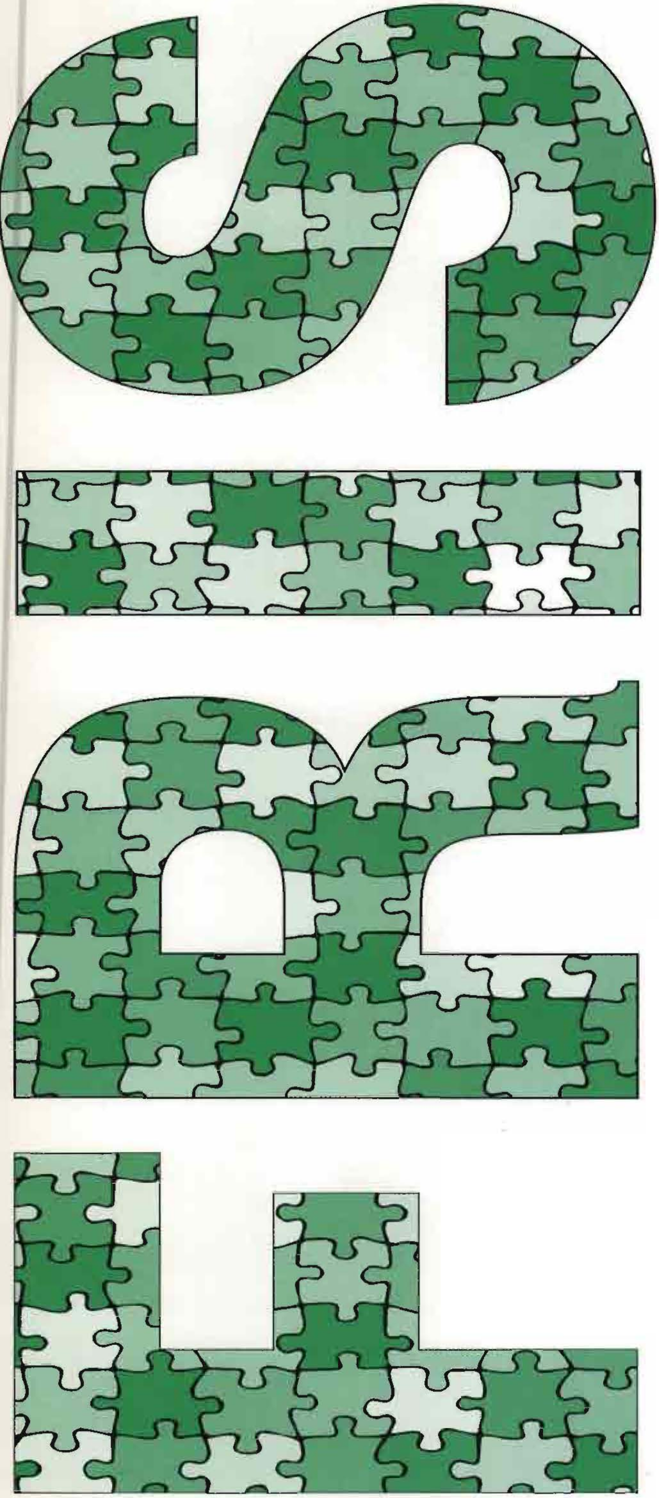
A practical approach to recipe improvement and optimization in the batch processing industry

DOI:
10.6100/IR452094

Document status and date:
Published: 01/01/1996

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)



A PRACTICAL APPROACH TO RECIPE IMPROVEMENT AND OPTIMIZATION IN THE BATCH PROCESSING INDUSTRY

ZOFIA VERWATER-LUKSZO

A PRACTICAL APPROACH TO RECIPE IMPROVEMENT AND OPTIMIZATION IN THE BATCH PROCESSING INDUSTRY

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de Rector Magnificus, prof.dr. J.H. van Lint, voor een commissie aangewezen door het College van Dekanen in het openbaar te verdedigen op dinsdag 30 januari 1996 om 16.00

door

Zofia Verwater - Lukszo

geboren te Olsztyn (Polen)

Dit proefschrift is goedgekeurd door de promotoren:

prof.ir. O. Rademaker
en
prof.ir. J.E. Rijnsdorp

Aan Casimir en Vincent

Cover design: The Communication Company, Breda
Druk: Drukkerij Helmer bv, Breda
Bindwerk: GAC bv, Breda

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Verwater-Lukszo, Zofia

A Practical Approach to Recipe Improvement and Optimization in the Batch Processing Industry / Zofia Verwater-Lukszo
Eindhoven: Technische Universiteit Eindhoven - Ill.
Proefschrift Technische Universiteit Eindhoven - Met lit. opg. -
Met samenvatting in het Nederlands.
ISBN: 90-386-0257-X
Trefw.: procesindustrie / experimentontwerp / optimalisatie

CONTENTS

Acknowledgement

This thesis describes the results of my research during five years within the framework of the project "Computer-Aided Recipe Improvement in the Process Industry". This work was financially supported by the Dutch Technology Foundation STW, which is gratefully appreciated.
The project was carried out in collaboration with the TNO/TPD Institute of Applied Physics and four process industries, namely AKZO Information Services, AVEBE, DSM Resins and Gist-brocades.

Many people contributed to this project.
Especially, I would like to thank the project leaders prof.ir. Onno Rademaker and prof.ir. John Rijnsdorp for their guidance in the research and their constant support in all the facets of my PhD life.

I would also like to mention two project members: Karel Keesman and Ger Otten, in particular. They are gratefully acknowledged for the pleasant collaboration. The results of their research and the FRIS-applications they did gave an extra support for my conclusions.

The experiments with epoxy resin Epikote 1001 were done by Laetitia Ingen Housz. I thank her for the precise work, which made the first real application of the FRIS-approach possible.

I also wish to extend my thanks to the software team: Jarek Michalak, Ronald van Gerwen and Walter Renes.

I am also very grateful to many students of the Faculty of Technical Physics and Mechanical Engineering who worked on this research project.

Moreover, I would like to thank Aaris Raats from Akzo Nobel Resins in Bergen op Zoom not only for the information about the Setal process, but also for his practical remarks on my research done in his factory.

During my efforts to write in English, I was supported by Lies Ralston-Leys who is gratefully acknowledged for correcting the text.

Last but not least, I am grateful to the "home front" for accepting the fact that writing a thesis not only takes time but also costs a lot of energy.

Notation, Symbols and Abbreviations

Mathematical/statistical notation:

$d\mathbf{x}(t)/dt$	derivative of **x** with respect to t
H_0	null hypothesis
max	maximize
MS_{PE}	pure mean square error
MS_{RESID}	residual mean square
MS_{TOTAL}	total mean square
PRESS	prediction sum of squares
R^2	coefficient of determination
R^2_{adj}	adjusted coefficient of determination
R^2_{PRESS}	coefficient of determination related to PRESS
RMS Error	root mean square error
SS_{LOF}	lack of fit sum of squares
SS_{PE}	pure error sum of squares
SS_{REGR}	regression sum of squares
SS_{RESID}	residual sum of squares
SS_{TOTAL}	total sum of squares
t-value	value of the t-statistics
$\mathbf{X}^{-1}$	inverse of **X**
$\mathbf{X}^T$	transpose of **X**
$\nabla J(\mathbf{x})$	gradient of J
χ^2	Chi-square statistic

Symbols:

$\mathbf{A}$	linear constraint matrix on $\mathbf{x}$
AC_end	final value of the acid number
ac_n	acid number measured at the n^{th} sample moment
$\mathbf{ax}_{MAX}$	upper linear constraints on $\mathbf{x}$
$\mathbf{ax}_{MIN}$	lower linear constraints on $\mathbf{x}$
BA_COR	amount of BA added once at the correction moment [gr]
c	term defined in (C.6), equal to 1 or 0
$\mathbf{c}(\mathbf{x})$	vector function of non-linear constraints on $\mathbf{x}$
$\mathbf{c1}_{MAX}$, $\mathbf{c2}_{MAX}$	upper general constraints
$\mathbf{c1}_{MIN}$, $\mathbf{c2}_{MIN}$	lower general constraints
C_{ii}	the $(i,i)^{th}$ element of $(\mathbf{X}^T\mathbf{X})^{-1}$
const	constant in performance criterion J, see (3.26)
Corr_alc	added amount of the corrective alcohol [mass unit]
$corr_n$	correction just before the n^{th} sample moment on the basis of the measured conditions at the $(n-1)^{th}$ moment
crit1	some preset value used by stopping criteria, see (3.57)
crit2	some preset value used by stopping criteria, see (3.58)
crit3	some preset value used by stopping criteria, see (D.1)
crit4	some preset value used by stopping criteria, see (D.2)
crit5	some preset value used by stopping criteria, see (D.3)
$\mathbf{cx}_{MAX}$	upper bound on $\mathbf{c}(\mathbf{x})$
$\mathbf{cx}_{MIN}$	lower bound on $\mathbf{c}(\mathbf{x})$
D	Domain of the multiplex-fitting method
DA	Design Area of the multiplex-fitting method
Delta_AC	the difference in the acid number with respect to the heart line
Delta_VIS	the difference in the viscosity with respect to the heart line [Pa·s]
d	$= 1$, or $= t_{BC}$; see (3.26)

DGEBA_COR	amount of DGEBA added once at the correction moment [gr]
E_1001	final amount of Epikote 1001[gr]
ENERGY	amount of the energy used for the production of Epikote 1001 [MJ]
$\mathbf{f}$	vector function
$\mathbf{g}$	vector function
$\mathbf{h}$	vector function
H_cor	amount of correction with alcohol [mass unit]
$\bar{J}(\mathbf{x}_i)$	mean of the criterion values for the vertex $\mathbf{x}_i$
$\mathbf{J}$	multi-objective criterion
J	performance criterion, performance index
$J(\mathbf{x}_i^N)$	performance criterion value at $\mathbf{x}_i^N$
K	number of distinct categories used in χ^2 test, see (C.15)
L_cor	amount of correction with an acid [mass unit]
Li	initial level of the approximating function
M4h	total biomass at the sample moment [kg]
N	iteration number
N	number of lines used for approximation of u(t)
nd	number of decision vectors in the multi-objective method of Woods
New_Delta_AC	the new difference in the acid number with respect to the heart line after correction
New_Delta_VIS	the new difference in the viscosity with respect to the heart line after correction [Pa·s]
n_k	number of experiments for category k in χ^2 test, see (C.15)
np	number of compared pairs by the multi-objective method of Woods
OHv_no	hydroxyl number
p	number of model terms
$\mathbf{p}$	gradient, direction of search
P4h	product amount at the sample moment [penicillin unit]

Pf	final amount of penicillin [penicillin unit]
Pf_cor	final product amount after correction [penicillin unit]
p_k	the relative frequency for category k in χ^2 test, see (C.15)
r_j	number of test-run repetitions at the j^{th} setting of recipe items
Sd	sugar dosage [kg/h]
Sl	temperature slope [°C/h]
Sl_cor	temperature slope after correction [°C/h]
s_n	slope of the n^{th} line piece
ss_n	see (3.35a)
t	time
$T(t;;t_0,t_F,\mathbf{x})$	temperature profile [°C]
t_0	initial time
t_1	the first break-point used in a piece-wise linear approximation of u(t)
Tb	beginning temperature [°C]
t_{BC}	batch cycle time
t_C	correction time
TEMP	temperature during the production of Epikote 1001 [°C]
TEMP_COR	process temperature after correction [°C]
t_F	final time
t_{IDLE}	time in waiting for the completion of the batch cycle time t_{BC}
TIME	processing time of Epikote 1001 [min]
TIME_AC12	time of the solvent process needed to reach the acid number equal to 12 [h]
TIME_SOL	duration of the solvent process [h]
t_{MAX}	upper bound on t_F
t_{MIN}	lower bound on t_F
t_{REST}	time of "secondary" operations
t_S	sample time
t_{TOT}	actual total batch time

$\mathbf{u}(t)$	vector of control variables
$\mathbf{u}(t;t_0,t_F)$	time-dependent recipe item vector
$u(t;t_0,t_F,\mathbf{x})$	approximation of u(t) on the interval [t_0,t_F]
u(t)	element of **u**(t), a time-dependent recipe item
$\mathbf{u}_{MAX}$	upper constraint on **u**(t)
$\mathbf{u}_{MIN}$	lower constraint on **u**(t)
v	recipe item vector selected for optimization of process duration at Akzo
VIS_end	final value of viscosity [Pa·s]
vis_n	viscosity measured at the n^{th} sample moment
w(t)	state variables defined to replace integral constraints
W	matrix used by the multi-objective method of Woods
WPE	Weight Per Epoxy group [gr/mole]
WPE_1001	the final WPE-number of the formed epoxy resin Epikote 1001 [gr/mole]
WPE_50	the WPE-number of the reactor mixture after 50 min [gr/mole]
WPE_70	the WPE-number of the reactor mixture after 70 min [gr/mole]
WPE_828	the WPE-number of DGEBA [gr/mole]
X	feasible area
X	data matrix used by regression
x	time-independent recipe item vector or simplex vertex
x'	first decision vector in the multi-objective method of Woods
x''	second decision vector in the multi-objective method of Woods
x(t)	vector of states variables
$\mathbf{x}_b^N$	the best approximation to the solution at the N^{th} iteration of the Nelder-Mead or the multiplex fitting method
$\mathbf{x}^c$	simplex vertex obtained by contraction
$\mathbf{x}^{COR}$	correction variable vector
$\mathbf{x}^e$	simplex vertex obtained by expansion
$\mathbf{x}_i^N$	i^{th} simplex vertex at the N^{th} iteration

$\mathbf{x}_{MAX}$	upper bound on $\mathbf{x}$
$\mathbf{x}_{MIN}$	lower bound on $\mathbf{x}$
$\mathbf{x}^{N}$	centroid of all simplex vertices except $\mathbf{x}_{w}^{N}$ at the Nth iteration
$\mathbf{x}_{opt}$	optimal $\mathbf{x}$
$\mathbf{x}^{r}$	simplex vertex obtained by reflection
$\mathbf{x}^{sc}$	simplex vertex obtained by shadow contraction
X^{spec}	solution of the end-specification problem
$\mathbf{x}_{w-1}^{N}$	the second worst approximation to the solution at the Nth iteration of the Nelder-Mead method
$\mathbf{x}_{w}^{N}$	the worst approximation to the solution at the Nth iteration of the Nelder-Mead method
$\mathbf{y}$	process output vector; response vector
y_{jk}	kth response measurement for the jth experimental settings
$\hat{y}_{i}$	estimated response of the ith experiment/test run
$\bar{y}$	mean of n measured responses
$\bar{y}_{j}$	response mean for the jth experimental setting
$\mathbf{y}_{MAX}$	upper bound on $\mathbf{y}$
$\mathbf{y}_{MIN}$	lower bound on $\mathbf{y}$
$\hat{y}_{model\backslash\, i}$	response of the ith experiment predicted by a model estimated without the ith test run
$\mathbf{y}^{S}$	process output vector, which should be measured at t_S
$\mathbf{y}^{S}{}_{m}$	measurement of $\mathbf{y}^{S}$ at t_S
$\mathit{\$}_{U}$	coefficient matrix
$\$_{U}$	value vector corresponding to $\mathbf{u}(t)$
$\mathit{\$}_{X}$	coefficient matrix
$\$_{X}$	value vector corresponding to $\mathbf{x}$
$\mathit{\$}_{Y}$	coefficient matrix
$\$_{Y}$	value vector corresponding to $\mathbf{y}$
f_{A}	function describing correction model for acid number

$f_{a,n}$	function describing correction model for acid number at the n^{th} sample moment
f_V	function describing correction model for viscosity
$f_{v,n}$	function describing correction model for viscosity at the n^{th} sample moment
α	coefficient
β	coefficient
γ	coefficient
δ	coefficient
δx	see (3.16)
ε	stochastic normally distributed error with zero mean and variance σ^2
ε_j	real number used by ε-constraint method, see (F.5)
ζ	number corresponding to the experimental error, see (3.58)
ζ_i	see (F.3)
η	coefficient
$\Theta(\boldsymbol{\tau}_0, \boldsymbol{\tau}_F)$	initial cost/final value function, e.g. $= \Psi(\mathbf{x}(t_0)) + \Phi(\mathbf{x}(t_F))$, see Table 3.12
$\lambda(t)$	Lagrange multiplier
μ	multiplier in transversality conditions
$\pi(t)$	adjoint (costate) variable
ρ_i	relaxation level in J_i used by the lexicographic multi-objective method
σ^2	variance
$\boldsymbol{\tau}_0$	$= (t_0, \mathbf{x}(t_0))$, see Table 3.12
$\boldsymbol{\tau}_F$	$= (t_F, \mathbf{x}(t_F))$, see Table 3.12
$\Phi(\mathbf{x}(t_F))$	final value function, see (3.37)
$\Psi(\mathbf{x}(t_0))$	initial cost function, see (3.37)
ω	weighting factor
$\mathbb{R}$	space of real numbers
$\Re$	arbitrary space

$\mathcal{H}(\mathbf{x}(t), \mathbf{u}(t),t)$	Hamiltonian
$\mathscr{L}(\mathbf{x}(t),\mathbf{u}(t),t)$	Lagrangean
i,j,k,m,n	indices

Abbreviations:

ANOVA	ANalysis Of VAriance
ARMA	Auto Regressive Moving Average
BA	Bisphenol A
COR	Corr_alc, see Notation
D_A	Delta_AC, see Notation
D_V	Delta_VIS, see Notation
DCS	Distributed Control System
DGEBA	DiGlycidylEther of Bisphenol A
EBF	European Batch Forum
EVOP	EVolutionary OPeration
FRIS	Flexible Recipe-Improvement System
ISA SP88	Instrument Society of America, the Working Group SP88 of the Standards and Practices Department
ITLDD	Inverse-T Lightly Doped Drain
LDD	Lightly Doped Drain
LP	Linear Programming
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
NAMUR	Normen Ausschuss für Messen Und Regeln
N_DA	New_Delta_AC, see Notation
N_DV	New_Delta_VIS, see Notation
NLP	Non-linear optimization problem with linear constraints
NNP	Non-linear optimization problem with non-linear constraints
OLDD	Oblique rotated implanted LDD

PRIMACS	PRIMAL Measurement and Control System
PRIMAL	Package for Real-time Interactive Modelling, Analysis and Learning
QP	Quadratic Programming
RSM	Response Surface Methodology
SCADA	Supervisory Control And Data Acquisition
SE	Standard Error
SPC	Statistical Process Control
SQL	Structured Query Language
SQP	Sequential Quadratic Programming
STW	Stichting voor de Technische Wetenschappen (the Dutch Technology Foundation)
TNO/TPD	Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek / Technisch Physische Dienst (Netherlands Organization for Applied Scientific Research / Institute of Applied Physics)
TPBVP	Two-Point Boundary Value Problem

Chapter 1. Introduction

1.1 Background of the Research

Batch manufacturing of higher added-value specialities has been a fast growing segment of the process industry (i.e. [petro]chemical, pharmaceutical, food and beverages, etc.) in most industrialized countries. One of the important advantages of batch plants lies in their flexibility. They can be designed to produce several types of products in the same equipment and the same pieces of equipment may be used for a variety of different processing operations. They are less expensive than continuous plants and take less time to build, and after the product has discharged they can be more easily adapted to produce other products [JUB86, ROS87, FIS90].

If one focuses on how batches are being produced, the ever returning common factor is the use of recipes. Recipes specify products and prescribe how products are to be produced. If one looks critically at the way recipes are being used within the process industry, one finds that they are actually unnecessarily inflexible and, in consequence, often not as efficient as they could be [RIJ91]. Different feedstock properties, changes in quality specifications, variations in process behaviour, new market conditions, additional practical experiences with the process and so on, are not reflected in the recipes, though it would often be profitable to adapt them to the changed conditions. New products and processes add an extra dimension (time-to-market) to the above perspective.

In fact, because the fundamental goal of an enterprise is to make profit, economical process optimization was, is and will still be a major topic in the process industry. Process optimization to reduce, among other things, the consumption of feedstocks and energy, and the production of waste materials, is also of importance in connection with environmental protection and in this sense business and environmental interests may coincide to a great extent.

In summary, in view of recent trends, the process industry has to cope with the following problems [VER94b]:

a) more short-term dynamics in supply and end-product markets as well as more unpredictable and turbulent demand patterns;

b) more complicated processes which may be more difficult to operate;
c) short series of the manufactured products;
d) stricter requirements on product quality;
e) greater emphasis on shorter and more reliable production time;
f) a growing number of product grades and brands;
g) a need for improved customer service level.

In practice, recipes are often only approximately adjusted to the actual process and market situation. Experienced operators develop and apply their own "feel" for the process even though this deviates from the formally prescribed procedures. This informal learning process builds an insight which is often important for efficient process operation, especially when handling exceptions. However, all too often this insight is gained through trial and error, which gives no guarantee that the "best" solution is being found in a reasonable time, if at all.

1.2 Research Objectives

From the above it will be clear that batch plants require the development of special techniques supporting recipe development and next, recipe adjustment during processing. This is the starting point for developing the approach described in this thesis. The solution is, on the one hand, a better exploitation of the data generated by the process, and, on the other hand, making the process generate data that may be needed for its improvement. In all probability the proposed methods will also be applicable for continuous processes.

This thesis describes a methodology and a coherent collection of techniques for systematic and efficient recipe generation, improvement and execution in batch processes by means of so-called recipe adaptation sets, together with a supporting software system. Experiment design, mathematical modelling, statistics and optimization are at the basis of the presented approach, called the FRIS- (Flexible Recipe Improvement System) or flexible recipe-approach.

In the proposed approach two main activities can be distinguished.

Firstly, during the development of a recipe adaptation set, future batch runs are suggested which make the plant generate the necessary data. Next, these data are used in either a sequential or model-based fashion for improving process performance, which is to be stated explicitly in terms such as product quality, or profit, added value per unit of time or batch duration.

