In: Proceedings of the 4th IMACS Symposium on Mathematical Modelling (MATHMOD). Vienna, Austria, February 5-7, 2003. 47.
- Vanrolleghem P.A., Insel G., Petersen B., Sin G., **De Pauw D.**, Nopens I., Doverman H., Weijers S. and Gernaey K. (2003). A comprehensive model calibration procedure for activated sludge models. In: Proceedings of the 76th Annual WEF Conference and Exposition. Los Angeles, USA, October 11-15, 2003.
- Bernard O., Le Dantec B., Chahuat B., Steyer J.-P., Lardon L., Lambert S., Ratini P., Lema J., Ruiz G., Rodriguez J., Vanrolleghem P.A., Zaher U., **De Pauw D.**, De Neve K., Lievens K., Dochain D., Schoefs O., Farina R., Alcaraz Gonzalez V., Lemaire P., Martinez J.A., Duclaud O. and Lavigne J.F. (2004). TELEMAT: An integrated system to remote monitor and control anaerobic wastewater treatment plants through the Internet. In: Proceedings of the 10th World Congress on Anaerobic Digestion (AD10). Montreal, Canada, August 29 - September 2, 2004.

- **De Pauw D.J.W.**, Zaher U. and Vanrolleghem P.A. (2004). Protocol and optimal experimental design to set up a monitoring and control system at an anaerobic digester. In: Proceedings of the 10th World Congress on Anaerobic Digestion (AD10). Montreal, Canada, August 29 - September 2 2004.
- Van Hulle S.W.H., Maertens J., **De Pauw D.J.W.** and Vanrolleghem P.A. (2004). Using parameter sensitivity analysis of the CANON biofilm process: What to measure, where to measure and under which conditions? In: Proceedings of the 2nd IWA Young Researchers Conference. Wageningen, The Netherlands, May 3-4 2004. 59-66.
- **De Pauw D.J.W.** and Vanrolleghem P.A. (2004). Designing and performing experiments for model calibration using an automated iterative procedure. In: Proceedings of the 6th International Symposium on Systems Analysis and Computing in Water Quality Management (WATERMATEX 2004). Beijing, China, November 3-5, 2004.
- **De Pauw D.J.W.** and Vanrolleghem P.A. (2005). Using the complex-step derivative approximation method to calculate local sensitivity functions. 17th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation. Paris, France, July 11-15, 2005. Accepted.
- Sin G., **De Pauw D.J.W.** and Vanrolleghem P.A. (2005). Towards a unified protocol for the calibration of dynamic models for wastewater treatment plants: a systems identification point of view. 3rd IWA Leading-Edge Conference & Exhibition on Water and Wastewater Treatment Technologies. Sapporo, Japan, June 6-8, 2005. Accepted.

Technische rapporten

- Rousseau D., Verdonck F., **De Pauw D.J.W.** and Vanrolleghem P.A. (2000). Technical manual for the RAP server 1.2 & add-in to WEST. Technical report, BIOMATH: Department of Applied Mathematics, Biometrics and Process Control. Ghent University, Ghent, Belgium.
- **De Pauw D.J.W.**, Chachuat B., Bernard O. and Vanrolleghem P.A. (2003). A set of procedures to estimate the model parameters on-line and off-line, Technical report, TELEMAT IST 2000-28156, "Telemonitoring and Advanced Telecontrol of High-Yield Wastewater Treatment", Deliverable 3.2.
- Zaher U., **De Pauw D.J.W.** and Vanrolleghem P.A. (2003). Protocol to set up the TELEMAT system on a new plant, Technical report, TELEMAT IST 2000-28156, "Telemonitoring and Advanced Telecontrol of High-Yield Wastewater Treatment", Deliverable 1.2.

Tijdschrift refereeing

- Mathematical and Computer Modelling of Dynamical Systems
- Water Research

Presentaties

Mondelinge presentaties

- De Pauw D.J.W. and Vanrolleghem P.A. (2003). Practical aspects of sensitivity analysis for dynamic models. 4th IMACS Symposium on Mathematical Modelling (MATHMOD 2003), Vienna, Austria, February 5-7, 2003.
- De Pauw D.J.W. and Vanrolleghem P.A. (2003). Optimal experimental design for model calibration. World Wide West meeting (WWWest 2003), Het Pand, Gent, Belgium, April 14-16, 2003.
- De Pauw D.J.W. and Vanrolleghem P.A. (2004). The use of genetic algorithms in parameter estimation and optimal experimental design problems. World Wide West meeting (WWWest 2004), Tsinghua University, Beijing, China, November 1-2, 2004.
- De Pauw D.J.W. and Vanrolleghem P.A. (2004). Designing and performing experiments for model calibration using an automated iterative procedure. 6th International Symposium on Systems Analysis and Computing in Water Quality Management (WATERMATEX 2004), Beijing, China, November 3-5, 2004.

Poster presentaties

- De Pauw D.J.W. and Vanrolleghem P.A. (2003). Optimal experimental design for model calibration: General procedure. 9th PhD Symposium on Applied Biological Sciences, Leuven, Belgium, October 16, 2003.