Secondly, the developed recipe adaptation set is used for improving any particular batch by compensating for known deviations in the beginning of the batch and during processing.

1.3 Contents of this Thesis

The thesis is structured as follows. In Chapter 2, the proposed FRIS-approach (Flexible Recipe-Improvement System) and the basic terms **recipe adaptation** and **recipe adaptation set** are defined and explained. Chapter 3 discusses a procedure for developing recipe adaptation sets using techniques from the area of design of experiments and statistical modelling. The approach is illustrated by an example of a simulated fermentation process. Particular attention is paid to the design of experiments for processes having time-dependent parameters, such as temperature or pressure as a function of time. Next, the recipe adaptation set is used for recipe generation or improvement. Chapter 4 discusses how a recipe adaptation set can be used for batch initialization and for correction during the execution of a batch. The methods associated with the presented approach were implemented in the software package FRIS, which facilitates the application of the approach to industrial R&D as well as production processes. This package is outlined in Chapter 5. In Chapter 6, the application of the approach is demonstrated by one industrial and one laboratory case. Finally, the achievements of this study are summarized in Chapter 7.

At the end of the thesis, a list of literature is given. References to the literature are indicated in the text by a three-letter-two-number code with the first three letters of the author's name and the year of publication.

Chapter 2. The Flexible Recipe-Improvement System

2.1 Recipe Types According to the ISA-S88 Terminology

2.1.1 Introduction

In the past, a need for standardisation of the batch-production terminology was recognized, in first instance by the chemical industry. As a result, in 1988 the Instrument Society of America (ISA) started a project group SP88. Almost at the same time, various organizations initiated similar activities in Europe, e.g. the NAMUR organisation in Germany, the ISA Netherlands Batch Working Group in the Netherlands and similar groups in other countries, all striving for the definition of a standard terminology for batch processing and batch control systems. The European organisations have come together in the European Batch Forum (EBF) and became an informal consulting body for ISA SP88. The final report of a batch terminology has recently been published and has been declared as the standard ISA S88 [ISA95].

Although, in our opinion, the ISA-terminology still has a lot of shortcomings, we shall try to utilize its terms as far as possible. Where appropriate, we will present some comments of our project group contributed to ISA SP88 and introduce a variety of extensions that are necessary in view of the new concepts introduced in this thesis.

2.1.2 Recipe Structure

According to the ISA S88-terminology, recipes contain the following categories of information: *header, formula, equipment requirements, procedure,* and other information concerning e.g. plant, process and economical constraints [ISA95]. The *header* comprises the administrative information such as the recipe and product identification, the version number, the name of the author etc.
The *formula* is a category including process inputs, process parameters and process outputs.

A *process input* is the identification and quantity of a raw material or other resource required to make the product.

A *process parameter* details information such as temperature, pressure and processing time.

Finally, a *process output* is the identification and quantity of a material and/or energy expected to result from one execution of the recipe.

Equipment requirements constrain the choice of the equipment that may be used for processing.

The *recipe procedure* defines the sequence of actions for carrying out the process.

2.1.3 Recipe Types

The ISA SP88 Committee has defined four levels of recipes that can be found in an enterprise, namely: *general, site, master*, and *control recipe.*

A *general recipe* defines a product and provides global information that is needed for the production, but without detailed specification of the equipment to be used.

A *site recipe* is specific to a particular site. It is usually derived from a *general recipe* to meet the requirements found at a particular manufacturing location, e.g. local feedstocks, units of measurement, language, etc. It still does not specify a particular set of process equipment.

In our opinion, the distinction of a *site recipe*, as defined, is impractical. Why should any recipe be site-specific, if the similar equipment is being used in another site? A well-integrated business unit shares knowledge and experience, hence strives for uniform recipes. Instead of *site recipe*, it makes more sense to have a term such as "plant-type recipe". This would also reflect the possibility to make the same product(s) in plants on a different scale, for instance in a multi-purpose plant when the demand is low, and in a dedicated plant when the demand is high.

In the context of the FRIS-approach, this type of recipe, as well as the *general recipe*, will not be used.

ISA SP88 defines a *master recipe* as a recipe which is equipment-dependent and which provides specific and unique batch-execution information describing how a product is to be produced in a given set of process equipment.

Finally, a *control recipe*, starting as a copy of the *master recipe*, contains detailed information for minute-to-minute process operation of a single batch.

It is at this level and just above, that this new standard seems already to be old-fashioned because, in spite of repeated attempts to alert its authors, it fails to adequately recognise the innovations described in this thesis. The next section introduces the basic concepts and the new perspectives they open, notably our approach using model-based or experimental optimization, together with the definition of an intermediate recipe: the **master control recipe**, and five different types of *control recipes*.

2.2 Recipe Adaptation

2.2.1 Further Comments on ISA-S88

In the framework of this thesis, it turns out to be useful to introduce the concept of a **master control recipe**, to be positioned between the *master recipe* and the *control recipes*, like a *master recipe* valid for a number of batches, but adjusted to the actual conditions, e.g. to actual prices or quality requirements, from which the individual *control recipes* per batch are derived. For what is the use of starting *control recipes* "as a copy of the *master recipe*" [ISA95, p.38] when the conditions correspond no longer to those under which the *master recipe* was derived?

Furthermore, at the start of a batch the initial conditions may differ from those prescribed by the *master recipe*, possibly even to the extent of making a successful completion unlikely. Examples are deviations in dosages, temperature, catalyst activity, equipment fouling and even available processing time. In these cases the FRIS-approach makes it possible to alter the still-adjustable process conditions, like reactor temperature and pressure, catalyst addition and maybe also reaction duration, so as to ensure the most successful completion of the run. This is called **(batch) initialization**, which implies the introduction of a new *control recipe*: the **initialized control recipe.**

After that, deviations may be detected during the batch run, and again these may be compensated for, at least partly, by application of the FRIS-approach leading to yet another recipe: the **corrected control recipe**.

In order to be able to derive those new kinds of control recipes, some suitable kind of **process model** must be available. Chapter 3 discusses in detail how to obtain such a model; for the time-being it suffices to note that it may be obtained from available process data, but that it is preferable to make the process generate more useful data by operating it systematically under a number of conditions that differ slightly from the nominal settings so as to allow statistical analysis of the results leading to a better model. The recipes for those runs will be called **experimental control recipes**. In contacts with plant personnel it is considered preferable to avoid the term "experimental" and refer to such runs as "test runs", which is commonly used to refer to runs that have to be carried out particularly carefully.

Thus, we have now introduced three different types of *control recipes*; the *control recipes* of the remaining runs, that really went according to "the copy of a *master recipe*", we shall call **routine control recipes**.

For monitoring and archiving purposes it is useful to retain a post processing record of what happened, which may be called the **accomplished control recipe**.

We have repeatedly tried to alert the ISA SP88 Committee that there is an important need to make a distinction between different phases of the *control recipe*, which in first instance was very rigidly defined by ISA. Unfortunately, the distinction between these different recipes is not clearly made in the ISA-standard and as a result it completely ignores the new possibilities of improving batch operation that will be described below. All that our comments achieved was that the ISA SP88 Committee has adopted that a *control recipe* starting as a copy of the *master recipe* may be subsequently modified before and during the processing according to scheduling, equipment and operational information. The following examples of possible modifications can be found in the ISA-standard:

- defining at the moment of batch initialization and later during processing, unless defined before, the equipment that will actually be used;
- adding or adjusting process inputs and parameters based on an "as-charged" raw materials quality or on run-time analysis;
- changing the actions described by the *procedure* based on some unexpected events.

2.2.2 Recipe Adaptation

As mentioned in the preceding section, it makes sense to insert a *master control recipe* between the *master recipe* and the *control recipe* as defined by ISA S88,

and to distinguish between the different kinds of control recipes.
To generate these various recipes and, if desired, to improve the *master recipe*, two new components are needed, i.e.

1) information from which the new recipes can be derived;
2) one or more procedures for deriving those new types of recipes.

2.2.2.1 Information: the Recipe Adaptation Set

The information needed takes the form of a **process performance** measure, often called **criterion**, which may be regarded as an economic (or quality) model, and further at least one **process model** together with additional information that will be specified later on.

The **process performance** measure or **criterion** should enable us to judge, as the term suggests, how well the process works in quantifiable terms, depending on product properties and quantity, value of feedstocks and utilities, batch duration etc. A useful measure may be in $ per batch, $ per month or, if quality or its variance is the prime concern, in a commensurate quantity.

The **process model** should express all process variables needed for calculation of the criterion values in terms of processing conditions, in particular those that are adjustable, so that it become possible to calculate that setting of the adjustables that results in the best criterion value, given the other processing conditions.

This sketches the crux of the Flexible Recipe-Improvement System: using a process model to find the operation of the process that produces the best performance under any circumstances. In addition, the very same approach is useful in the area of **(master) recipe improvement**, whether in R&D or in production: once "the right" criterion has been agreed upon, and a suitable model has been derived from the process data already available and/or the experimental test runs mentioned earlier, the FRIS-approach will ensure the fastest evolution from the existing recipe to the best one under the given circumstances, and subsequently greatly facilitate finding optimal recipes under different circumstances and/or other definitions of process performance.

The criterion and the model, together with the necessary additional information specified in Section 2.3.2, make up what is called the **recipe adaptation set**.

2.2.2.2 Procedure(s): the Recipe Adapter

The procedure for utilizing the information in the recipe adaptation set is called the **recipe adapter**. In essence, it is a procedure optimizing the criterion by manipulating the adjustable process settings, taking all relevant requirements and constraints into consideration. It is also the "machine" that may be used to produce a *master recipe*. From time to time it can also be used to replace the existing *master recipe* by one that is better adapted to the prevailing circumstances, for example new feedstocks and other market requirements; in this manner the best *master control recipe* for any number of similar batches can be found. Further, if the circumstances deviate from its prescriptions, the recipe adapter gives best possible initialized and run-time *control recipes* for each individual batch.

2.2.2.3 The Scope of Recipe Adaptation

Before we move on to the components of recipe adaptation sets, we must insert a note about its **scope**. In order to avoid misinterpretation, it must be emphasized first of all, that the FRIS-approach does not necessarily deal with the whole batch processing train or line, but may focus on those process phase(s) that most strongly influence the overall performance. Hence, it will often be limited to the most important reaction phase and ignore "secondary" operations such as dosage, heating, cleaning etc. insofar as these really have only a minor influence or none at all. This is not a drawback of the approach but its strength, because it focuses on the principal issues(s). In multistep-reaction processes, the FRIS-approach may be applied to a number of successive phases so as to achieve the best overall performance. An example is an application in DSM Resins concerning the preparation of a powder resin in the two-step reaction process, after which, in the third process step, the product was used to prepare a powder coating, which in the final step was applied to metal objects to test the quality of the coating [SME95].

2.3 Components of a Recipe Adaptation Set

2.3.1 Performance Criterion

A recipe adaptation set, the corner-stone of the FRIS-approach, may be developed in laboratory experiments, pilot plant operation, and/or during normal production by a systematic introduction of acceptably small changes in certain process inputs and parameters. It can be used to find improved process conditions to cope with variations in process and/or market situations. The search for these improved conditions proceeds in the context of the optimization of a relevant economic, quality or other criterion, which reveals how "the best performance" may be achieved. Examples of such best performance criteria include:

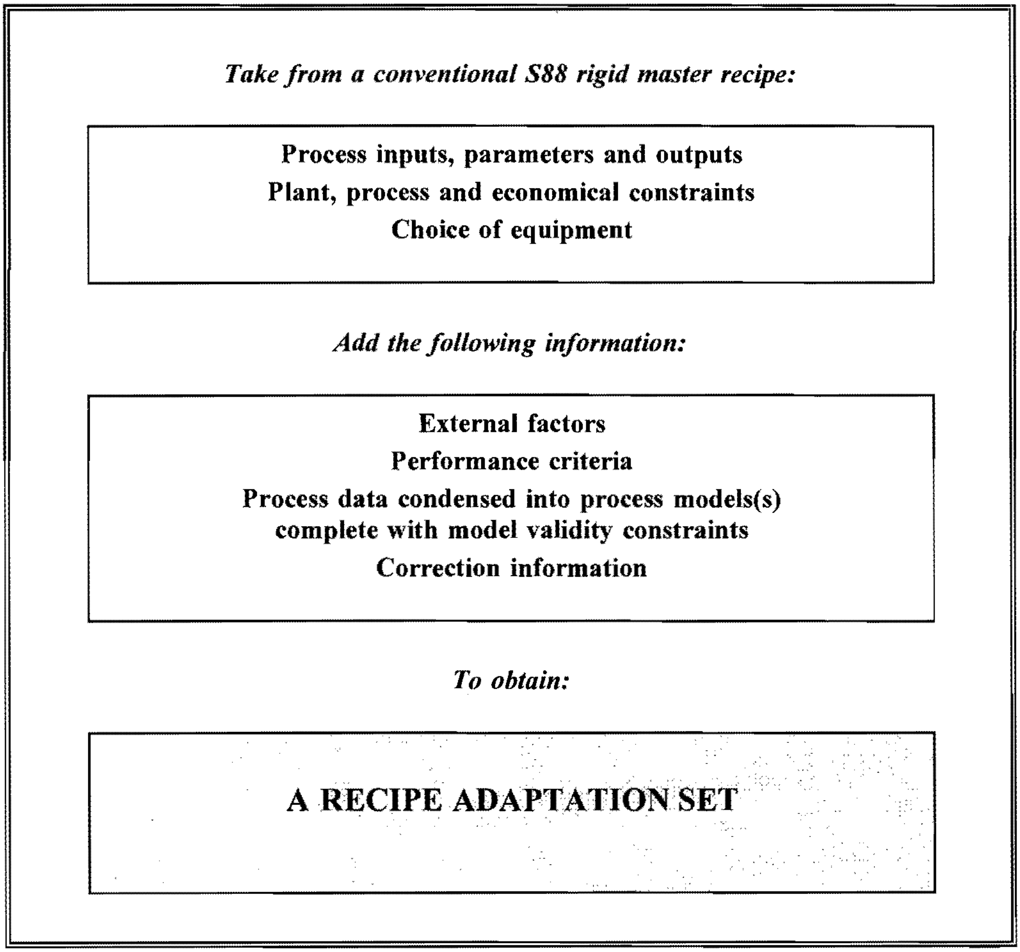
- highest product quality;
- smallest variations in the end product;
- shortest production time;
- highest production rates;
- highest added value per unit of time or per batch;
- lowest costs of e.g. feedstocks and energy;
- lowest environmental pollution;
- a formula for calculating the most profitable combination of quality, cost and quantity.

To make calculation possible, the performance criterion included in the recipe adaptation set must be expressed as a mathematical formula. (Section 3.3.1 presents some typical forms.) This may sound easier than it really is. Our experience is that the formulation of a suitable economic performance criterion is often a difficult and laborious matter: the process people talk about quality and quantity rather than about profit. Sometimes it is even so, that the aim of recipe generation or improvement is to find a region in the factor space in which end specification conditions are satisfied. In other words, no performance criterion is defined but just a list of e.g. quality requirements. The actual aim is to find at least one single point in the desired space of adjustable parameters, namely at least one recipe, which satisfies the end specification. If no criterion is formulated, mathematical optimization can not be performed. For such problems we have developed the so-called triplet-choice multi-objective method, which is described in Section 3.6.3.2, and the end-specification approach. An application of the latter to an industrial process is described in Section 6.1.2.

2.3.2 Other Components

A recipe adaptation set is always associated with a *master recipe*, which is already developed or which is in the development phase at the moment. It is obtained, as presented in Figure 2.1, by taking *process inputs*, *parameters* and *outputs*, as defined in Section 2.1, plant, process and economical constraints and equipment requirements from the corresponding *master recipe*, and supplementing them by external factors, performance criteria, process data condensed into process model(s) and the so-called correction information (see below).

It should be mentioned that the FRIS-approach pays much attention to handling constraints in an efficient manner. A very important distinction is made between constraints, which **cannot** be exceeded and constraints, which **should not** be exceeded. The latter type may offer lucrative opportunities: if the solution of the optimization problem lies on the boundary of the permitted area, then it can be profitable to shift the appropriate constraint(s) a little. If the promising shifting is allowable, then the allowed operating area may be expanded.



Take from a conventional S88 rigid master recipe:

Process inputs, parameters and outputs
Plant, process and economical constraints
Choice of equipment

Add the following information:

External factors
Performance criteria
Process data condensed into process models(s) complete with model validity constraints
Correction information

To obtain:

A RECIPE ADAPTATION SET

Figure 2.1 The change of a rigid master recipe into a recipe adaptation set

To be more specific, the components of a recipe adaptation set are as follows:

1. adjustable or at least measurable *recipe items* that may possibly affect the result of the process, i.e.: *process inputs* (e.g. dosage of ingredients), *process parameters* (e.g setpoints of reactor pressure and temperature controls, which may actually be varied during the run) and any **external factors** (e.g. quality of feedstocks);
2. *process outputs* (e.g. product quality, energy consumption, yield);
3. relevant plant, process, and market constraints;
4. at least one performance criterion;
5. values (often prices of ingredients and products) of variables used in the performance criterion;
6. **nominal process model**, with its validity constraints;
7. any available **auxiliary models**, with their validity constraints;
8. **correction information**;
9. **unit-configuration.**

It may be useful to specify the meaning of a few new terms here.

External factors are factors other than process input and parameters, which may also affect the process, e.g. quality of used materials or catalyst activity, cooling jacket or coil fouling, ambient temperature.

Nominal process model: In this thesis any models associated with a recipe adaptation set are of the "black-box" type. They relate process outputs to selected recipe items. The main process model is called a "nominal process model". Such a model should be valid around the prescribed ("nominal") process conditions. If present, the other models for neighbouring operating conditions form the group of **auxiliary models**. These models can become the nominal model, if the search for the best recipe leads to conditions for which the initially nominal model is not valid. The FRIS-software checks continuously model validity and automatically selects right models.

The **correction information** is necessary to correct a batch during processing. It includes the information about the choice of sample and correction moments, possible process outputs measured at the sample moment(s), correction variables (e.g. extra ingredients, processing time) and correction models.
It will be comprehensively described in Chapter 4.

With a **unit-configuration** we mean the specification of a set of processing units and other equipment that is expected to be used in the production of a batch corresponding to the recipe adaptation set, where:

- on a single unit-configuration only one batch can be run at the same time;
- equipment units of a configuration may be combined in various manners; a batch does not necessarily have to use all the equipment;
- the equipment combination may be determined at the beginning of the batch;
- the same recipe adaptation set must be valid for the various equipment combinations. This means that one of two possibilities occurs:
  a) the process output does not depend on the used equipment combination (this is the most common situation);
  b) if the process output does depend on it, the used combination is included in the recipe adaptation set as an external factor, i.e. extra recipe item.

The unit-configuration may be seen as a specification defining a subset of the allowable equipment described in the *master recipe*.

2.4 The FRIS-Approach

2.4.1 The Two Activity Domains of the FRIS-Approach

In the FRIS-approach two main domains of activities can be distinguished: firstly, the development of a recipe adaptation set intended for generation of a *master recipe* or improvement upon it, and generation of an actual *master control recipe*. Subsequently, its application to process monitoring, or to the improvement of any particular batch, by generating and adjusting a best *control recipe*. Table 2.1 summarizes these activities.

As presented in Table 2.1, the FRIS-approach shows these activity domains, aiming at two, usually quite different, groups of users in the process industry, namely those in R&D and those in Production. The development of a recipe adaptation set and the development of a *master (control) recipe* are in the domain of R&D, the application of a recipe adaptation set is in the domain of Production department.

It should be added that the R&D functions will often be combined with those of Production, especially in a laboratory or a pilot-plant, or during experimentation. Nevertheless, the partition of the approach into a recipe development part and an operational application part, and the distinction of two groups of users is quite useful because:

a) in R&D, costs and time-to-market are of prime importance and in Production, best plant performance, or fastest performance improvement;

b) development of a recipe adaptation set together with a *master (control) recipe* is a relatively long-term occupation involving a number of batches, whereas an application of such a set regards, in principle, one individual batch;

c) development of a recipe adaptation set has freedom of choice as the user may decide how and at which moment test runs are to be done; recipe application, however, has to realize production requirements before the end of the individual batch run;

d) in the operational application phase, it is common practice to organize different types of batches per scheduling sequence, with the consequence that various recipe adaptation sets will be used one after the other. In contrast, recipe development utilizes similar types of batches in different scheduling sequences.

Table 2.1 Activities of the FRIS-approach

The Flexible Recipe-Approach	
1. Development of a recipe adaptation set and recipe improvement	**2. Application of a recipe adaptation set**
Aim: • Generation/improvement of a *master recipe* or a *master control recipe*	**Aim:** • Generation/adjustment of a *control recipe* (batch initialization and batch correction) • Process monitoring
Techniques: • Design of experiments • Process modelling • Model-based recipe improvement • Experimental process optimization	**Techniques:** • Recipe optimization in the context of batch initialization and batch correction

2.4.2 The Development of a Recipe Adaptation Set in Brief

The development of a recipe adaptation set is the activity which is principally done by R&D people. They may use the developed recipe adaptation set to determine the best *master recipe* in a systematic fashion, or to improve upon it, and to generate a *master control recipe*, e.g. owing to new research findings, changed prices or customer demands.
The principal techniques involved in this activity are: design of experiments, modelling, model-based recipe improvement and experimental (sequential) optimization.

Design of experiments defines the successive experiments (test runs) and provides a pattern for the introduction of variations in selected recipe items in order to maximize process information required for finding e.g. process models or optimal process conditions, in a minimal number of runs.
In essence, without upsetting normal production the process is made to produce data about itself.

During **modelling**, the parameters of a process model are estimated on the basis of the available data. For ease of discussion, it is helpful to make a distinction between "white" and "black" modelling. The former constructs the model equations on the basis of the so-called first-principle laws from physics, thermodynamics, chemistry and chemical engineering. The latter estimates the parameter values of arbitrary postulated mathematical formulae (a parameterized model) from the records of input/output data by minimizing an criterion, which gives an indication of the difference between the actual and estimated process output data. Usually, such a black-box model does not reflect the internal process structure. A more detailed description of both types of models is given in Section 3.2.1.

It should be emphasized that, actually, in practice neither "white" nor "black-box" modelling is used, but a judicious combination, resulting in more or less "grey" models. As mentioned before, this distinction is only made for ease of discussion.

Unlike white models, which are seldom available in many branches of the process industry, black-box models may be relatively easily obtained by experimentation, provided that small variations in process operation are allowed. Therefore, black-box models, more precisely "dark grey" models, of the type henceforth to be called *transition models*, are presently used in the FRIS-approach: most of the time a parameterized model is postulated in the form of simple

polynomial, and parameters are estimated from the observed process data, but process knowledge is extensively used in design of experiments, and in the selection of variables and model-structure. A *transition model* gives the relation between the initial process conditions and the final output without describing what happens in between initial and final moments, or, more generally, it relates the process conditions at one moment to the conditions at a later moment, given all important inputs to the plant in the period in between.
However, if a white process model happens to be available, it makes sense to incorporate it in the recipe adaptation set.

The **model-based recipe improvement** searches for improved process conditions, i.e. by optimizing the predefined performance criterion, subject to the estimated process model and defined constraints, using a suitable optimization method. Because the FRIS-approach employs, in the first instance, the black-box transition models, the corresponding optimization methods have static character. Section 3.4 describes what kind of optimal control problem will be obtained in case a white process model is available. That section also explains how the FRIS-approach corresponds to the optimal-control approach.

During **experimental optimization** one searches for optimal operating conditions by sequential comparison of the process data, without using mathematical models, like in zero-order optimization methods. To speed-up the classical experimental optimization, we have developed a new method, called multiplex fitting, which proceeds towards an optimum by local approximation of the process surface.

2.4.3 The Run-Time Application of a Recipe Adaptation Set in Brief

Once defined, a recipe adaptation set can be applied in Production for batch-by-batch generation and adjustment of an improved, i.e. initialized or corrected, *control recipe*, and in **process monitoring**, which may be helpful in improving models and recipes, and in the detection of disturbances (e.g. catalyst deactivation, fouling, changes in feedstocks). The principal mathematical technique involved in generation and adjustment of a *control recipe* is recipe optimization in the context of batch initialization and correction.

With **batch initialization** we mean generation before the process is actually started, of a *control recipe* adapted to detected deviations in process and/or market conditions, whereas **batch correction** deals with adaptation of a *control recipe* during the batch run, based on measured deviations.

To sum up, the strength of the FRIS-approach is that it improves the efficiency of determination of the best *master recipe* and, owing to the separation between the process model and the market model, it allows for an almost instantaneous improvement upon the *master recipe* to rapid economic changes without process remodelling, i.e. the generation of the best actual *(master) control recipe*. Furthermore, the approach enables repeated utilization of the recipe adaptation set to find (near-) optimal *control recipes* under varying market and process conditions.

Chapters 3 ÷ 4 describe in more detail the concepts introduced on the basis of a fermentation process.

Chapter 3. Development of a Recipe Adaptation Set and Master Recipe Improvement

A recipe adaptation set, as defined in Chapter 2, includes two kinds of components: conventional ones and flexible model-based ones. The development of a recipe adaptation set always starts with the definition of the conventional recipe components presented in Figure 2.1, that is, relevant process inputs, process parameters, and other components such as process outputs, plant, process and market constraints, and choice of equipment. For practical reasons recipe adaptation does not cover all process inputs and parameters. Rather, one should select those items, which are expected to have a strong impact on the criterion and/or constraints that may turn out to play an active role.

The second step is to investigate whether the external factors, like the quality of feedsctocks or the ambient temperature, have an effect on production results. If this is the case, it is recommendable to measure them and to incorporate them in a recipe adaptation set.
In the next step a performance criterion and constraints should be formulated. This is a subject of Sections 3.3.1 ÷ 3.3.2.

Other essential parts of a recipe adaptation set are process models. As already mentioned in Section 2.4.2, two kinds of process models can be distinguished: white (first principle or mechanical) models on the one hand, and black-box (empirical) models on the other.

Before both types of models will be described, we will concentrate on the methods of design of experiments, which support the development of black-box models of the transition type.

As pointed out in Chapter 2, a recipe adaptation set can be used in generation and improvement of a master recipe or a master control recipe, and subsequently in generation and adjustment of a control recipe. This chapter describes the former activities, Chapter 4 the latter.

At this point it should be mentioned that, upon request, this chapter has been written in such a way that it may also serve as an introduction to the FRIS-approach for those who want to put it into practice themselves.

3.1 Design of Experiments

3.1.1 Introduction

The field of "design of experiments" has its own terminology: any process input or parameter selected for experimentation is called a **factor**, a measured process output is called a **response**. For the remainder of this thesis, in the context of experimentation, these terms will be used.

Because experiments are time-consuming and may interfere with normal production, the importance of an efficient design procedure for the planning of experiments can not be stressed strongly enough. Experiments can be planned and next carried out in one of the following manners:

1) as unstructured experiments: one tries to experiment for various process conditions, which are chosen rather arbitrarily;
2) as "one-factor-at-a-time" experiments: during one experiment only the value of one factor is changed;
3) according to any customary design of experiments scheme, which will be described below.

The most important advantages of design of experiments in comparison with one-factor-at-a-time or unstructured experiments are:

1) efficiency is greater, because the aimed objective can be realised in a relatively small number of experiments;
2) the procedures are structured so that they are clear in execution and reporting;
3) the data lend themselves better to modelling and in the continuous improvement of the process operation.

The concepts of Response Surface Modelling (RSM) are very useful in design of experiments [BOX87]. This is a strategy intended to estimate a process model and to use it to predict response(s) and to optimize performance. This strategy forms an essential part of the proposed model-based flexible recipe-approach.

The selection of factors is an important step. Some rules of thumb can be given:

1) try to select not too many factors, e.g. not more than four: the growing number of factors means an even stronger growth in the number of experiments;
2) because of noise, choose the factors' settings not too close together (see Formula (3.1) below);
3) choose the factors settings not too far apart, so that a linear or quadratic approximation of the measured effect may adequately describe responses in the corresponding area;
4) take into account that possibilities for experimentation during commercial production tend to be more limited than those in comparison with a pilot plant, so the allowed variations in a commercial plant are usually smaller than in a pilot plant.

If, in first instance, more than four factors are selected, it is recommendable to use the so-called Screening Design method (e.g. Plackett-Burman [PLA46], Upperman [UPP93]) or the fractional factorial design [BOX87, MON91] to identify the most important ones. In this manner the number of experiments can be reduced.

The Principal Components Analysis can be helpful to investigate whether the involved factors are correlated. In such a case, the variations in the original factors can be accounted for by a smaller number of so-called principal components [MAN94].

It is important to realize that the effect of unknown influences, like e.g. catalyst ageing, may be reduced when the experiments proposed by the design method are done in a random order. Because the response measurements are noisy, it is necessary to know how many experiments, performed under the same process conditions, so-called replicates, are needed. An often used - rule of thumb is [BAN94]:

$$r \geq 8 * \sigma^2/(\bar{y}_1 - \bar{y}_2)^2 \qquad (3.1)$$

where:

r : number of experiments in the same point

σ^2 : variance of the measured response

$\bar{y}_1$, $\bar{y}_2$: average response for the first and second process conditions, respectively.

Among the established design of experiments methods, factorial-like and optimal designs are most suitable to be used to investigate multifactor response surfaces. There is a lot of literature on this subject [e.g. BOX87, MON91, BAN94], which can be consulted for a comprehensive description of the methods; a brief description is given Appendix A.

There are also many software packages supporting design of experiments, e.g. RS/Discover, STATGRAPHICS, SAS, ECHIP, but the expertise for using them in an industrial environment is not widely available, and their use provides no more than individual pieces of a puzzle instead of solving the puzzle as a whole. The development of the FRIS-approach is an attempt to change this situation.

3.1.2 Design of Experiments for Processes Having Time-Dependent Parameters

It should be noted that batch operation can often be improved by varying process parameters such as temperature or pressure as a function of time. Classical design of experiments methods have been developed for time-independent factors only. When a batch process involves time-dependent recipe items, the search for the optimal profile of such recipe item would require dynamic optimisation, for which purpose a dynamic process model is necessary, which is rarely available. Fortunately, in many batch-processes time-dependent recipe items can be approximated by simple, often piecewise linear or lower-order polynomial (spline) functions of time, with or without constraints [RIJ91, VER95]. In the FRIS-approach, a simple and practical way of improving process operation is by such an approximation of time-dependent recipe items.

Let $\mathbf{u}(t;t_0,t_F)$ be a vector of time-dependent recipe items on the interval $[t_0,t_F]$. Each element $u(t;t_0,t_F)$ of this vector can be approximated by:

$$u(t;t_0,t_F) \cong \mathit{u}(t;t_0,t_F,\mathbf{x}) \qquad (3.2)$$

where:

u : approximation of u on the interval $[t_0,t_F]$, often in the form of a piece-wise linear, not necessarily continuous, function;

$\mathbf{x}$: parameter vector associated with the approximating function; the elements of this vector serve as factors (recipe items) during experimentation (where confusion is unlikely, the parameter $\mathbf{x}$ may be omitted).

For example, this means: a time-dependent recipe item u(t;t_0, t_F) may be approximated on the interval [t_0, t_F] by a linear function u(t;t_0,t_F,**x**) of time, where **x** denotes a vector with the initial level u(t_0) as the first element, and the slope of the line as the second element. In such a case the search for an optimal profile of the time-dependent recipe item is replaced by a search for the optimal values of the parameters **x** and so this problem can be solved with the known design of experiments methods. This will be illustrated by an example.

Let us consider a simulated fed-batch fermentation process (see Appendix B and [OVE92]). The micro-organisms produce penicillin in a reactor, provided that during the reaction sugar is added. It is known that the temperature for the maximal growth rate of the micro-organisms is higher than the temperature for the maximal rate of product formation. It is not known, however, what temperature profile in the batch reactor has to be chosen so as to obtain the maximal amount of the final product, denoted by Pf. If one decides to find this profile T(t;t_0,t_F,**x**) by design of experiments, one first has to choose a suitable form of u(t;t_0,t_F,**x**).

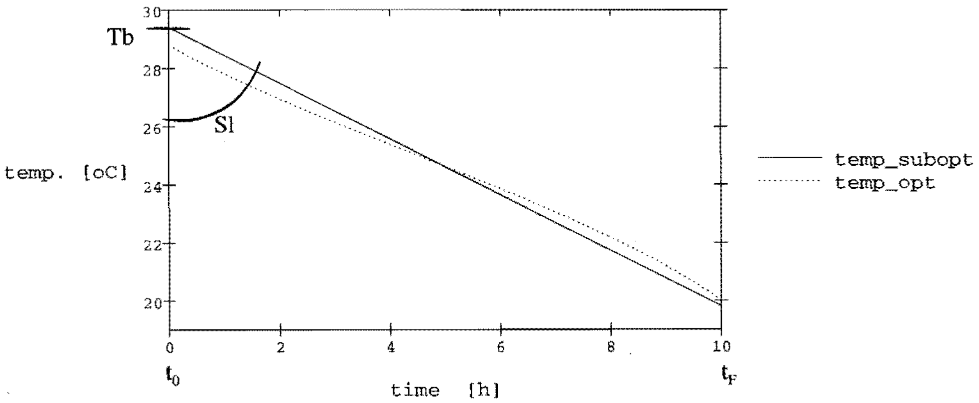


Figure 3.1 An optimal temperature profile of the fermentation process temp_opt (dotted line) can be approximated by a suboptimal linear profile temp_subopt (continuous line) with the initial temperature Tb and the slope Sl.

In Figure 3.1 the optimal temperature profile is shown by the dotted curve; of course, usually this profile is seldom known beforehand and will never be known if you proceed as intended. In the examination of this example, this profile was actually obtained by dynamic optimization of the simulated process. For that purpose the so-called two-point boundary value problem TPBVP based on a white process model had to be solved as shown in Appendix B.

A simple candidate for a sub-optimal approximation is:

$$T(t;t_0,t_F,\mathbf{x}) = Tb + Sl\ (t-t_0) \qquad \text{for } t \in [t_0,\ t_F] \qquad (3.3)$$

where:

$\mathbf{x}$ = [Tb, Sl]
Tb : initial temperature [°C]
Sl : temperature slope [°C/h].

If the initial temperature value Tb and the slope Sl are varied according to an design of experiments method and the response Pf is estimated as the function of these factors, then the performance index can be optimized and the best approximation of the optimal profile can be found.

In this case, experiments were simulated, in which the temperature was varied according to the Box-Behnken design method (see Appendix A), with Tb and Sl as factors under the following constraints:

$$27 \leq Tb \leq 30 \qquad (3.4)$$
$$-1 \leq Sl \leq -0.5. \qquad (3.5)$$

The estimated process model was found to be:

$$Pf(t_F{=}10\ h) = -752.629 - 61.002\ Tb - 287.512\ Sl - 12.199\ Tb \cdot Sl - 1.220\ Tb^2 - 36.986\ Sl^2 \qquad (3.6)$$

The maximization of the production yield Pf gives: Tb = 29.41 and Sl = - 0.96. The linear sub-optimal profile (3.3) with the coefficients found by the optimization of (3.6) is indicated by the continuous line in Figure 3.1. It may be added that the sub-optimal yield, corresponding to this linear approximation, is 40.87, a result which is only 0,4% less than the true dynamic optimum of 41.04.

The optimal profile is hardly ever known beforehand. Experience and knowledge of the process may then be helpful; indeed, on the basis of such experience, a suitable approximation may be proposed, for example like (3.3). Next, one may try to extend the first approximation with one having more degrees of freedom, of course provided experimentation according to such a more "extensive" approximation is not too time-consuming.

In this particular example, the second approximation attempt was as follows:

$$T(t;t_0,t_F,\mathbf{x}) = \begin{cases} Tb & \text{for } t \in [t_0,\ t_1] \\ Tb + Sl\ (t - t_1) & \text{for } t \in [t_1,\ t_F] \end{cases} \tag{3.7}$$

where:

$\mathbf{x}$ = $[Tb,\ Sl,\ t_1]$

Tb : initial temperature, held fixed in the time-interval $[t_0,\ t_1]$

Sl : temperature slope after t_1

t_1 : time break-point in the approximation.

In this case the optimization of the response Pf according to the new estimated process model of these three factors resulted in $t_1 \cong t_0$. Therefore it was safe to conclude that the optimal profile may be approximated by (3.3) [MEI94].

3.2 Process Modelling

3.2.1 Characteristics of White and Black-Box Process Models

This sub-section presents the most important characteristics of white models as well as black-box models.

For white models:

1) the development of such models can only be done by experts and, moreover, is usually very time-consuming;
2) often they consist of many hundreds or even thousands of differential and/or, possibly non-linear, algebraic equations, which are time-consuming to solve. Of course, simpler models may sometimes be obtained by making assumptions concerning the process, but in such a situation it may be not evident how these simplifying assumptions influence model adequacy;
3) they have a relatively large validity region, depending on the assumptions

made during the model development, but often they are not very accurate;

4) they do not only describe the input-output relations, but also the "inner behaviour" of the process, and all model parameters and variables have, in principle, a clear physical meaning.

For black-box models:

1) the development of such models is based on experimental data and not on first principles;
2) often model equations may be quite simple;
3) the model is valid only in the region between and around the performed experiments and any extrapolation can be risky. On the other hand, the model can be as accurate in this small area as the accuracy of measurements permits;
4) because these models describe input-output relations only, they say nothing about what happens inside during processing.

From here on, whenever process models are mentioned, transition models of the black-box type are meant. To avoid confusion, white models will be always called "white" or "first-principles models". How they may be used for recipe improvement will be discussed in Section 3.4.

3.2.2 The Introduction of a Transition-Model Development Scheme

For process modelling, the following transition-model development procedure was developed in the context of this thesis (see Figure 3.2):

1. Define the purpose of the experimentation and modelling;
2. Investigate the available process knowledge;
3. Define an experiment design task;
4. Perform experiments;
5. Estimate model parameters;
6. Conduct statistical tests to check model adequacy; if necessary, apply model reduction or extension, input/output transformations and/or robust regression and return to 5; if statistical techniques do not give satisfactory results, define new (extra) experiments and return to 3 or give up;
7. Investigate the tentatively accepted process model: visualize its response surface, compute the effects of factors and their confidence intervals, predict responses under various circumstances to learn more about the model, and possibly also about the process.

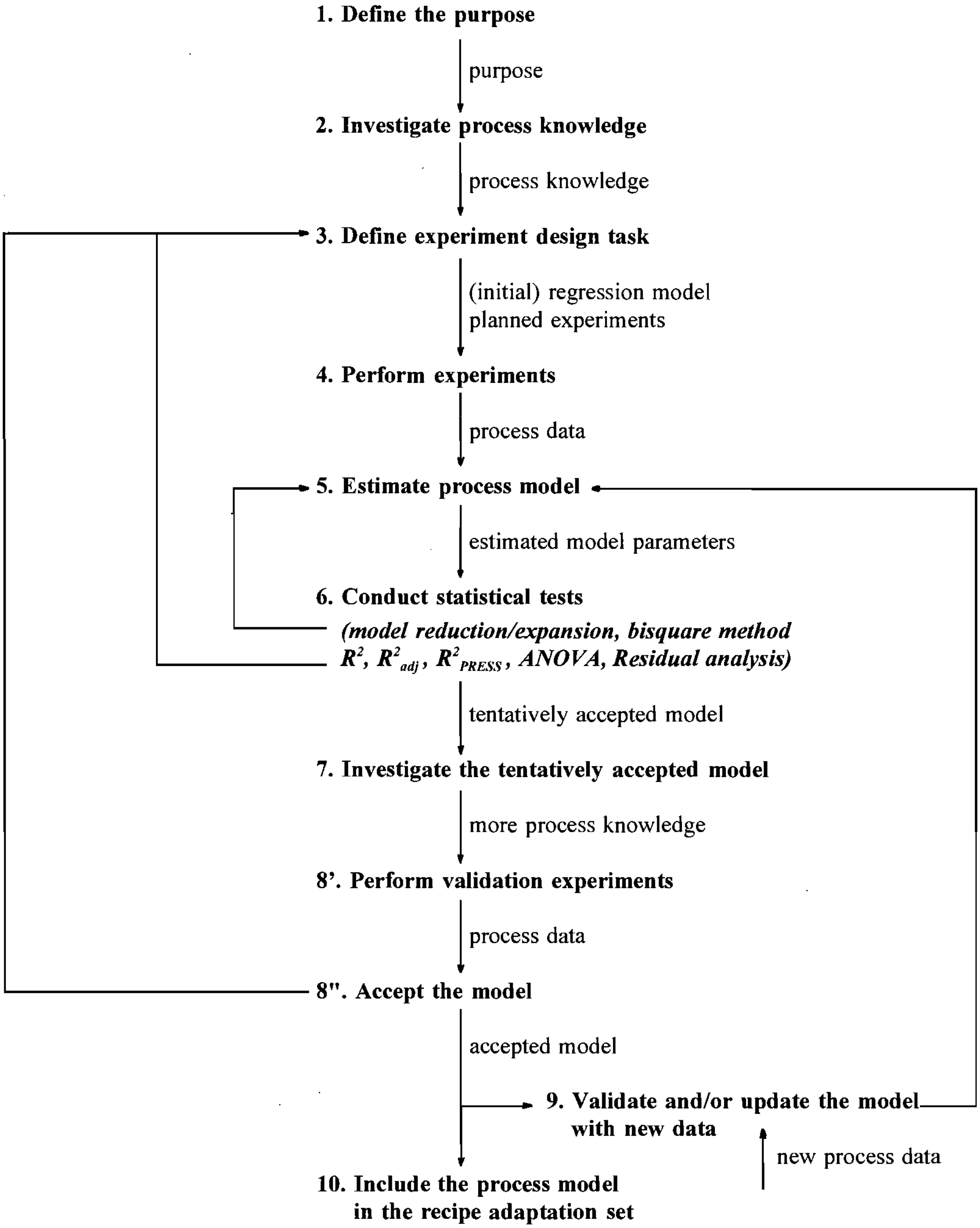

1. Define the purpose
purpose
2. Investigate process knowledge
process knowledge
3. Define experiment design task
(initial) regression model
planned experiments
4. Perform experiments
process data
5. Estimate process model
estimated model parameters
6. Conduct statistical tests
(model reduction/expansion, bisquare method
R^2, R^2_{adj}, R^2_{PRESS}, ANOVA, Residual analysis)
tentatively accepted model
7. Investigate the tentatively accepted model
more process knowledge
8'. Perform validation experiments
process data
8". Accept the model
accepted model
9. Validate and/or update the model with new data
new process data
10. Include the process model in the recipe adaptation set

Figure 3.2 The transition-model development scheme

8. Define and perform validation experiment(s) to validate and accept the model; if a model is not valid, define new (extra) experiments and return to 3;
9. Use all batch process data for model validation and/or model updating, so as to improve the model;
10. Include the validated process model cum annexis in the recipe adaptation set.

3.2.2.1 Defining the Purpose of the Experimentation and Modelling

This would appear to be a rather obvious part of any experimentation, but in practice it is by no means always clear at once to everybody involved. As already mentioned in Chapter 2, some possible goals are:

- improving process yield;
- determining what recipe settings will ensure that the desired end specification is reached;
- reducing production costs;
- reducing processing time.

3.2.2.2 Investigating the Available Process Knowledge

This is a very important part of modelling.
Below a set of standard questions is given to support knowledge structuring.

1) What are important process, plant and economical constraints?
2) Can the various factors be varied independently from each other? If recipe items are correlated, apply Principal Components Analysis first. In the case that the factors are concentrations in a mixture prepare so-called Mixture experiments [MON91];
3) Does the process in question involve time-dependent parameters?
4) What are the allowed variations for each factor?
   What, if any, complications are to be expected?
5) Are there any factor interactions possible?
6) How accurate are the measurements?
7) How many repetitions of an experiment under the same process conditions are needed?

The proper understanding of the process information may help design better experiments.

3.2.2.3 Defining an Experiment-Design Task

An experiment-design task is a set of experiments defined to realize the specified goal. Because this issue is very extensive, it was separately described in Section 3.1 which showed that the selection of an appropriate experiment-design scheme also involves a choice of the factor variation size, the number and type of experiments, and the type of initial regression model. As will be shown below, transition models are built using multiple polynomial regression (the response is regressed on a number of factors together with their powers and products), but the choice of the appropriate initial regression model is not a statistical issue, it should rather be based on process knowledge and the experience obtained during previous experimentation.

3.2.2.4 Performing Experiments

Experiments should be performed very carefully to obtain good results so as to avoid *un*necessary replication of experiments. As will be shown below, repetition of experiments is necessary to obtain an estimate of the experimental error and, as a consequence, a more precise estimation of the effect of the factors.

Moreover, because the statistical methods involved in the data analysis require that the observation errors are independently distributed random variables, it is recommendable to perform experiments in random order, as mentioned in Section 3.1.1.

3.2.2.5 Estimating Model Parameters

As mentioned before, the relationship between the process response and the selected "factors", i.e. recipe items, is represented by a mathematical model. After an appropriate model structure is chosen, e.g.

$$y = \mathbf{X}\,\beta + \epsilon \tag{3.8}$$

where:

y : process output, here called response

β : vector of the unknown model parameters (in the context of regression analysis called vector of model coefficients)

$\mathbf{X}$: data matrix including the column vector **1**, which is necessary when a model with a constant term is estimated

ϵ : random error with zero mean value and variance σ^2

the model parameters can be estimated from the experimental data set using appropriate statistical methods. A common estimation method is to minimize the sum of squares of the differences between the measured response values and the values predicted by the regression equation for the set of experimental data - hence the term "least-square regression analysis".

As well as providing values for the unknown model parameters, regression analysis also yields estimates of their standard deviation, so that both point estimates and confidence intervals for the factors become available, which information enables certain hypotheses to be tested, e.g. whether the values of model parameters differ significantly from zero or not.

The data matrix **X** depends on the structure of the process model to be estimated. The transition models used in this thesis are of the polynomial type, so that the **X** matrix contains the experimental values for each factor, including interaction terms and/or powers.

For a complete discussion of the least square regression method see, for example, [DRA81]. As the method is quite common, we shall concentrate on the analysis of the estimated model, rather than on the estimation procedure itself.

As already mentioned in Section 3.1.1, there are many statistical software packages with graphical support that can be helpful in modelling and data interpretation. The analyses for this thesis have been done using the RS/Discover package, which we found to be the most complete and user-friendly software in this area at the time a package had to be selected.

The modelling of the fermentation process introduced in Section 3.1.3 is used as an example. Let us assume that three factors are varied during experimentation: initial temperature Tb, temperature slope Sl and sugar dosage Sd. In first instance, the three selected factors are varied according to the factorial design (see Appendix A) in the following range:

$$Tb_{MIN} = 27, \qquad Tb_{MAX} = 30; \tag{3.9a}$$
$$Sl_{MIN} = -1, \qquad Sl_{MAX} = -0.5; \tag{3.9b}$$
$$Sd_{MIN} = 0.85, \qquad Sd_{MAX} = 1.15. \tag{3.9c}$$

Table 3.1 Worksheet with experiments. The measured response is the final product amount Pf.

Exp.	Tb	Sl	Sd	Pf
1	30.0	-0.50	0.85	26.08
2	27.0	-0.50	0.85	34.87
3	30.0	-0.50	1.15	33.18
4	28.5	-0.75	1.00	40.77
5	27.0	-1.00	1.15	38.15
6	30.0	-1.00	1.15	45.73
7	27.0	-0.50	1.15	44.70
8	28.5	-0.75	1.00	40.65
9	27.0	-1.00	0.85	30.05
10	28.5	-0.75	1.00	40.06
11	30.0	-1.00	0.85	35.91

Altogether eleven experiments are performed in random order (eight according to factorial design plus three extra experiments in the centre point), which are presented in Table 3.1. The final amount of the product Pf is the measured and modelled response.

On the basis of these experiments the following process model is estimated:

$$\begin{aligned}Pf = \; & 37.314 - 0.821\cdot(\sim Tb) - 1.339\cdot(\sim Sl) + 4.394\cdot(\sim Sd) + \\ & - 4.181\cdot(\sim Tb \cdot \sim Sl) - 0.089\cdot(\sim Tb \cdot \sim Sd) - 0.086\cdot(\sim Sl \cdot \sim Sd)\end{aligned} \qquad (3.10)$$

where:

$$\sim Tb = ((Tb - 28.5) / 1.5) \qquad (3.11)$$

$$\sim Sl = ((Sl + 0.75) / 0.25) \qquad (3.12)$$

$$\sim Sd = ((Sd - 1) / 0.15) \qquad (3.13)$$

Table 3.2 presents the accessory statistical information. It shall be used to explain what statistical analyses have to be done in order to properly arrive at an acceptable model.

The column 1 of Table 3.2 ("Term") presents the number of the factor and its symbol, the sign "~" indicating that this factor has been transformed as indicated in the column 6 ("Transformed Term"). In this case the so-called orthogonal scaling is used:

$$(x - x_M) / \delta x \qquad (3.14)$$

where:

$$x_M = (x_{MAX} + x_{MIN}) / 2 \qquad (3.15)$$
$$\delta x = (x_{MAX} - x_{MIN}) / 2 \qquad (3.16)$$

so that the value of any selected factor after such scaling is between -1 and +1.

Table 3.2 Model MOD1 for the worksheet of Table 3.1

```
   Least Squares Coefficients, Response Pf, Model MOD1

1 Term         2 Coeff.   3 Std. Error  4 T-value  5 Signif.  6 Transformed Term

---------------------------------------------------------------------------------
1 1            37.313636      1.001673
2 ~Tb          -0.821250      1.174566                           ((Tb-2.85e+01)/1.5)
3 ~Sl          -1.338750      1.174566                           ((Sl+7.5e-01)/2.5e-01)
4 ~Sd           4.393750      1.174566                           ((Sd-1)/1.5e-01)
5 ~Tb*Sl       -4.181250      1.174566      -3.56      0.0236
6 ~Tb*Sd       -0.088750      1.174566      -0.08      0.9434
7 ~Sl*Sd       -0.086250      1.174566      -0.07      0.9450

No. cases = 11         R-sq. =   0.8768     RMS Error = 3.322
Resid. df =  4   R-sq-adj. =   0.6920         Cond. No. = 1
~  indicates  factors  are  transformed.
```

It is recommendable to use orthogonal scaling during the model estimation procedure. Firstly, because such scaling helps to reduce numerical instability during the computation of the inverse of the matrix ($\mathbf{X}^T\mathbf{X}$) needed by the least squares algorithm. Secondly, if the used design scheme is orthogonal, that is: if **X** has orthogonal columns, the variance of the estimated model coefficients is minimal and the coefficients can be estimated independently, which means that if a term is removed from the model, the remaining coefficients do not change. Examples of orthogonal designs are the 2^k factorial designs and fractional designs with orthogonally scaled factors.

The standard 2^k design does not give enough information to estimate the experimental error. To estimate it, some runs have to be repeated. By augmenting the orthogonal 2^k design with a number of experiments in the centre point, the design remains orthogonal and estimation of the experimental error will be possible. This "augmented" factorial design has been used in this example.

In columns 2 and 3, Table 3.2 gives the estimated coefficient values ("Coeff.") and their standard error ("Std. Error"), respectively; the latter indicating a measure of uncertainty of the estimated coefficient, defined as follows:

$$SE(\beta_i) = \sigma\,(C_{ii})^{1/2} \tag{3.17}$$

where:

β_i	:	the ith model coefficient
$SE(\beta_i)$	:	standard error of β_i
C_{ii}	:	the (i,i)th element of $(\mathbf{X}^T\mathbf{X})^{-1}$
σ	:	estimated standard deviation of the random error ϵ.

Columns 4 and 5 in Table 3.2 give information for testing the null-hypothesis that the actual values of the coefficients being estimated are in fact zero, in which case the associated term can be removed from the model. This procedure is described in the following sub-section.

Table 3.2 further presents the Root Mean Square error ("RSM Error"), which is a measure of the response variability that is not explained by the fit:

$$\text{RSM Error} = (MS_{RESID})^{1/2} \tag{3.18}$$

where:

$MS_{RESID} = SS_{RESID} / (n-p)$	:	residual mean square	(3.19)
$SS_{RESID} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$	:	residual sum of squares	(3.20)
y_i	:	measured response of the ith experiment	
$\hat{y}_i$	:	estimated response of the ith experiment	
n	:	number of experiments	
p	:	number of model terms (including the constant term, if present)	

If the model is correct, this value gives an estimate of σ. This will become clear after the explanation of the lack of fit which is discussed in Section 3.2.2.6.

The additional information given by "R-sq." and "R-sq-adj.", presented at the bottom of Table 3.2, gives some indication for checking the fit and it is also discussed in Section 3.2.2.6.

The last item in Table 3.2, "Cond. No.", is the condition number indicating to what extent the matrix **X** is ill-conditioned. This number can vary from one, which indicates a perfectly orthogonal design, to infinity, which indicates a design containing at least one singularity, i.e. one factor being an exact linear combination of the others, as a consequence one factor must be left out of consideration. A condition number higher than 100 indicates some collinearity among factors which can cause the model parameters to be much more poorly estimated than would happen when using an orthogonal design.

3.2.2.6 Conducting Statistical Tests

To check model adequacy and, if considered necessary, to apply model reduction or extension, input/output transformation and/or robust regression, we have selected the following statistical techniques:

1) Investigation of the **coefficient of determination** and the **adjusted coefficient of determination**, denoted by R^2 and R^2_{adj}, respectively.
2) Investigation of the **significance of model parameters**.
3) **Analysis of Variance**.
4) **Residual analysis**.
5) Investigation of **the goodness-of-fit** with the χ^2-test.
6) Investigation of the **PRESS test**.

Appendix C gives a description of these generally accepted techniques.

The statistical analysis starts with an investigation of the **coefficient of determination R^2** (in Table 3.2 presented by "R-sq.") and the **adjusted coefficient of determination R^2_{adj}** (in Table 3.2 presented by "R-sq-adj."), which say something about how well correlated the fitted values produced by the model are with the actual response values, in other words they give an indication about the explanatory ability of the model. A high value of these coefficients (the maximal possible value is 1) indicate a good fit. Because the values of both statistics R^2 and R^2_{adj} for the fermentation model MOD1 are satisfactory ($R^2 = 0.8768$, $R^2_{adj} = 0.6920$), we may go on with the test of the **significance of the model parameters**. The following null hypothesis is tested for each estimated parameter β:

$$H_0: \quad \beta \overset{?}{\neq} 0$$

As one can see in Table 3.2, the "significance", actually the insignificance of the terms Tb*Sd and Sl*Sd is very high, namely 0.9434 and 0.9450, respectively, i.e. their significance levels are very low: (1-0.9434)·100% and (1-0.9450)·100%, respectively. This means that both terms may be removed from the model. The reduced model, called MOD1_RED, is as follows:

$$\begin{aligned} Pf = \; & 37.314 - 0.821\cdot(\sim Tb) - 1.339\cdot(\sim Sl) + 4.394\cdot(\sim Sd) \\ & - 4.181\,(\sim Tb \cdot \sim Sl), \end{aligned} \tag{3.21}$$

see also Table 3.3.

Table 3.3 Reduced model MOD1_RED

```
  Least Squares Coefficients, Response Pf, Model MOD1_RED

1 Term        2 Coeff.   3 Std. Error  4 T-value  5 Signif.  6 Transformed Term

-----------------------------------------------------------------------------------
1 1          37.313636     0.818997
2 ~Tb        -0.821250     0.960359                               ((Tb-2.85e+01)/1.5)
3 ~Sl        -1.338750     0.960359                               ((Sl+7.5e-01)/2.5e-01)
4 ~Sd         4.393750     0.960359      4.58       0.0038        ((Sd-1)/1.5e-01)
5 ~Tb*Sl     -4.181250     0.960359     -4.35       0.0048

No. cases = 11         R-sq. =  0.8764       RMS Error = 2.716
Resid. df =  6    R-sq-adj. =  0.7941           Cond. No. = 1
~ indicates factors are transformed.
```

The **Analysis of Variance**, investigating sums of squares of residuals and resulting variances, for the model MOD1_RED is presented in Table 3.4.

Table 3.4 The variance analysis of the model MOD1_RED

```
  Least Squares Summary ANOVA, Response Pf, Model MOD1_RED

Source                          df     Sum Sq.     Mean Sq.    F-Ratio  Signif.
-----------------------------------------------------------------------------------
1 Total(Corr.)                  10    358.3067
2  Regression                    4    314.0368     78.5092     10.64    0.0068
3   Linear                       3    174.1739     58.0580      7.87    0.0168
4   Non-linear                   1    139.8628    139.8628     18.96    0.0048
5  Residual                      6     44.2699      7.3783
6   Lack of fit                  4     43.9810     10.9953     76.13    0.0130
7   Pure error                   2      0.2889      0.1444

                         R-sq. =  0.8764
                     R-sq-adj. =  0.7941
Model obeys hierarchy. The sum of squares for linear terms
is computed assuming nonlinear terms are first removed.
F(4,2) as large as 76.13 is a rare event =>
    likely that significant terms are missing from model.
```

The F-ratio, used to determine whether the non-linear terms, as a group, make a significant contribution to the model fit, is 139.8628/ 7.3783 = 18.96 (see row 4 in Table 3.4). As one can see in the adjacent significance column, such an F-ratio would occur in about 0.48 percent of the cases if the true fit were a linear model. As this seems a very rare event, it appears likely that the second-order model fits significantly better than a linear one.

Because, in this case, there are replicated observations (i.e. two or more experiments made at an identical set of factor values), it is possible to test the so-called lack of fit. The F statistic computed to estimate a lack of fit is F = 10.9953 / 0.1444 = 76.13 (see row 6 in Table 3.4). Such an F-ratio would occur about in 1.3% of the cases if there were no model terms missing, which means the calculated F-value is significant. Because this is a rare event (one expects at least 10%), it is likely that there are additional model terms that should be added to the model.

It may be mentioned that the pure mean square error (row 7, column "Mean Sq.") gives an estimate of σ^2, irrespective of whether the fitted model is correct or not. Only if the model is adequate, the residual mean square MS_{RESID} and the mean square due to lack of fit also give an estimate of σ^2, otherwise they estimate not only σ^2 but also a bias term caused by the inadequacy of the model.

In the presented example the process variance σ^2 estimated on the basis of the eleven experiments is 0.1444.

Because model MOD1_RED has lack of fit, we had to update the experiment-design task described in Section 3.2.2.3 by defining extra experiments to enable the estimation of a full second-order model. Six of these experiments, presented in random order in Table 3.5, are defined according to the central composite faced design (see Appendix A), and the other four, marked with an asterisk, are experiments in the centre point. The last column of this table shows the measured response.

Table 3.5 Additional experiments of central composite faced design

Exp.	Tb	Sl	Sd	Pf
12	28.5	-0.50	1.00	38.14
13	28.5	-0.75	0.85	36.90
14	28.5	-1.00	1.00	40.38
15	28.5	-0.75	1.15	46.46
16*	28.5	-0.75	1.00	40.07
17	30.0	-0.75	1.00	37.19
18*	28.5	-0.75	1.00	40.12
19	27.0	-0.75	1.00	39.06
20*	28.5	-0.75	1.00	41.03
21*	28.5	-0.75	1.00	38.49

The new process model is estimated on the basis of all 21 experiments and after removing non significant terms, the following model, called - MOD_RED, results:

$$\begin{aligned} Pf = \quad & 40.574 - 0.844\cdot(\sim Tb) - 1.295\cdot(\sim Sl) + 4.471\cdot(\sim Sd) + \\ & - 4.181\cdot(\sim Tb \cdot \sim Sl) - 2.755\cdot(\sim Tb^2) - 1.620\cdot(\sim Sl^2). \end{aligned} \tag{3.22}$$

See also Tables 3.6 ÷ 3.7.

Table 3.6 Estimated second-order model MOD_RED

```
Least Squares Coefficients, Response Pf, Model MOD_RED

1 Term          2 Coeff.    3 Std. Error  4 T-value  5 Signif.  6 Transformed Term
------------------------------------------------------------------------------------
1 1             40.573708      0.279304
2 ~Tb           -0.844000      0.277748                         ((Tb-2.85e+01)/1.5)
3 ~Sl           -1.295000      0.277748                         ((Sl+7.5e-01)/2.5e-01)
4 ~Sd            4.471000      0.277748      16.10     0.0001   ((Sd-1)/1.5e-01)
5 ~Tb*Sl        -4.181250      0.310532     -13.16     0.0001
6 ~Tb**2        -2.755393      0.488228      -5.64     0.0001
7 ~Sl**2        -1.620393      0.488228      -3.32     0.0051

No. cases = 21        R-sq. =   0.9764     RMS Error = 0.8783
Resid. df = 14    R-sq-adj. =   0.9662      Cond. No. = 3.503
~ indicates factors are transformed.
```

Table 3.7 Analysis of variance for the model MOD_RED

```
Least Squares Summary ANOVA, Response Pf, Model MOD_RED

Source             df    Sum Sq.     Mean Sq.    F-Ratio  Signif.
-----------------------------------------------------------------------------
1 Total(Corr.)     20    456.8922
2  Regression       6    446.0920     74.3487     96.38    0.0000
3   Linear          3    223.7920     74.5973     96.70    0.0000
4   Non-linear      3    222.3000     74.1000     96.05    0.0000
5  Residual        14     10.8002      0.7714
6   Lack of fit     8      6.6232      0.8279      1.19    0.4282
7   Pure error      6      4.1770      0.6962

                             R-sq. =   0.9764
                         R-sq-adj. =   0.9662
Model obeys hierarchy. The sum of squares for linear terms
is computed assuming nonlinear terms are first removed.
F(8,6) as large as 1.189 is not a rare event =>
    no evidence of lack of fit.
```

In this case, as one can see in Table 3.7, there is no lack of fit (Significance of lack of fit = 0.4282 $\gg$ 0.1). This model will be further investigated as a potential candidate for acceptance.

Another useful method to check a model is **residual analysis**. Figure 3.3 presents the histogram of model MOD_RED, which shows one negative outlier with the residual between -2.5 and -2.0, which is about three times the estimated standard deviation $\sigma = (0.6962)^{½} = 0.834$, calculated as the root of the pure mean square error, presented in Table 3.7. The histogram is not altogether symmetric. However, the occurence of more positive (thirteen) than negative residuals (eight) can simply be explained by the presence of one outlier and the assumption that the estimated mean of the error ϵ must be zero. On the basis of the presented histogram there seems to be no need to transform the model.

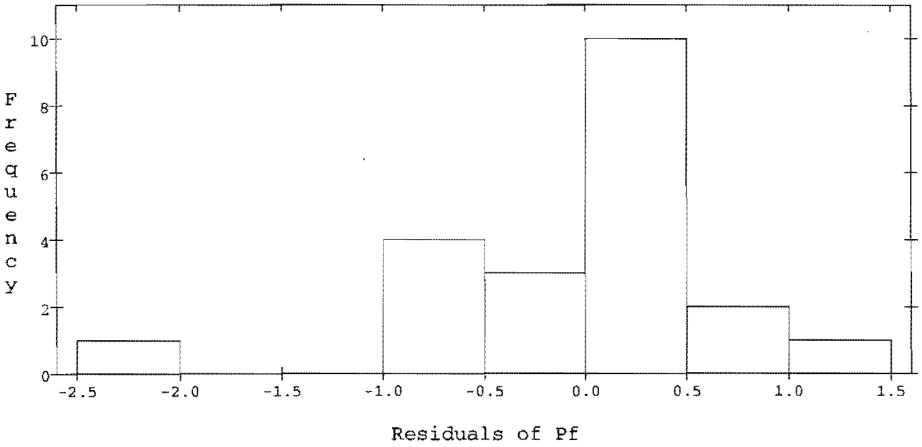


Figure 3.3 Histogram of residuals according to the model MOD_RED

Figure 3.4 presents the residuals in the time sequence. The negative outlier is marked with a circle. There seems to be no reason to suspect process "drift".

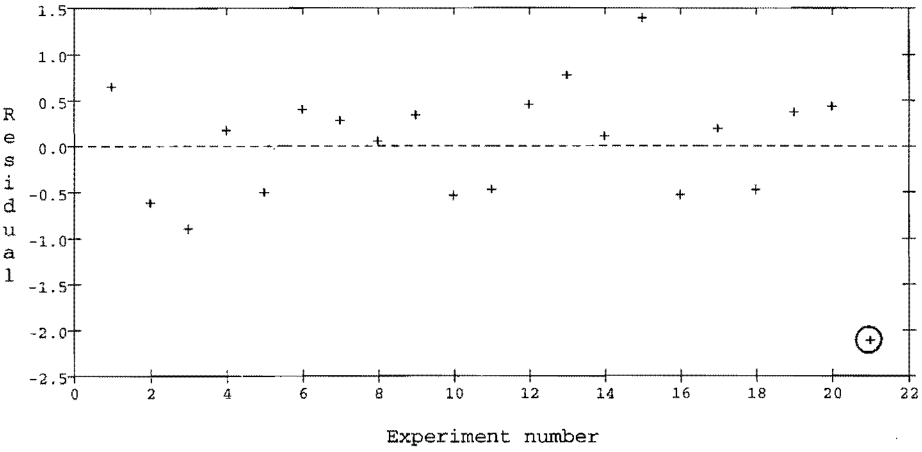


Figure 3.4 Residuals according to the model MOD_RED versus experiment number

Figure 3.5 shows the residuals versus fitted output value in model MOD_RED. No unusual structure is apparent.

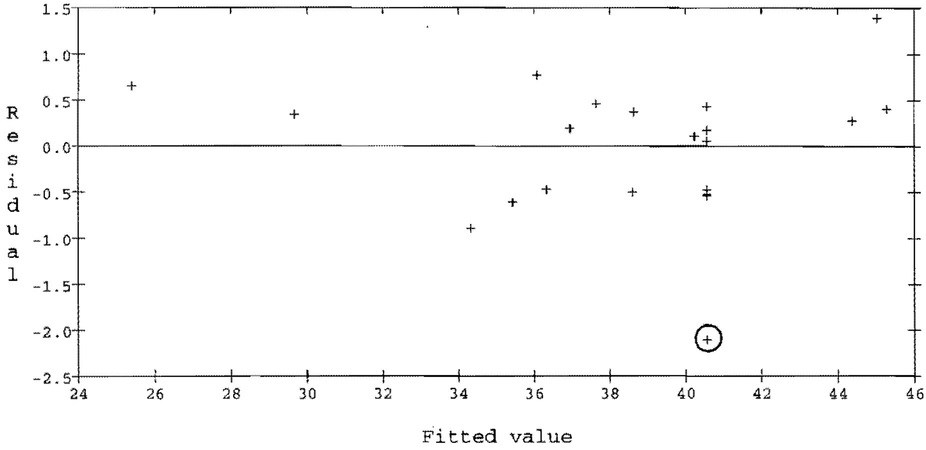


Figure 3.5 Residuals versus fitted value according to the model MOD_RED

Figure 3.6 shows the probability plot of the considered model. Apart from an outlier, marked with a circle, this plot raises no serious doubts concerning the distribution.

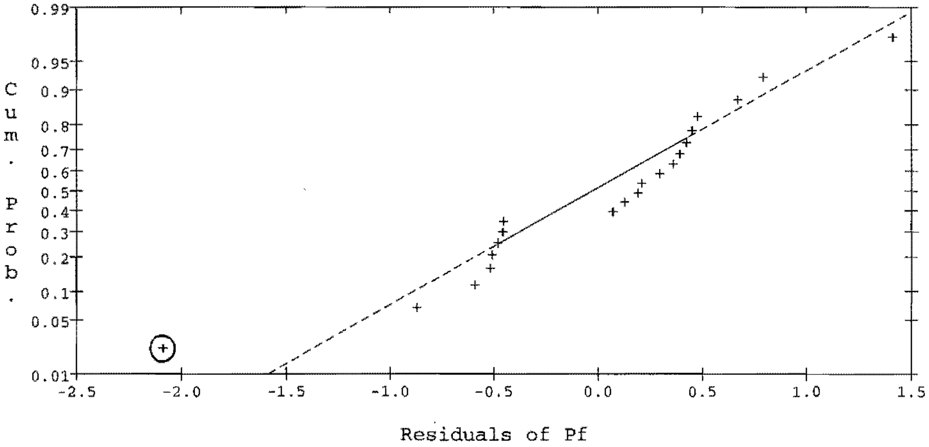


Figure 3.6 Normal probability plot of residuals according to the model MOD_RED

All four residual plots show that the negative outlier is present. In such a situation, a robust regression method, sometimes also called bisquare regression, is preferable [DRA81]. The bisquare method supported by the RS-package weighs the residuals unevenly, using the Tukey's weighting function: the method recalculates the coefficients with the least square method, using the results of the previous fit as a starting point, and it continues in this way until they converge to stable values.

The result of the bisquare estimation applied to the fermentation example is presented in Table 3.8 as the model MOD_RED_ROB. It is as follows:

$$\begin{aligned} Pf = \; & 40.621 - 0.841\cdot(\sim Tb) - 1.288\cdot(\sim Sl) + 4.462\cdot(\sim Sd) + \\ & - 4.176\cdot(\sim Tb \cdot \sim Sl) - 2.775\cdot(\sim Tb^2) - 1.642\cdot(\sim Sl^2) \end{aligned} \qquad (3.23)$$

The model MOD_RED_ROB, presented by (3.23) has the same terms as the previously estimated second-order model, and only slightly changed parameters, which results in slight improvement of the coefficients R^2 and R^2_{adj}.

Table 3.8 Refined process model MOD_RED_ROB estimated with bisquare regression method

```
 Bisquare Coefficients, Response Pf, Model MOD_RED_ROB

1 Term       2 Coeff.   3 Std. Error   4 T-value  5 Signif.  6 Transformed Term
--------------------------------------------------------------------------------
1 1          40.620705      0.277765
2 ~Tb        -0.840804      0.276218                               ((Tb-2.85e+01)/1.5)
3 ~Sl        -1.287789      0.276218                              ((Sl+7.5e-01)/2.5e-01)
4 ~Sd         4.461946      0.276218      16.15      0.0001         ((Sd-1)/1.5e-01)
5 ~Tb*Sl     -4.175997      0.308821     -13.52      0.0001
6 ~Tb**2     -2.775243      0.485538      -5.72      0.0001
7 ~Sl**2     -1.642294      0.485538      -3.38      0.0045

No. cases = 21        R-sq. =  0.9766      RMS Error = 0.8735
Resid. df = 14    R-sq-adj. =  0.9666       Cond. No. = 3.503
~ indicates factors are transformed.
```

To decide whether the bisquare regression is to be recommended, the error variance of least-square regression, denoted by $MS_{RESID\_least\_square}$, might be compared with the error variance associated with robust regression, denoted by MS_{RESID_robust}. If the first one is lower than the second, then it can be concluded that the bisquare method is not appropriate.

In this example, the efficiency of bisquare regression compared to least squares regression is as follows:

$$\text{efficiency} = MS_{RESID\_least\_square} / MS_{RESID\_robust} = (0.8783 / 0.8735)^2, \quad (3.24)$$

which is about 1.01; the error variance of least square regression is a little bit higher than the error variance associated with robust regression. This indicates only a very slight improvement by applying robust regression in this case.

The residual analysis does not always tell the whole story. To check the regression assumption about the normal distribution of the random error ε, χ^2-test may be used to test the so-called **goodness-of-fit**, i.e to check the hypothesis whether the population is normally distributed, especially in dubious cases [HAY94]. However, this procedure is not supported by the RS-software and must be, if necessary, prepared by the user. See Appendix C.

The **PRESS-test** gives an indication about the prediction ability of the model. See also Appendix C. Also this test is not available in the RS-software for models estimated with bisquare regression and it must be, if necessary, prepared by the user. Its use was not attempted in the fermentation example.

After all necessary statistical tests are performed, the best process model can be tentatively accepted. In the case of the fermentation process it is the model MOD_RED_ROB (3.23).

3.2.2.7 Investigating the Tentatively Accepted Process Model

By visualizing the response surface, the factors' main and interaction effects and their confidence intervals (see below), and by predicting process outputs for various factor settings one can learn more about the model and at the same time often also about the process itself.

Visualizing the response surface

Figure 3.7 gives an example of the contour plot of the selected response surface as a function of the initial temperature Tb and the temperature slope Sl. The boundaries of the presented contour plot correspond with the experimentation boundaries given by (3.4) and (3.5). The sugar dosage is fixed at its optimal level: Sd=1.15.

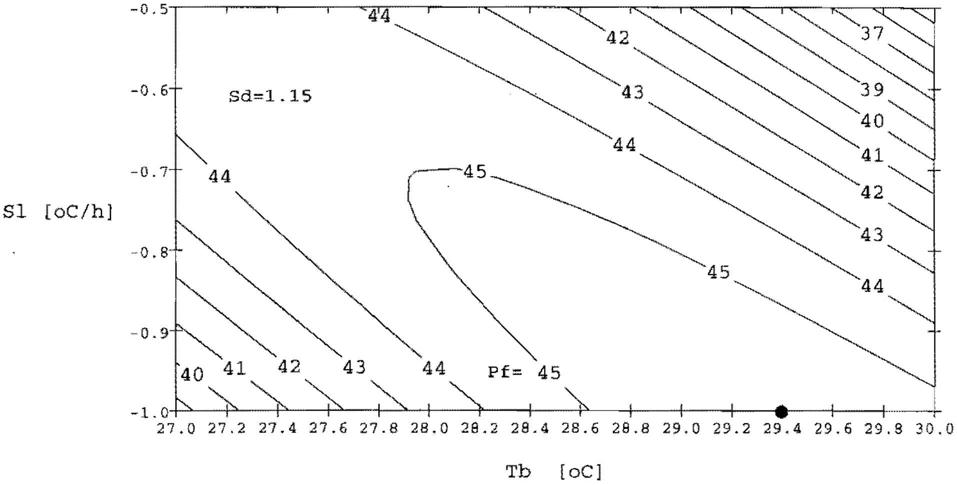


Figure 3.7 Contour plot of the estimated response Pf for the fixed optimal value of Sd=1.15. The maximum is marked by a dot.

The response surface has the optimum on the boundary of the rising ridge.

The visualisation of the response surface in contour plots, like in Figure 3.7, also gives information about the effects of the factors.

Investigating the effects

The estimated main effect of a factor is defined as the maximal change in the fitted response in the experimentation area due to this factor, if all other factors are fixed at their mean level [MON91]. For linear models, the main effect is calculated by changing the factor from its lowest value to its highest. For higher-order models the main effect is equal to the largest response change due to this factor. To calculate it, two points are isolated between the highest and lowest factor values which produce the largest effect.

Because the effects do not include information about their precision, their confidence intervals are also mentioned. The lower and upper confidence bounds define the range of values in which the values will lay with a specified level of confidence.

Table 3.9 shows the main effects and interaction effects calculated for the accepted model MOD_REF_ROB. The main effect of the initial temperature Tb at the mean level of the temperature slope, i.e Sl=-0.75, is equal to -3.68 over the interval [28.27; 30.00]. Although the main effect of Tb as well Sl is negative, the situation changes if one of these factors is fixed at its minimal level.

Table 3.9 Main effects (in bold) and interaction effects with 95% confidence bounds for the model MOD_RED_ROB

Factor	Settings	Settings of other factors	Lower bound	Estimated effect	Upper bound
Tb	28.27 to 30	Sl=-1	-0.70	1.14	2.98
		Sl=-0.75	**-5.24**	**-3.68**	**-2.12**
		Sl=-0.5	-10.34	-8.50	-6.66
Sl	-0.85 to -0.5	Tb=27	0.72	2.66	4.60
		Tb=28.5	**-4.72**	**-3.18**	**-1.65**
		Tb=30	-10.97	-9.03	-7.09
Sd	0.85 to 1.15	Tb=28.5 & Sl=-0.75	**7.74**	**8.92**	**10.11**

If there is an interaction term in the model, an interaction graph or table may be produced to show the main effect of one factor at several fixed levels of the other factors. This may be helpful to determine whether the effect of one factor changes with the value of others, and by how much, yielding conclusions that are valid over the experimentation area.

3.2.2.8 Performing Experiments to Validate and Definitively Accept the Model

The statistical tests described in Section 3.2.2.6, may give an indication as to whether the model is valid or not, but it is recommendable to carry out extra validation experiments using factor values other than those used in the original experiments. The design of validation experiments depends on the type of model, its accuracy and the experiments already performed, e.g. the validation experiments can be planned around the expected optimum or, for linear models, they can be planned around the centre point to explore whether the assumption of linearity is acceptable. If the validation experiments do not disqualify the tentative model, they may, if desired, be used in further improvement and definitive acceptance of the model. When talking about "definitive" acceptance, we actually mean "definitive for the time-being": new process data may indicate the need for model updating, for which purpose the procedure described in Section 3.2.2.5 has to be resumed. However, if the validation experiments do not confirm the tentative model, one has to go back to the experiment-design task as defined in Section 3.2.2.3, by selecting a new type of model, defining new (additional) experiments and/or new constraints.

To validate the tentatively accepted model MOD_RED_ROB, one experiment was done for the factor values: Tb = 29.4 , Sl = -1.0 and Sd = 1.15. These are process conditions in the expected boundary optimum, where no experiments were performed before. Table 3.10 presents the measured response and, for comparison, the predicted response, too. Because the predicted value of the response does not tell the entire story, as a rule statisticians provide an interval, e.g. a 95% confidence interval determined by its lower and upper bound, within which the value of the response in question would be expected to lie.

Table 3.10 Validation experiment for the optimal settings

Factor settings	Lower bound	**Predicted response**	Upper bound	**Measured response**
Tb=29.4, Sl=-1.0, Sd=1.15	43.18	**45.73**	48.28	**45.97**

On the basis of this experiment, here is no reason to assume that the model is not adequate, so the model MOD_RED_ROB is definitively accepted. After inverse transformation to the physical scales the following process model is obtained:

$$\begin{aligned} Pf = & -748.031 + 60.878\ Tb + 273.756\ Sl + 29.807\ Sd \\ & - 11.150\ Tb \cdot Sl - 1.225\ Tb^2 - 25.926\ Sl^2 \end{aligned} \qquad (3.25)$$

This model will subsequently be used for both the generation/improvement of a master recipe or a master control recipe, and the adjustment of a control recipe.

3.2.2.9 *Using Batch Process Data from Other Batch Runs for Model Validation and/or Model Updating*

In principle, any data relating to the same type of batch process can be used for model validation and, if desired, for model improvement. If the process is slowly changing, it is recommendable to give lower weights to the old data and repeatedly estimate the process model. Such a procedure belongs to process monitoring.

The validated process model cum annexis can be included in the recipe adaptation set. The following sections show how it may be used.

3.2.3 Conclusions

The previous section described a systematic strategy for black-box process modelling based on statistical analyses together with the transition-model development scheme that comprises a logical sequence of steps and learning loops for improving the choice of the model structure and the experiment-design task-definition on the basis of the intermediate results.

It should be emphasized that, as shown, statistical analyses alone are not enough to definitively accept a model. The investigation of the tentatively accepted model and the results of validation test-runs is also a very important step.

3.3 Model-Based Recipe Generation and Improvement

The idea of model-based recipe generation and improvement, is to use a process model to compute the most desirable process conditions by optimization of an appropriate performance criterion.
Let us consider situations where only one single performance criterion is involved. How multi-objective problems can be tackled is presented in Section 3.6.

3.3.1 Performance Criterion

The performance criterion provides the measure for the optimization. A great variety of performance criteria is possible; several will be considered in Chapter 6. For the exposition of the theory of this chapter the following performance criterium is considered as being typical of the FRIS-approach:

$$J = \frac{\text{const} + \mathbf{\$}_Y^T\,\mathbf{y} + \mathbf{\$}_X^T\,\mathbf{x} + \mathbf{\$}_U^T \int_{t_0}^{t_F} \mathbf{u}(t)\,dt}{d} \rightarrow \max \qquad (3.26)$$

where:

$$d = \begin{cases} 1 & \qquad (3.26a) \\ t_{BC} & \qquad (3.26b) \end{cases}$$

t_{BC} : the batch cycle time; for the operation at hand this equals:
$t_{BC} = (t_F - t_0) + t_{REST} + t_{IDLE}$ (3.26c)
t_{REST} : time of "secondary" operations such as dosage, heating, cleaning etc.
t_{IDLE} : time in waiting for the completion of the batch cycle time t_{BC}
J : performance criterion (performance index)
const : the negative of the costs of other parts of the production chain
$\mathbf{y}$: process output vector
$\mathbf{x}$: time-independent recipe item vector
$\mathbf{u}(t)$: time-dependent recipe item vector
$\mathbf{\$}_Y$: value vector corresponding to $\mathbf{y}$
$\mathbf{\$}_X$, $\mathbf{\$}_U$: value vector corresponding to $\mathbf{x}$ and to $\mathbf{u}(t)$, respectively; as these vectors correspond to costs, they usually have only negative components.

The denominator d of the performance criterion is either equal to 1, as defined by (3.26a), or to the total batch cycle time, as defined by (3.26b), i.e. the sum of the transformation time $(t_F - t_0)$ of the process phase under consideration and the remaining time t_{REST} needed for the complete batch cycle, which is the time of ingredients dosage, heating, other reaction phases, cleaning etc.

In the simplest case, denominator d = 1 (3.26a). This covers the cases in which one is interested in:

1) maximization of the total added value per batch;
2) maximization of the production yield y (then $\mathbf{y}$ is one-element vector and $\$_Y = 1$;
   the values $\mathbf{\$}_X$ and $\mathbf{\$}_U$ are zero);
3) minimization of the production time according to the specified constraints, e.g. on quality, (then there is only one output y, namely the processing time, $\$_Y = -1$, and $\mathbf{\$}_X$, $\mathbf{\$}_U$ are zero).

If the denominator is equal to the total batch cycle time, then the performance criterion defines the average added value of the process per unit of time. This may serve as an indication of the expected profit per unit of time and may be recommended for economic process optimization. In certain cases, t_{BC} is actually not a constant, but dependent on the transformation time (t_F - t_0). As long as optimization of the transformation time (t_F - t_0) satisfies:

$$t_{IDLE} \geq 0 \tag{3.27}$$

then t_{BC} is not affected, since it is determined by other operations in the production chain. However, if optimization would lead to a negative t_{IDLE}, then t_{BC} has to be increased in order to satisfy (3.27), which may greatly complicate the optimization, as we shall see in Figure 3.9. Often this will lead to a decrease of the performance criterion, in which case the optimum transformation time will correspond to zero idle time.

It should be mentioned that the values $\$_Y$, $\$_X$ and $\$_U$ corresponding to process outputs and inputs often indicate internal value assignments, or the actual prices, taking into consideration, if needed, costs of transport, storage, marketing etc. Thus, the vector values do not necessarily correspond to prices on the market.

3.3.2 Constraints

Of course, the economic optimization has to be subordinated to the quality assurance, usually expressed in terms of constraints on the end specification of the product. Moreover, when there are known relations originating from e.g. mass or energy balances, they can be seen as a supplement to the model equations and used here equality constraints. Besides, depending on the situation, some extra constraints on process output, recipe items or in the form like (3.33) or (3.34), may be added, so to constrain, for example, the energy consumption.

In general, the following constraints may be considered:

$$\mathbf{x}_{MIN} \leq \mathbf{x} \leq \mathbf{x}_{MAX} \tag{3.28}$$
$$\mathbf{ax}_{MIN} \leq \mathbf{Ax} \leq \mathbf{ax}_{MAX} \tag{3.29}$$
$$\mathbf{cx}_{MIN} \leq \mathbf{c(x)} \leq \mathbf{cx}_{MAX} \tag{3.30}$$
$$\mathbf{u}_{MIN} \leq \mathbf{u}(t) \leq \mathbf{u}_{MAX} \tag{3.31}$$
$$\mathbf{y}_{MIN} \leq \mathbf{y} \leq \mathbf{y}_{MAX} \tag{3.32}$$

$$\mathbf{c1}_{MIN} \leq \$_Y^T\, \mathbf{y} + \$_X^T\, \mathbf{x} + \$_U^T \int_{t_0}^{t_F} \mathbf{u}(t)\, dt \leq \mathbf{c1}_{MAX} \tag{3.33}$$

$$\mathbf{c2}_{MIN} \leq \{\, \$_Y^T\, \mathbf{y} + \$_X^T\, \mathbf{x} + \$_U^T \int_{t_0}^{t_F} \mathbf{u}(t)\, dt \,\}/\, t_{BC} \leq \mathbf{c2}_{MAX} \tag{3.34}$$

where:

$\mathbf{x}_{MIN}$, $\mathbf{x}_{MAX}$	:	lower, upper bound on **x**, respectively
A	:	linear constraint matrix on **x**
$\mathbf{ax}_{MIN}$, $\mathbf{ax}_{MAX}$	:	lower, upper linear constraints on **x**, respectively
c(**x**)	:	vector function of non-linear constraints on **x**
$\mathbf{cx}_{MIN}$, $\mathbf{cx}_{MAX}$	:	lower, upper bound on **c**(**x**), respectively
$\mathbf{u}_{MIN}$, $\mathbf{u}_{MAX}$	:	lower, upper constraint on **u**(t), respectively
$\mathbf{y}_{MIN}$, $\mathbf{y}_{MAX}$	:	lower, upper bound on **y**, respectively
$\$_{Y}^{T}$, $\$_{X}^{T}$, $\$_{U}^{T}$	:	coefficient matrix
$\mathbf{c1}_{MIN}$, $\mathbf{c2}_{MIN}$	:	lower general constraints
$\mathbf{c1}_{MAX}$, $\mathbf{c2}_{MAX}$	:	upper general constraints.

These constraints may be divided into four groups:

1) constraints on time-independent items (3.28 ÷ 3.30), including constraint on the total batch time as defined by (3.27);
2) constraints on time-dependent items (3.31);
3) constraints on process outputs (3.32);
4) general integral constraints (3.33 ÷ 3.34), which may involve all recipe items and process outputs; they can be used, for example, to constrain the energy consumption.

3.3.3 Generation and Improvement of a Master Recipe

The objective is to determine, on the basis of an accepted recipe adaptation set, the operating process conditions, that optimize defined performance criterion. See Figure 3.8.

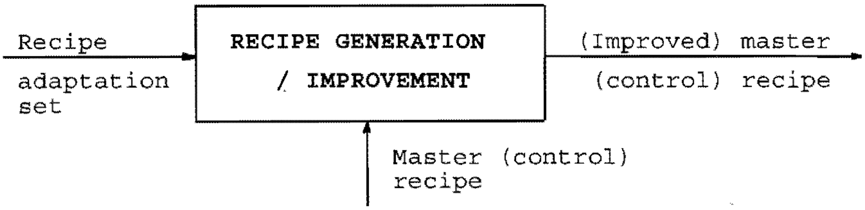


Figure 3.8 Scheme of recipe generation and/or improvement

In the case of a single performance criterion, the recipe-generation/improvement problem turns out to be an optimization problem of the LP (linear programming), QP (quadratic programming), NLP (non-linear problem with linear constraints) or NNP (non-linear optimization problem with non-linear constraints) type, (depending on the type of the process model, constraints and performance criterion) as shown in Figure 3.9.

To determine the optimization type one has to check first of all whether the final time t_F of the process phase under consideration is a free item, and whether time-dependent recipe item vctor **u**(t) affects the performance criterion. Section 3.1.2 presented how a time-dependent recipe item u(t) can be approximated by a number of time-independent recipe items to be used in a process model. The same set of time-independent items, which uniquely describes a time-dependent recipe item u(t), is used to transform the chosen performance criterion into a mathematical criterion as a function of time-independent items only.

Suppose a time-dependent item u(t) can be approximated by a number of straight lines; see Figure 3.10 for the approximation by three such lines. Then the following approximation of the integral of this time-dependent item over time can be used:

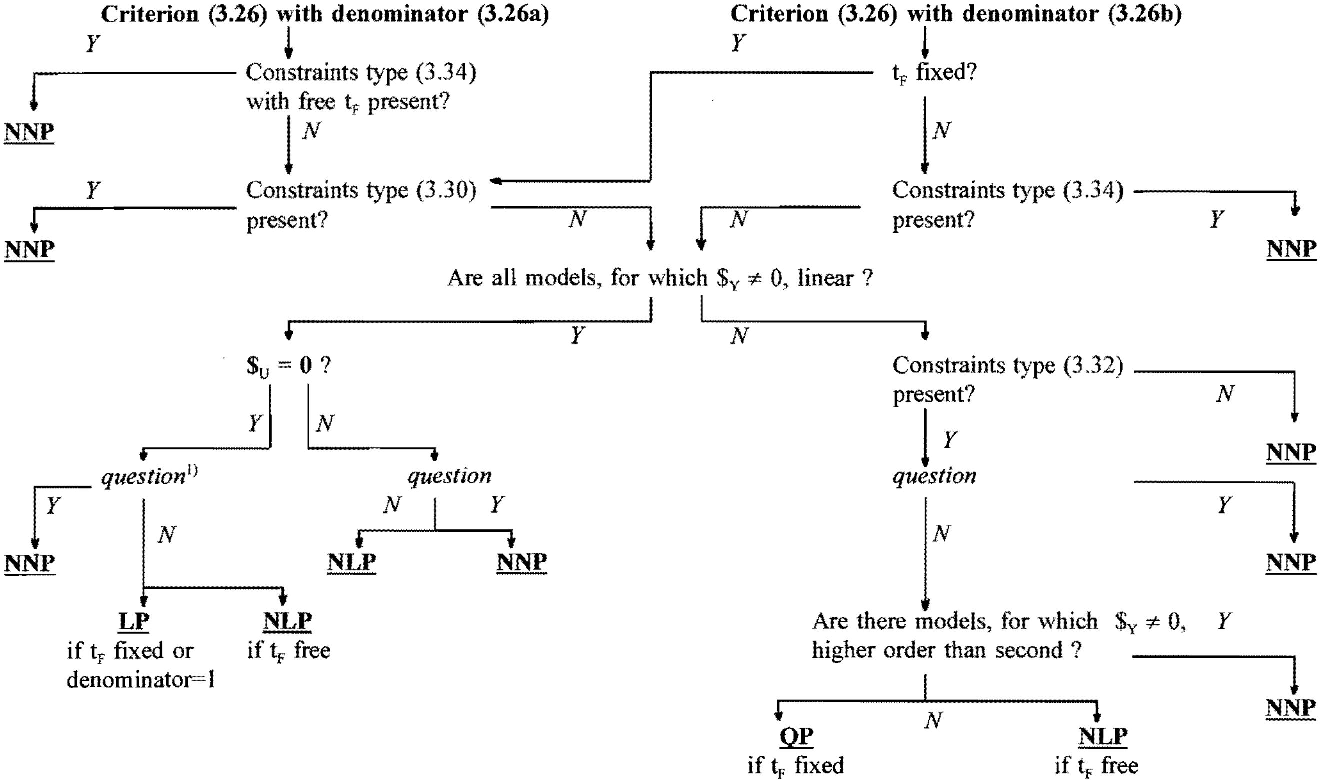
$$\int_{t_0}^{t_F} u(t)\, dt \approx Li\ (t_F - t_0) + 1/2 \sum_{n=1}^{N} ss_n(t_F - t_n)^2 \tag{3.35}$$

where:

$$ss_n = s_n - \sum_{k=1}^{n-1} s_k \tag{3.35a}$$

Li : initial level of the approximating function
s_n : slope of the n^{th} line, n=1,...,N
t_n : time break-point of the n^{th} line, n=1,...,N and $t_N = t_F$
N : number of lines used for approximation of u(t)

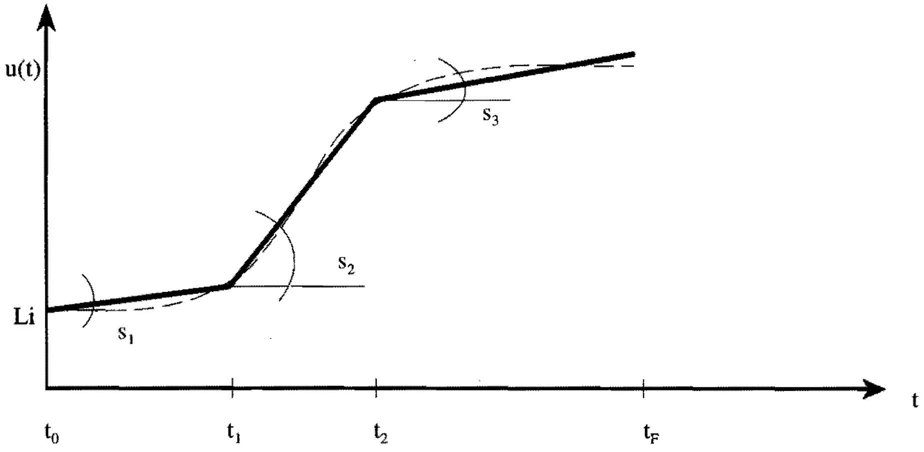
By applying such an approximation, the integral term in the performance criterion is transformed into a polynomial or another non-linear term. The type of the ultimate optimization criterion together with the specified constraints determine the type of optimization problem. Most of the time it will be a non-linear optimization problem with linear (NLP) or non-linear constraints (NNP).


Criterion (3.26) with denominator (3.26a)
Y
Constraints type (3.34) with free t_F present?
NNP
N
Y
Constraints type (3.30) present?
NNP
N
Criterion (3.26) with denominator (3.26b)
Y
t_F fixed?
N
Constraints type (3.34) present?
N
Y
NNP
Are all models, for which $\$_Y \neq 0$, linear ?
Y
N
$\$_U = 0$?
Y
N
question[1)]
Y
NNP
N
LP
if t_F fixed or denominator=1
NLP
if t_F free
question
N
Y
NLP
NNP
Constraints type (3.32) present?
N
NNP
Y
question
Y
NNP
N
Are there models, for which $\$_Y \neq 0$, higher order than second ?
Y
NNP
N
QP
if t_F fixed
NLP
if t_F free

[1)] *question* = constraints type (3.31), (3.33) or, (3.34) with fixed t_F, present?

Figure 3.9 Determining a optimization type corresponding to a recipe-improvement problem

Such problems can be solved with an SQP-method (Sequential Quadratic Programming) designed to optimize a smooth non-linear function subject to constraints, which may include simple bounds on the optimization variables, linear constraints and smooth non-linear constraints, see e.g. [SCA85, LAW94].



$$\int_{t_0}^{t_F} u(t)\,dt \approx \int_{t_0}^{t_1} (Li + s_1(t - t_0))\,dt + \int_{t_1}^{t_2} (Li + s_1(t_1 - t_0) + s_2(t - t_1))\,dt$$
$$+ \int_{t_2}^{t_F} (Li + s_1(t_1 - t_0) + s_2(t_2 - t_1) + s_3(t - t_2))\,dt \qquad (3.35b)$$

Figure 3.10 Approximation of the time dependent item u(t) presented by a dotted line by three line-pieces presented as a continues line.

The solution found may be used to generate or improve a master (control) recipe. When the recipe suggested by the optimal solution is not satisfactory, the formulation of the problem has to be re-examined with a view to reformulating the performance criterion or the constraints. If that does not help, then the used process model must be re-examined, and the whole modelling procedure may have to be repeated.

3.4 Recipe Generation and Improvement on the Basis of First-Principle Models

Up till now, this thesis has dealt with recipe generation/improvement on the basis of black-box models. In some cases, batch processes can be modelled on the basis of first principles: the fundamental physical laws, such as the laws of conservation of mass and energy, and the relationships of chemical reactions such as those involving the Arrhenius temperature dependence.
When a batch process is modelled in this manner, it can often be described by ordinary differential equations. The problems involved in this type of modelling were already described in Section 3.2.1.

In this section, we want to concentrate on recipe generation and improvement on the basis of first-principle models and especially on the solution of the corresponding optimization problem.

A fairly general form of a white model of a batch process is:

$$d\mathbf{x}(t)/dt = \mathbf{g}(\mathbf{x}(t),\ \mathbf{u}(t),\ t) \tag{3.36a}$$

$$\mathbf{y}(t_F) = \mathbf{h}(\mathbf{x}(t_F)) \tag{3.36b}$$

where:

t	:	time, $t \in [t_0, t_F]$	
$\mathbf{x}(t)$	:	vector of state variables;	$\mathbf{x} : [t_0, t_F] \rightarrow \mathbb{R}^n$
$\mathbf{u}(t)$	:	vector of control variables;	$\mathbf{u} : [t_0, t_F] \rightarrow \mathbb{R}^m$
$\mathbf{g}$	:	vector function	$\mathbf{g}: \mathbb{R}^n * \mathbb{R}^m * [t_0, t_F] \rightarrow \mathbb{R}^n$
$\mathbf{y}(t_F)$	:	measured process output vector	$\mathbf{y} \in \mathbb{R}^p$
$\mathbf{h}$	:	vector function	$\mathbf{h}: \mathbb{R}^n \rightarrow \mathbb{R}^p$
$\mathbb{R}$	:	space of real numbers.	

Comparing this notation with that used before in the context of transition models (see Section 3.3) one can see that the time-independent recipe items $\mathbf{x}$ there, other than the final time t_F, are a sub-set of the initial states $\mathbf{x}(t_0)$ here. The time-dependent recipe items $\mathbf{u}(t)$ there are a sub-set of the control variables $\mathbf{u}(t)$ here. Finally, the process outputs $\mathbf{y}$ of transition models can be seen as a function of the final states $\mathbf{x}(t_F)$ here.

The optimization problem corresponding to recipe generation or improvement can be formulated here as the following optimal control problem:

$$\underset{\mathbf{u}(t)}{\text{maximize}} \quad \int_{t_0}^{t_F} f(\mathbf{x}(t),\ \mathbf{u}(t),\ t)\ dt - \Psi(\mathbf{x}(t_0)) + \Phi(\mathbf{x}(t_F)) \tag{3.37}$$

subject to:

$$d\mathbf{x}(t)/dt = \mathbf{g}(\mathbf{x}(t),\ \mathbf{u}(t),\ t) \tag{3.36a}$$

$$\mathbf{y}(t_F) = \mathbf{h}(\mathbf{x}(t_F)) \tag{3.36b}$$

$\mathbf{x}(t_0)$, $\mathbf{x}(t_F)$, t_F may be free, to be chosen optimally

$$\mathbf{x}_{MIN} \le \mathbf{x}(t_0) \le \mathbf{x}_{MAX}; \qquad t_{MIN} \le t_F \le t_{MAX} \tag{3.28'}$$

$$\mathbf{ax}_{MIN} \le \mathbf{Ax}(t_0) \le \mathbf{ax}_{MAX} \tag{3.29'}$$

$$\mathbf{cx}_{MIN} \le \mathbf{c}(\mathbf{x}(t_0)) \le \mathbf{cx}_{MAX} \tag{3.30'}$$

$$\mathbf{u}_{MIN} \le \mathbf{u}(t) \le \mathbf{u}_{MAX} \tag{3.31'}$$

$$\mathbf{y}_{MIN} \le \mathbf{h}(\mathbf{x}(t_F)) \le \mathbf{y}_{MAX} \tag{3.32'}$$

$$\mathbf{c1}_{MIN} \le \{\ \$_Y^T\ \mathbf{h}(\mathbf{x}(t_F)) + \$_X^T\ \mathbf{x}(t_0) + \$_U^T \int_{t_0}^{t_F} \mathbf{u}(t)\ dt\ \} \le \mathbf{c1}_{MAX} \tag{3.33'}$$

$$\mathbf{c2}_{MIN} \le \{\ \$_Y^T\ h(\mathbf{x}(t_F)) + \$_X^T\ x(t_0) + \$_U^T \int_{t_0}^{t_F} \mathbf{u}(t)\ dt\ \}/\ t_{BC} \le \mathbf{c2}_{MAX} \tag{3.34'}$$

where:

$\Psi(\mathbf{x}(t_0))$: initial cost function

$\Phi(\mathbf{x}(t_F))$: final value function, often a function of the process outputs $\mathbf{y}(t_F)$.

It should be mentioned that, like in the case of recipe improvement on the basis of the black-box models, the constraints (3.33') and (3.34') are used when one decides to constraint the total energy consumption.

Table 3.11 compares the terms used in the criterion of the optimal control problem (3.37) with the terms used in the criterion as in formula (3.26) in the context of black-box transition-models. For convenience, it is assumed that t_0 = 0 and the processing time (t_F - t_0) is equal to the final time t_F. Three situations are distinguished:

(i) the performance criterion is to minimize the processing time $t_{F;}$

(ii) the performance criterion is other than t_F, which is fixed;

(iii) the performance criterion is other than t_F, which is an optimization variable.

Note that, when the processing time has to be minimized (situation *(i)*), in conventional dynamic optimization, the following criterion will be maximized:

$$J = - \int_{t_0}^{t_F} dt \tag{3.38}$$

whereas in the context of black-box transition models, the processing time is treated as a measured response y and the following criterion is to be maximized:

$$J = \$_Y \, y \tag{3.39}$$

where $\$_Y = -1$ and $y = t_F$.

Table 3.11 Comparison of optimization criteria used in the context of first-principle models and black-box transition-models

terms used in the context of white models	**Performance criterion: minimize t_F** ***(i)***	**Performance criterion is other than t_F, which is fixed** ***(ii)***	**Performance criterion is other than t_F, which is an optimization variable** ***(iii)***
$f(\mathbf{x}(t), \mathbf{u}(t), t) =$	- 1	as $\$_U^T \, \mathbf{u}(t)$ in formula (3.26)[1]	as $\$_U^T \, \mathbf{u}(t)$ in formula (3.26)[1]
$\Psi(\mathbf{x}(t_0)) =$ (initial cost function)	0	as $\$_X^T \, \mathbf{x}$ in formula (3.26)	as $\$_X^T \, \mathbf{x}$ in formula (3.26), but with $\$_X^T = 0$ if $x_i = t_F$
$\Phi(\mathbf{x}(t_F)) =$ (final value function)	0	as $\$_Y^T \, \mathbf{y}$ in formula (3.26)	as $\$_Y^T \, \mathbf{y}$ in formula (3.26)

[1] $$J = \{ \text{const} + \$_Y^T \, \mathbf{y} + \$_X^T \, \mathbf{x} + \$_U^T \int_{t_0}^{t_F} \mathbf{u}(t) \, dt \} / \, d \quad \rightarrow \quad \max \tag{3.26}$$

$$d = \begin{cases} 1 & \text{for the case } (i) \\ 1 \text{ or } t_{BC} & \text{(constant)} \qquad \text{for the case } (ii) \\ t_{BC} & \text{(not constant)} \qquad \text{for the case } (iii) \end{cases}$$

If the performance criterion is other than to minimize processing time, but t_F is an optimization variable (situation *(iii)*), then in the context of transition models it will be an element of **x**, and in the context of white models it will be handled as a separate variable, namely free end-time.

It is worth emphasizing that the optimal control problem (3.37), (3.28' - 3.34'), stemming from recipe generation or improvement, distinguishes itself from optimal control problems as usually defined, by the following features:

1) the optimization function can contain not only a final value function but also an initial cost function;
2) final time can be a free variable, which in many cases may have to be chosen optimally;
3) not only final states, but also initial states can be free variables, which may be chosen
optimally;
4) the optimization problem almost always involves constraints on states variables, which are notoriously difficult to handle;
5) constraints may be imposed (lower/upper bounds, linear and non-linear constraints) on initial and final states variables;
6) integral constraints may be imposed on control variables; these constraints may also involve initial and final states, and possibly final time.

To solve that rather different kind of problems, first of all each integral constraint such as (3.33') or (3.34'), if present, should be replaced by an additional differential equation for newly defined state variables **w**(t) (see formula 3.40) with boundary conditions (3.41) and constraints (3.42, 3.43). Thus, (3.33') can be replaced by:

$$d\mathbf{w}(t)/dt = \mathcal{S}_U^T\, \mathbf{u}(t), \tag{3.40}$$

$$\mathbf{w}(t_0) = 0 \tag{3.41}$$

$$\mathbf{c1}_{MIN} - \mathcal{S}_Y^T\, \mathbf{h}(\mathbf{x}(t_F)) - \mathcal{S}_X^T\, \mathbf{x}(t_0) \le \mathbf{w}(t_F) \tag{3.42}$$

$$\mathbf{w}(t_F) \le \mathbf{c1}_{MAX} - \mathcal{S}_Y^T\, \mathbf{h}(\mathbf{x}(t_F)) - \mathcal{S}_X^T\, \mathbf{x}(t_0) \tag{3.43}$$

The same method should be used if constraints in the form of (3.34') are present. The total number of states will be n+nc1+nc2, where nc1, nc2 are numbers of integral constraints type (3.33') and (3.34'), respectively.

The necessary optimality conditions for optimal control problems are given by the Maximum Principle of Pontryagin, see e.g. [HES66], [LEO92]. When the final time and the initial/final states are not fixed, some additional conditions are needed to determine their optimal value. These conditions are called transversality conditions.

Table 3.12 gives a general view of the necessary optimality and transversality conditions for possible dynamic optimization problems corresponding to various cases of recipe generation/improvement (for convenience, the total number of states is represented there by n). It is also assumed that no constraints on state variables are involved and that the so-called rank condition is satisfied, that is that the number of active constraints (constraints which at a moment hold as equalities) is not greater than the number of control variables.

In the context of the FRIS-project the first-principle models, one of an irreversible first-order reaction $A \rightarrow B$, and other of a fermentation process, are applied to generation of a master recipe, see [STE92], [THI94]. Other examples of optimal control problems for (fed-) batch processes can be found in e.g. [RIJ91], [VAN93], [OVE92].

A great drawback of the approach based on first-principle models is that state constraints are very difficult to handle. Moreover, the optimality conditions result in a so-called two-point boundary value problem, TPBVP, which is often hard to solve. Another important drawback is that the optimal control solution is very model-specific. If a white process model is made after a number of simplifying assumptions, the resulting optimal profiles/initial conditions are actually sub-optimal. However, the most important problem is that white process models are seldom available in many branches of the process industry. The effort for building a useful first-principle model often exceeds one man-year and therefore it may be not worthwhile to develop it for a short-lived process.

Table 3.12 The necessary optimality conditions including transversality conditions for general constrained optimal control problems

Definition of general constrained control problem	**The necessary optimality conditions (Maximum Principle of Pontryagin)**	**Transversality conditions**
$\max\limits_{u(t)} J = \int_{t_0}^{t_F} f(\mathbf{x}(t),\mathbf{u}(t),t)dt + \Theta(\tau_0,\tau_F)$ subject to: $dx(t)/dt = \mathbf{g}(\mathbf{x}(t),\mathbf{u}(t),t)$ $\kappa_j(\mathbf{u}(t)) = 0 \quad j=1,\ldots,r'$ $\kappa_k(\mathbf{u}(t)) \geq 0 \quad k=r'+1,\ldots,r$ $\nu_j(\tau_0,\tau_F) = 0 \quad j=1,\ldots,E$ $\nu_k(\tau_0,\tau_F) \geq 0 \quad k=E+1,\ldots,E+I$ where: $\mathbf{x}(t) = (x_1(t),\ldots,x_n(t))^T$ $\mathbf{u}(t) = (u_1(t),\ldots,u_m(t))^T$ $\tau_0 = (t_0,x_1(t_0),\ldots,x_n(t_0))^T$ $\tau_F = (t_F,x_1(t_F),\ldots,x_n(t_F))^T$ $\Theta(\tau_0,\tau_F)$: initial cost/ final value function	If $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$ are optimal, then $\exists$ ($\lambda_j(t)$, j=1,...,r) such that $\partial\mathcal{L}^*/\partial u_i = 0 \quad i=1,\ldots,m$ $\kappa_j(\mathbf{u}^*(t)) = 0 \quad j=1,\ldots,r'$ $\lambda_k(t) \geq 0,\ \kappa_k(\mathbf{u}^*(t)) \geq 0$ $\lambda_k(t)\ \kappa_k(\mathbf{u}^*(t)) = 0 \quad k=r'+1,\ldots,r$ $d\pi_i(t)/dt = -\ \partial\mathcal{L}^*/\partial x_i(t) \quad i=1,\ldots,n$ $d\mathbf{x}^*(t)/dt = \partial\mathcal{L}^*/\partial\boldsymbol{\pi}(t) = \mathbf{g}(\mathbf{x}(t),\mathbf{u}(t),t)$ where: $\mathcal{L}(\mathbf{x}(t),\mathbf{u}(t),t) = \mathcal{H}(\mathbf{x}(t),\mathbf{u}(t),t) + \sum \lambda_j(t)\kappa_j(\mathbf{u}(t))$ Lagrangean $\mathcal{H}(\mathbf{x}(t),\mathbf{u}(t),t) = f(\mathbf{x}(t),\mathbf{u}(t),t) + \boldsymbol{\pi}^T(t)\ \mathbf{g}(\mathbf{x}(t),\mathbf{u}(t),t)$ Hamiltonian $\lambda_j(t)$, j=1,...,r multipliers $\boldsymbol{\pi}(t)$: adjoint (costate) variable; continuous with piecewise-continuous derivatives	To assure that τ_0^*,τ_F^* are optimal the following transversality conditions must also be satisfied: $\mathcal{H}^*(t_0) - \partial\Theta^*/\partial t_0 - \sum_{j=1}^{E+I}\mu_j\ \partial\nu_j^*/\partial t_0 = 0$ $\pi_i(t_0) + \partial\Theta^*/\partial x_i(t_0) + \sum_{j=1}^{E+I}\mu_j\ \partial\nu_j^*/\partial x_i(t_0) = 0$ $\mathcal{H}^*(t_F) + \partial\Theta^*/\partial t_F + \sum_{j=1}^{E+I}\mu_j\ \partial\nu_j^*/\partial t_F = 0$ $\pi_i(t_F) - \partial\Theta^*/\partial x_i(t_F) - \sum_{j=1}^{E+I}\mu_j\ \partial\nu_j^*/\partial x_i(t_F) = 0$ i=1,...,n where: μ_j: multipliers (not included in Hamiltonian) with the following properties: - μ_j are constants, j = 1,...,E+I - $\nu_j(\tau_0^*,\tau_F^*) = 0 \quad j=1,\ldots,E$ - $\mu_j \geq 0,\ \nu_j(\tau_0^*,\tau_F^*) \geq 0,\ \mu_j\nu_j(\tau_0^*,\tau_F^*) = 0 \quad j=E+1,\ldots,E+I$ the asterisk * by functions and derivatives means that they are evaluated at $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$, τ_0^*,τ_F^*

3.5 Experimental Process Optimization

Experimental process optimization is an approach quite different from the model-based one. It is also used for recipe generation and improvement, but it is based only on data from test runs and, in principle, without using a mathematical process model. In this section, the discussion is limited to problems where only one performance criterion, denoted by J, has to be maximized.

The **experimental** strategy searches for a process optimum by evaluating the responses corresponding to a number of test runs performed for various settings of recipe items. There are a number of possibilities, for example:

1) the (Nelder-Mead) simplex method;
2) the multiplex-fitting method;
3) line optimization.

What these methods have in common is their heuristic character and their goal: to find a process optimum without utilizing a process model. With this, they can be distinguished from **model-based** recipe improvement methods. Moreover, model-based methods are directed towards improvement, and experimental ones towards optimization. Therefore, we prefer to talk about "experimental optimization", rather than about "experimental improvement". The following sub-sections describe the three empirical strategies.

3.5.1 The Nelder-Mead Simplex Method

In 1962 Senedley proposed an empirical optimization technique, which uses simplices instead of a factorial design [SPE62]. Nelder and Mead modified it for function minimization [NEL65].

The idea is to evaluate the process response in a number of experimental settings forming a so-called simplex in the factor space and continually forming new simplices toward the optimum. Basically outlined, the algorithm repeatedly creates a new simplex by "reflecting" the point with the worst response with respect to the hyperplane of the remaining points and next adapts it to the local landscape, with the goal of ultimately enclosing the optimum in the simplex interior. Next, the simplex is shrunk towards the vertex with the best response (see below for the two-dimensional example).

An n-dimensional simplex, where n is the number of factors varied during experimentation, is defined by n+1 non-collinear points called vertices and denoted by $\mathbf{x}_1, ..., \mathbf{x}_{n+1}$ because of their correspondence to recipe item vectors. For two factors a simplex will be a triangle, for three factors a tetrahedron, etc.

To execute an iteration of a Nelder-Mead simplex algorithm, only ordinal information about responses corresponding to the simplex vertices is needed, and what one actually needs to know is only which set of process conditions gives the best response, and which set gives the worst and which the second worst. Thus, one has at each iteration (N) a triplet of vertices ($\mathbf{x}_b^N$, $\mathbf{x}_{w-1}^N$, $\mathbf{x}_w^N$) each corresponding to one set of recipe items values, where $\mathbf{x}_b^N$ is the vertex that is the best approximation to the solution (i.e. $J(\mathbf{x}_b^N)$ is the highest value of the criterion) while $\mathbf{x}_{w-1}^N$ and $\mathbf{x}_w^N$ are the second worst and the worst (i.e. $J(\mathbf{x}_{w-1}^N)$, $J(\mathbf{x}_w^N)$ are the second lowest and lowest value of the criterion), respectively, in the current (N^{th}) simplex. As we shall see one does not need any order information about the remaining vertices to drive this algorithm.

The algorithm proceeds by "moving the simplex" away from the current worst vertex $\mathbf{x}_w^N$. The four possible operations are called reflection, expansion, contraction and shadow contraction. When the optimum is enclosed in the simplex interior, the fifth operation, so-called shrinkage, is applied.

Altogether, these five operations are defined as follows:

1) reflection: $\mathbf{x}_W^N \rightarrow \mathbf{x}^r$

$\mathbf{x}^r = (1+\alpha)\ \bar{\mathbf{x}}^N - \alpha\ \mathbf{x}_w^N;$ $\alpha > 0;$ e.g. $\alpha = 1$ (3.44)

2) expansion: $\mathbf{x}^r \rightarrow \mathbf{x}^e$

$\mathbf{x}^e = \beta\ \mathbf{x}^r + (1 - \beta)\ \bar{\mathbf{x}}^N;$ $\beta > 1;$ e.g. $\beta = 2$ (3.45)

3) contraction: $\mathbf{x}_W^N \rightarrow \mathbf{x}^c$

$\mathbf{x}^c = \gamma\ \mathbf{x}_w^N + (1 - \gamma)\ \bar{\mathbf{x}}^N;$ $0 < \gamma < 1;$ e.g. $\gamma = 1/2$ (3.46)

4) shadow contraction: $\mathbf{x}^r \rightarrow \mathbf{x}^{sc}$

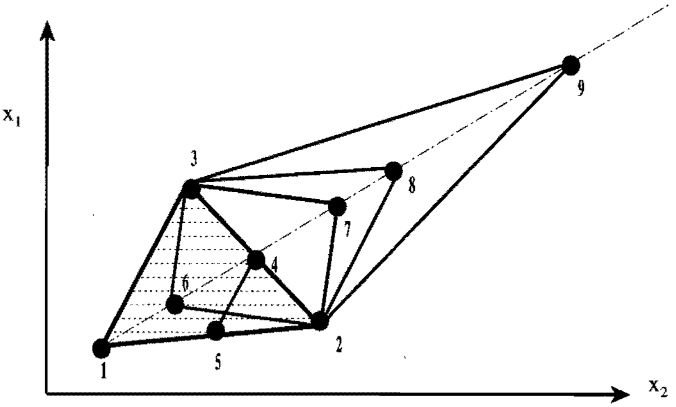
$\mathbf{x}^{sc} = \delta\ \bar{\mathbf{x}}^N + (1 - \delta)\ \mathbf{x}^r;$ $0 < \delta < 1;$ e.g. $\delta = 1/2$ (3.47)

5) shrinkage: $\mathbf{x}_i^N \rightarrow \mathbf{x}_i^{N+1}$ i = 1,...,n+1

$\mathbf{x}_i^{N+1} = \eta\ \mathbf{x}_i^N + (1 - \eta)\ \mathbf{x}_b^N;$ $0 < \eta < 1;$ e.g. $\eta = 1/2$ (3.48) where $\bar{\mathbf{x}}^N$ for N^{th} iteration is the centroid of all points except $\mathbf{x}_w^N$ and is calculated as follows:

$$\bar{\mathbf{x}}^N = (\sum_{i=1}^{n+1} \mathbf{x}_i^N - \mathbf{x}_w^N) / n \tag{3.49}$$

They are presented in Figure 3.11 for the case n=2.



$1 = \mathbf{x}_1^N = \mathbf{x}_w^N$

$2 = \mathbf{x}_2^N = \mathbf{x}_b^N = \mathbf{x}_2^{N+1}$

$3 = \mathbf{x}_3^N = \mathbf{x}_{w-1}^N$

$4 = \bar{\mathbf{x}}^N = \mathbf{x}_3^{N+1}$

$5 = \mathbf{x}_1^{N+1}$

$6 = \mathbf{x}^c$

$7 = \mathbf{x}^{sc}$

$8 = \mathbf{x}^r$

$9 = \mathbf{x}^e$

Figure 3.11 Start simplex for two factors, marked by dotted lines, and after various possible operations:
reflection 1 → 8
expansion 8 → 9
contraction 1 → 6
shadow contraction 1 → 7
shrinkage 1 → 5 & 3 → 4 & 2 → 2.

Figure 3.12 illustrates the Nelder-Mead simplex algorithm applied to the optimization of a two-dimensional function.

A more detailed description of the basic operations moving a simplex in the factor space and our method of dealing with cases in which constraints are present, is detailed in Appendix D.

A novel and promising method of dealing with multi-criteria optimization, also employing the Nelder-Mead empirical optimization method, is described in Section 3.6.3.2.

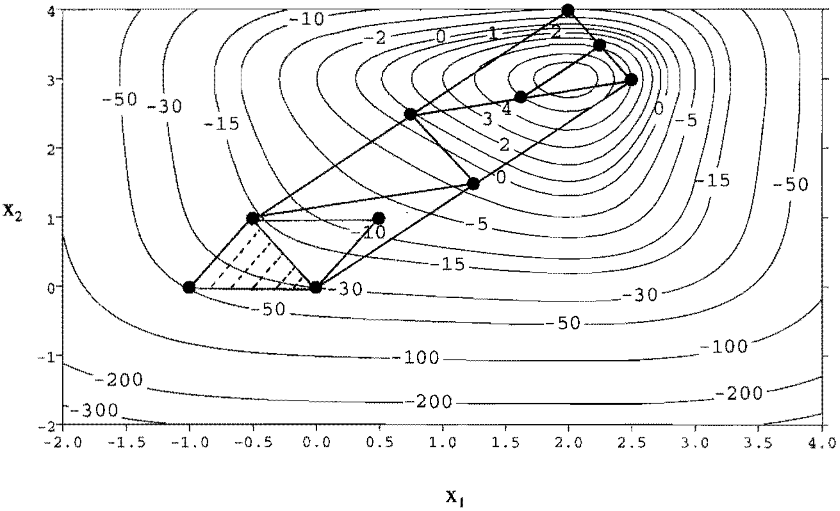


Figure 3.12 Illustration of the Nelder-Mead simplex algorithm applied to the optimization of the function: $f(x_1,x_2) = (x_1 - 1)^4 + (x_2 - 1)^4 - 4 (x_1-1)-4 (x_2-1)$. The start simplex is marked by dotted lines.

3.5.2 The Multiplex-Fitting Method

The Nelder-Mead simplex method makes a rather limited use of the information coming from the test runs, only those belonging to the present simplex are utilized by the procedure; once a point is removed from the simplex, its data will never be used again. Moreover, the reflection procedure does not utilize information about the response surface. Therefore, we have developed an alternative experimental process optimization method, called **multiplex fitting**, which defines new experimental settings not only by comparison of previous test run results, but also on the basis of an estimated model of the response surface. This method was found to be more efficient than the simplex method.

The principle of multiplex fitting is based on a local approximation of the response surface, for which initially the so-called restricted quadratic model is chosen. It can be presented as follows:

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \beta_{n+1} \sum_{i=1}^{n} x_i^2 \qquad (3.50)$$

where:

y : response
β_i : model coefficient to be estimated, i = 0,...,n+1
n : number of recipe items (factors)
x_i : i^{th} recipe item, i = 1,...,n.

In order to estimate n+2 model coefficients and the variance of the error, at least n+3 test runs have to be done. These first n+3 test runs can be planned according to the Nelder-Mead simplex method as discussed before.

The restricted quadratic model (3.50) will be used as long as the number of the test runs is insufficient to estimate a full second-order model:

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i\, x_i + \sum_{i=1}^{n} \beta_{n+i}\, x_i^2 + \sum_{i=1}^{n-1} \sum_{j>i}^{n} \beta_{2n + (i-1)(n-0.5i) + j - i}\, x_i x_j \qquad (3.51)$$

Here 0.5(2+n)(1+n) coefficients must be estimated, therefore at least 1+0.5(2+n)(1+n) test runs results must be available. To estimate the model coefficients, the regression method, described in Section 3.2.2.5, can be used.

Before describing the algorithm, we would like to introduce the concepts of the **domain** and the **design area**, defined for each iteration step *N*. Both terms are visualized in Figure 3.13.

The **domain** D^N of the N^{th} iteration is a rectangular area determined by the last p+1 test runs, where "p" is the number of estimated model coefficients, i.e. p = n+2 for a restricted quadratic model and p = 1+0.5n(n+1) for a full second-order model. The formal definition is as follows:

$$D^N = \{\, \mathbf{x} \in \mathbb{R}^n: \quad \min_{1 \le j \le p+1} x_i^j \le x_i \le \max_{1 \le j \le p+1} x_i^j \,\} \qquad (3.52)$$

where:

x_i^j : the i^{th} element of the j^{th} point , i=1,...n, j=1,...,p+1; only points corresponding to the last p+1 test runs are considered
p : number of model coefficients to be estimated.

As will be presented below, the size of the domain will change for each iteration step. If in this domain test-run points are also present, other than the last p+1, they can also be used for model estimation. However, the user has the possibility to assign lower weight factors to these points, if he suspects that the older information is out of date.

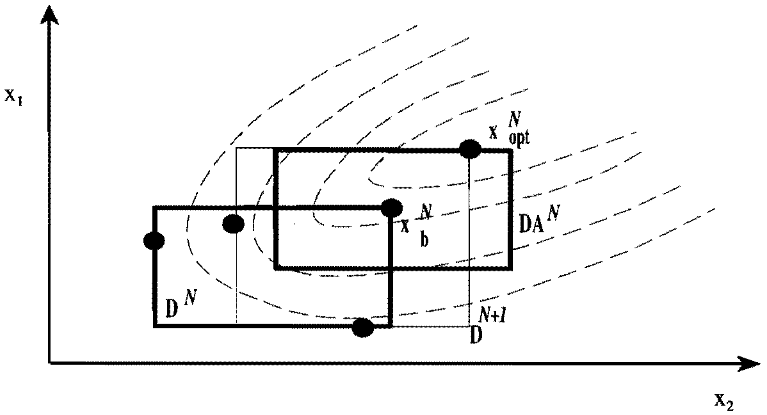


Figure 3.13 Domain D^N and design area DA^N of the N^{th} iteration of the multiplex fitting method applied to two factors x_1 and x_2; D^{N+1} is the domain of the $(N+1)^{th}$ iteration. Points correspond to the performed test runs; $\mathbf{x}_b^N$ is the best point of the domain D^N; $\mathbf{x}^N_{opt}$ is the designed test run in DA^N .

The other area used by the procedure is the so called **design area**. In each iteration its size is the same as the size of the domain. Its centre point is the best point of the domain, denoted here by $\mathbf{x}_b^N$ (mostly, in the beginning, it will be a point at the domain boundary); in other words the design area follows from placing the domain around its best point. The design area DA^N of the N^{th} iteration is defined as follows:

$$DA^N = \{ \mathbf{x} \in \mathbb{R}^n: \; x_i^b - \tfrac{1}{2} (x_{i,\,max} - x_{i,min}) \leq x_i \leq x_i^b + \tfrac{1}{2} (x_{i,\,max} - x_{i,min}) \} \tag{3.53}$$

where:

x_i^b : i^{th} element of the best domain point $\mathbf{x}_b^N = (x_1^b,..., x_n^b)$

$$x_{i,max} = \max_{1 \leq j \leq p+1} x_i^j \tag{3.54}$$

$$x_{i,min} = \min_{1 \leq j \leq p+1} x_i^j \tag{3.55}$$

In the design area the multiplex fitting procedure has to calculate (model-based) the settings of the next test run corresponding to a new point $\mathbf{x}^N_{opt}$.

The multiplex-fitting algorithm for n recipe items works as follows:

```
(0)  set $N = 0$;
     do p+1 = n+3 test runs according to the Nelder-Mead simplex method and choose the best one $\mathbf{x}_b^0$;
(1)  define the domain $D^N$ according to the definition (3.52) on the basis on the last p+1 test runs;
(2)  define the design area $DA^N$ according to the definition (3.53);
(3)  determine whether the domain contains points corresponding to the previous test runs, other than the last p+1. If this is so, include also these points in the estimation procedure, with adjusted weights, if desirable;
(4)  depending on the number of test runs "included" in the domain $D^N$ estimate a restricted quadratic or a full second-order process model;
(5)  optimize the specified performance criterion J(x), see formula (3.26), inside the design area $DA^N$; this optimization can be done with the SQP optimization method [LAW94];
(6)  in the found optimum $\mathbf{x}^N_{opt}$ perform a test run to obtain $J(\mathbf{x}^N_{opt})$;
(7)  $\mathbf{x}_b^{N+1} = \begin{cases} \mathbf{x}^N_{opt} & \text{if } J(\mathbf{x}^N_{opt}) \geq J(\mathbf{x}_b^N) \\ \mathbf{x}_b^N & \text{otherwise} \end{cases}$   (3.56)
(8)  check the stopping criteria (see 3.57 ÷ 3.58),
     decide to stop **or** go to (9);
(9)  replace the "oldest" point from the last p+1 points of the domain $D^N$ by $\mathbf{x}^N_{opt}$;
(10) $N = N + 1$, go to (1)
```

By defining a domain in step *(1)*, one point is different from the points of the previous domain. As a consequence, a new domain can have a size different from the previous one. As may be expected, in the beginning the domain will tend to increase in size, but in due time it will become smaller when the procedure approaches the optimum. The size of the domain and the progress of the improvement gives an indication to the user as to when to stop. In the first instance, the following stopping criterion is checked:

$$x_{i,max} - x_{i,min} \leq crit1_i \qquad i = 1,...,n \tag{3.57}$$

where:

$crit1_i$: some preset value limiting the domain size in the ith direction.

This criterion checks for each recipe item x_i whether its range in DA is smaller than some preset value $crit1_i$. If it is satisfied, i.e. the domain is sufficiently small, then the test runs corresponding to the all points of the domain should be repeated a number of times according to the rule (3.1). Next, the second stopping criterion, see (3.58), is investigated. It checks whether the mean of the criterion values corresponding to the previous best and to the actual best process conditions do differ less than *crit2*, taking into consideration the experimental error.

$$| \bar{J}(\mathbf{x}_b^{N-1}) - \bar{J}(\mathbf{x}_b^{N}) | \le crit2 + 2\zeta \tag{3.58}$$

where:

$\bar{J}(\mathbf{x}_b^{N-1})$, $\bar{J}(\mathbf{x}_b^{N})$: mean of the criterion values for the process conditions $\mathbf{x}_b^{N-1}$ and $\mathbf{x}_b^{N}$, respectively

crit2 : some preset value limiting the size of the criterion improvement

ζ : a number corresponding to the experimental error, defined as in (3.59 ÷ 3.60)

The computation of the experimental error is analogous to the computation of the pure error in the context of experiment design on the basis of replications as described in Appendix C. The pure mean square error for one response of interest, calculated on the basis of replicated measurements of this response for (n+1) various conditions, divided by a number of degrees of freedom, gives an estimate of variance σ^2 of the experimental error (see formula (C.11) for comparison):

$$\sigma^2 = \frac{\sum_{j=1}^{n+1} \sum_{k=1}^{r_j} (y_{jk} - \bar{y}_j)^2}{(\sum_{j=1}^{n+1} r_j) - (n+1)} \tag{3.59}$$

where:

y_{jk} : k^th^ measurement of the response y at the j^th^ point of DA

$\bar{y}_j = 1/r_j \sum_{k=1}^{r_j} y_{jk}$: response mean for the jth point of DA

r_j : number of test run repetitions at the jth point of DA.

Depending on the performance criterion that is optimized, the number ζ in the stopping criterion (3.58) is computed as follows:

$$\zeta = \$^{T}_{Y}\,\sigma / d \tag{3.60}$$

where $\$^{T}_{Y}$ and the denominator d are defined as in the performance criterion (3.26).

This concludes our introduction to the new method for experimental optimization. A comparison of the multiplex-fitting method with the Nelder-Mead simplex method on the basis of two test fun-ctions was done in [BEE94]. It was shown there that multiplex fitting reached the optimum after far fewer test runs than the simplex method, as Table 3.13 illustrates.

Table 3.13 Comparison of the Nelder-Mead simplex method with multiplex fitting

Test function	**Rosenbrock's Parabolic Valley** $f(x_1,x_2)=-100(x_2-x_1^2)^2-(1-x_1)^2$	$f(x_1,\ldots,x_k)=-\sum_{i=1}^{k}[(x_i-i)^4-4(x_i-i)]$ **Three examples considered for k = 2 , 3 , 4**
number of test runs to reach optimum with the Nelder-Mead method	**175**	**92 for k = 2** **112 for k = 3** **487 for k = 5**
number of test runs to reach optimum with the multiplex fitting method	**65**	**37 for k = 2** **68 for k = 3** **165 for k = 5**

3.5.3 Line Optimization

When, during model-based recipe generation or improvement, the optimum point has been reached on the boundary, a rather different situation is encountered than when the optimum lies inside the optimization area. In such a case, it may be profitable to enlarge the optimization area, if possible, by shifting the boundary a little bit, and then compute how many profit can be expected by enlarging of the area. But, if there are possibilities to enlarge the optimization area much more, i.e. there are allowed larger variations in recipe items than during model-based optimization, the experimental optimization according to a line-optimization procedure may be carried out to test along the path of steepest ascent where a better optimum lies beyond the original optimization area.

Appendix F describes the line-optimization procedure developed for the purpose of the FRIS-approach.

3.6 Multi-Objective Optimization

During the recipe generation and/or improvement phase one has very often not a single criterion, but a variety of objectives that, more often than not, turn out to be conflicting: an improvement in any one of them may be accompanied by a worsening in others. Such problems are called multi-objective or multi-criterion optimization problems.

For simplicity, it is assumed that there is only one decision maker. It must be emphasised that the decision maker is actually searching for an optimal *compromise*, rather than for a hypothetical numerical optimum of a multi-objective criterion; that is, to some extent, the criterion serves as an artifice to arrive at an "optimal" decision.

As an example of a multi-objective optimization typical for the process industry one may consider the search for a recipe resulting in an end product satisfying a number of quality requirements. Another example is the development of a recipe involving a search for the recipe-item values which result in a product with added value as large as possible and, at the same time, with environmental pollution as small as possible.

In principle, a conventional multi-objective optimization problem can be formulated as follows [CHA83]:

$$\max_{\mathbf{x} \in X} \mathbf{J}(\mathbf{x}) = \max_{\mathbf{x} \in X} \left(J_1(\mathbf{x}), J_2(\mathbf{x}), \dots, J_k(\mathbf{x}) \right)^T \qquad (3.61)$$

where:

$\mathbf{J}(\mathbf{x})$	:	multi-objective criterion
$J_i : \mathbb{R}^n \rightarrow \mathbb{R}$	:	individual objective, i = 1,2,...,k

$\mathbf{x} \in X$: recipe items vector; here decision variable vector

$X = \{\mathbf{x} \in \mathbb{R}^n\colon\ g_j(\mathbf{x}) \geq 0,\ j = 1,...m\}$: the feasible area determined by a set of constraints

$g_j : \mathbb{R}^n \rightarrow \mathbb{R}$: constraint function

$\mathbb{R}$: space of real numbers.

A useful classification of the solution methods for multi-objective optimization problems is given by Ignizo, distinguishing three classes of methods [IGN82]:

1) methods based on some measure of optimality;
2) methods searching for Pareto-optimal solutions;
3) interactive methods.

3.6.1 Methods Based on a Measure of Optimality

Methods of this group make an attempt to measure alternatives in one way or another, for example, by weighting each objective and then optimizing their weighted sum. Methods of this category are: the weighting, the lexicographic and the so-called ε-constraint methods. They are described in Appendix F.

3.6.2 Methods Searching for Pareto-Optimal Solutions

Methods of this group work without assigning any weights to conflicting objectives and without ranking them. Here the optimal solution $\mathbf{x}_{\text{opt}}$, as defined by Pareto, appears to be the natural extension of optimizing a single criterion, in the sense that in multi-objective optimization any further improvement in any one objective requires a worsening of at least one other objective [PAR06]. If the individual objective functions are consistent (identical local optima), then there is no problem. The opposite - though - is usually the case and then the solution may be difficult to obtain. This is especially the case when the individual objective functions are non-linear. Further, often the solution, if found, it not what the user wished to find.

3.6.3 Interactive Methods

Interactive methods use the information obtained from the decision maker in an iterative process to assign appropriate importance levels, e.g. weights, to all individual objectives. During the solution process the user learns about the nature of the problem under consideration.

One of the methods of this group, which appeared, at first, to be quite useful to the recipe generation and improvement, was proposed by Woods [WOO85].

3.6.3.1 The Method of Woods

According to the method of Woods, information provided by the user consists of preferences, which follow from the comparison, time after time, of two sets of objective values **J**(**x**') and **J**(**x**'') corresponding to two different decision vectors **x**' and **x**''.

Such pair-wise comparisons of two individual objectives, which is quite easy but laborious, results in the ranking of the decision vectors. This ranking may then be used for estimation of the appropriate weighting factors. Next, based on the Nelder-Mead algorithm described in Section 3.5.1, and using this ranking information, a new decision vector $\mathbf{x}_{new}$ is generated, and the user must again compare it with all previous decision vectors. This iteration process will terminate when the weighting factors are found acceptable by the decision maker. Thus, the actual multi-objective problem is reduced to the optimization of a single criterion: the weighted sum of the individual objectives:

$$\max_{\mathbf{x}\in X} \sum_{i=1}^{k} \omega_i \ J_i(\mathbf{x}). \tag{3.62}$$

where:

ω_i : weighting factor.

Such a single-criterion optimization can be done by a standard optimization method, e.g. the SQP-method.

The method of Woods applied to the problem (3.61) works as follows:

(0) set $N = 1$, where N is an iteration number,
choose nd = n+1 different not-collinear decision vectors
$\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{nd}\}$, where "nd" represents the number of decision vectors,
choose starting values for the weighting factors
$\boldsymbol{\omega}^0 = [\omega_1{}^0, \omega_2{}^0, ..., \omega_k{}^0]^T = [1,...,1]^T$;

(1) compare **J**(**x**') with **J**(**x**'') for each pair {**x**', **x**''} of decision vectors $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{nd}\}$,
where $\mathbf{x}' \neq \mathbf{x}''$ and decide which one is better;

(2) from the comparisons estimate the weighting factor vector $\boldsymbol{\omega} = [\omega_1^N, \omega_2^N, \ldots, \omega_k^N]^T$ by optimizing the following criterion, which expresses the basic idea of the method to change the weights as little as possible at each iteration:

$$\min_{\boldsymbol{\omega}^N} \sum_{i=1}^{k} (\omega_i^{N-1} - \omega_i^N)^2 \tag{3.63}$$

subject to:

$$\sum_{i=1}^{k} \omega_i^N = k \qquad \text{(normalizing constraint)} \tag{3.64}$$

$$\omega_i^N \geq 0 \text{ , } i = 1,\ldots,k; \qquad \text{(non-negativity constraint)} \tag{3.65}$$

$$\mathbf{W}^N \boldsymbol{\omega} \geq 0 \qquad \text{(preference constraint)} \tag{3.66}$$

where:

$$\mathbf{W}^N = \begin{vmatrix} J_1^1(\mathbf{x}')-J_1^1(\mathbf{x}'') & J_2^1(\mathbf{x}')-J_2^1(\mathbf{x}'') & \ldots & J_k^1(\mathbf{x}')-J_k^1(\mathbf{x}'') \\ J_1^2(\mathbf{x}')-J_1^2(\mathbf{x}'') & J_2^2(\mathbf{x}')-J_2^2(\mathbf{x}'') & \ldots & J_k^2(\mathbf{x}')-J_k^2(\mathbf{x}'') \\ \ldots\ldots & \ldots & & \ldots \\ J_1^{np}(\mathbf{x}')-J_1^{np}(\mathbf{x}'') & J_2^{np}(\mathbf{x}')-J_2^{np}(\mathbf{x}'') & \ldots & J_k^{np}(\mathbf{x}')-J_k^{np}(\mathbf{x}'') \end{vmatrix} \tag{3.67}$$

np : number of compared pairs ($\mathbf{x}'$, $\mathbf{x}''$) formed from the decision vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{nd}\}$, whereby $\mathbf{x}'$ is judged to be better than $\mathbf{x}''$

(3) **if** the weighting factors $\omega_1^N, \omega_2^N, \ldots, \omega_k^N$ are accepted by the decision maker **or** they have converged, **then**
optimize the single-criterion problem:

$$\max_{\mathbf{x} \in X} \sum_{i=1}^{k} \omega_i^N J_i(\mathbf{x}) \tag{3.62'}$$

with the found weighting factors and exit

else

use a Nelder-Mead algorithm to obtain a new decision vector $\mathbf{x}_{new}$, to be added to the set of decision vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{nd}\}$

```
        used in (1), set nd = nd + 1, set $N = N + 1$ and go to (2)
    endif.
```

As one can see, the addition of a new decision vector $\mathbf{x}_{\text{new}}$ has the consequence that the matrix $\mathbf{W}^{N+1}$ contains np rows more than $\mathbf{W}^{N}$, and $\mathbf{W}^{N+2}$ np+1 rows more than $\mathbf{W}^{N+1}$, which means that the decision maker must compare more and more pairs in each iteration. If his decisions are not consistent, which may occur when the number of pairs is very large, then the algorithm may end in a dead loop. This is indeed a great disadvantage of the method of Woods. This was the motivation to develop a new, quite different interactive strategy, which is much more useful during recipe generation and/or improvement because it deals more directly with the quintessence of the multi-objective optimization. It will be described below.

3.6.3.2 The Triplet-Choice Multi-Objective Method

It became apparent that a method, based on identification of the best, the worst, and the second worst process outcomes in each iteration, e.g. as the Nelder-Mead method described in Section 3.5.1, can also be used for multi-objective optimization after a number of modifications.

The new method, called "the Triplet-choice Multi-Objective Method" is also based on a triplet-choice as in the Nelder-Mead method. After the decision maker has chosen the best, the worst and the second worst process outcomes, the procedure suggest a new test run. Next, the decision maker must compare the results of the new test run proposed by the algorithm with the best, worst and second worst test runs to identify a new triplet, and the procedure repeats. The decisions of the user are usually quite easy. Also, the algorithm is not likely to be troubled with inconsistent choices as in the pair-wise comparisons of the method of Woods. Moreover, for all multi-objective optimization methods described above, all individual objectives must be defined beforehand as mathematical functions, for which often experiments must be performed to estimate process models for each response involved. On the contrary, the Triplet-choice Multi-Objective Method starts with a set of well-chosen test runs and without any models. It is also not necessary that the individual objectives are formulated mathematically, it is enough to know what they are, and that they can be of various types, e.g. taste, colour, appearance, quality, property, process time or profit. In other words, it is sufficient that each individual objective J_i is an operator:

$$J_i: \quad \mathbb{R}^n \quad \rightarrow \quad \Re \tag{3.68}$$

where:

$\mathbb{R}$: space of real numbers

$\Re$: arbitrary space, e.g. of taste, colour or appearance.

Moreover, the number of the individual objectives may be enlarged or reduced during optimization. Besides, if all responses involved may be measured in real units, i.e.

$$J_i: \quad \mathbb{R}^n \quad \rightarrow \quad \mathbb{R}, \tag{3.69}$$

then the decision maker can decide, when enough information is gathered to estimate process models, to stop with real test runs and to continue with the models to get the value of each individual criterion. In this manner the number of test runs can be limited. The procedure stops when the decision maker has found recipe-item values which correspond, in his opinion, to the best choice for the given multi-objective problem.

This method can also be used for solving so-called end-specification problems that are typical for the recipe generation phase. The **end-specification problem** can be defined as follows:

Find X^{spec}

where:

$$X^{spec} = \{\mathbf{x} \in X : \ \mathbf{y}_{MIN} \leq \mathbf{y} \leq \mathbf{y}_{MAX}\} \tag{3.70}$$

$\mathbf{x} \in X \subset \mathbb{R}^n$: recipe items vector

X : the feasible recipe items area

$\mathbf{y}$: process response vector

$\mathbf{y}_{MIN}, \mathbf{y}_{MAX}$: lower, upper bound on **y**, respectively.

The individual wishes regarding each response can be seen as individual objectives of multi-objective optimization.

Section 6.1 shows an example of such a problem tackled with the Triplet-choice Multi-Objective Method, which does not investigate the whole X^{spec}, but only its elements.

This method can also be useful when process models for each response of interest are available. In principle, by solving the inverse problem X^{spec} can be found, but the solution of such an inverse problem is often impossible. On the contrary, the Triplet-choice Multi-Objective Method can always help the user to find the best process conditions.

The Triplet-choice Multi-Objective Method works as follows:

```
(0)  set $N = 0$;
     decide which individual objectives are involved;
     choose n+1 sets of the values of recipe items $(\mathbf{x}_1^N,\ldots, \mathbf{x}_n^N, \mathbf{x}_{n+1}^N)$;
     carry out n+1 test runs according to the chosen recipe item values;
(1)  from the n+1 sets of the recipe item values select the triplet
     $(\mathbf{x}_b^N, \mathbf{x}_{w-1}^N, \mathbf{x}_w^N)$: $\mathbf{x}_b^N$ with the best, $\mathbf{x}_{w-1}^N$ with the second worst
     and $\mathbf{x}_w^N$ with the worst process outcome;
(2)  compute the centroid $\mathbf{x}^N$ as defined by (3.49) in the Nelder-Mead method:
     $\bar{\mathbf{x}}^N = ( \sum_{i=1}^{n+1} \mathbf{x}_i^N - \mathbf{x}_w^N) / n$                    (3.49)
(3)  if $N = 0$ then go to (4);
     else
          decide if the set of the individual objectives involved must be enlarged or reduced;
          if the set of the individual objectives is not changed then
          if you are satisfied with $\mathbf{x}_b^N$, exit
          endif,
     endif;
(4)  compute the reflection point $\mathbf{x}^r$ according to (3.44):
               $\mathbf{x}^r = (1+\alpha)\ \bar{\mathbf{x}}^N - \alpha\ \mathbf{x}_w^N$;        e.g. $\alpha = 1$        (3.44)

     if $\mathbf{x}^r$ not feasible, then apply (4.1) ÷ (4.3)

     (4.1) set:   $\delta_{low} = 0$;   $\delta_{up} = 1$;   $\delta_{good} = 1$;   $\delta = 1/2$;   $j = 0$;
     (4.2) compute $\mathbf{x}^{sc}$ according to the definition (3.47):
           $\mathbf{x}^{sc} = \delta\ \bar{\mathbf{x}}^N + (1 - \delta)\ \mathbf{x}^r$;                  (3.47)
           if $\mathbf{x}^{sc}$ feasible then
                $\delta_{good} = \delta$;   $\delta_{up} = \delta$;   $\delta = (\delta_{low} + \delta) / 2$;
           else
                $\delta_{low} = \delta$;   $\delta = (\delta_{up} + \delta) / 2$;
           endif;
           $j = j + 1$; repeat (4.2) until $j = 5$.
```

```
    (4.3) if   $\delta_{good} > 0.99$, then compute $\mathbf{x}^c$ according to (3.46):
                   $x^c = \gamma\, \mathbf{x}_w^N + (1 - \gamma)\, \bar{\mathbf{x}}^N$;  e.g. $\gamma = 1/2$        (3.46)
                   replace $\mathbf{x}^r$ by $\mathbf{x}^c$
               else
                   replace $\mathbf{x}^r$ by $\mathbf{x}^{sc}$
               endif.
          endif;
(5)       according to Figure 3.14 accept expansion, reflection, contraction, shadow contraction or decide to shrink;
          if $\mathbf{x}^e$ is not feasible, then expansion is skipped and $\mathbf{x}^r$ is accepted
          endif;
(6)       if shrinkage is done, then
               set $\mathbf{x}_i^{N+1} = \eta\, \mathbf{x}_i^N + (1 - \eta)\, \mathbf{x}_b^N$,        $i = 1,..., n+1$;
               carry out n test runs according to the recipe item values
               $\mathbf{x}_i^{N+1}$, where $i = 1,.., n+1$ and $\mathbf{x}_i^{N+1} \neq \mathbf{x}_b^N$;
          else set:
               $\mathbf{x}_i^{N+1} = \mathbf{x}_i^N$        for $\mathbf{x}_i^N \neq \mathbf{x}_w$ , $i = 1,..., n+1$   (3.71)

               $\mathbf{x}_i^{N+1} = \begin{cases} \mathbf{x}^r & \text{if reflection accepted} \\ \mathbf{x}^e & \text{if expansion accepted} \\ \mathbf{x}^c & \text{if contraction accepted} \\ \mathbf{x}^{sc} & \text{if shadow contraction accepted} \end{cases}$
                                            for $\mathbf{x}_i^N = \mathbf{x}_w$ , $i = 1,..., n+1$   (3.72)

               carry out one test run according to $\mathbf{x}_i^{N+1}$ defined as in (3.72);
          endif;
(7)       set $N = N+1$ and go to (1), unless $N \geq Nmax$.
```

This section started by describing a number of methods known from the literature, that are proposed for solving multi-objective optimization problems. Because the assumptions on which these methods are based can be quite difficult to satisfy in real situations, we developed a useful method based on only three choices: the user selects the best, the worst and second worst set of recipe item values. The most important advantage of the proposed method is its universal character expressing itself in simplicity of application and in assumptions about the individual objectives.

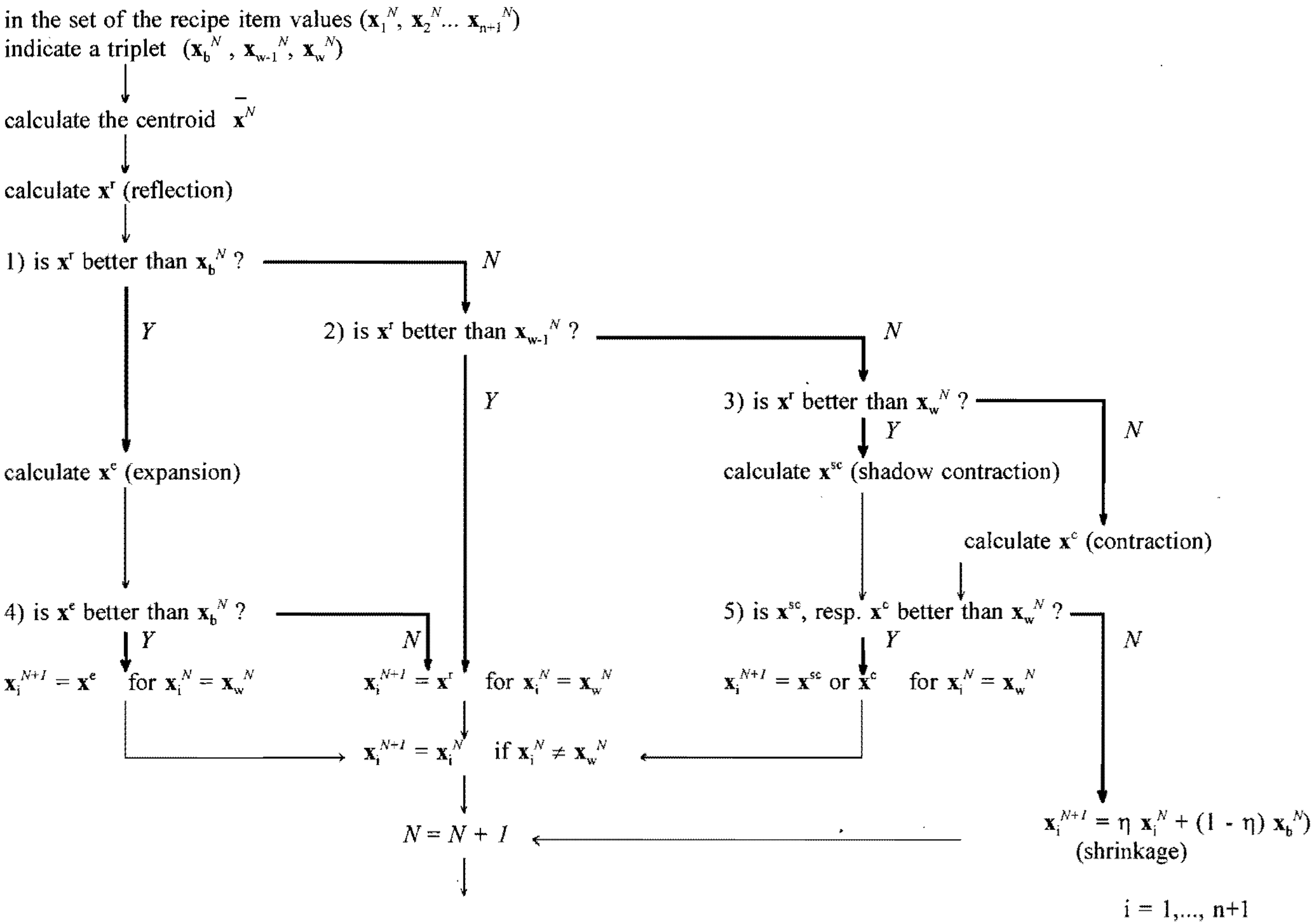

in the set of the recipe item values ($\mathbf{x}_1^N, \mathbf{x}_2^N \dots \mathbf{x}_{n+1}^N$)
indicate a triplet ($\mathbf{x}_b^N, \mathbf{x}_{w-1}^N, \mathbf{x}_w^N$)
calculate the centroid $\bar{\mathbf{x}}^N$
calculate $\mathbf{x}^r$ (reflection)
1) is $\mathbf{x}^r$ better than $\mathbf{x}_b^N$?
N
Y
2) is $\mathbf{x}^r$ better than $\mathbf{x}_{w-1}^N$?
N
Y
3) is $\mathbf{x}^r$ better than $\mathbf{x}_w^N$?
Y
N
calculate $\mathbf{x}^e$ (expansion)
calculate $\mathbf{x}^{sc}$ (shadow contraction)
calculate $\mathbf{x}^c$ (contraction)
4) is $\mathbf{x}^e$ better than $\mathbf{x}_b^N$?
Y
N
5) is $\mathbf{x}^{sc}$, resp. $\mathbf{x}^c$ better than $\mathbf{x}_w^N$?
Y
N
$\mathbf{x}_i^{N+1} = \mathbf{x}^e$ for $\mathbf{x}_i^N = \mathbf{x}_w^N$
$\mathbf{x}_i^{N+1} = \mathbf{x}^r$ for $\mathbf{x}_i^N = \mathbf{x}_w^N$
$\mathbf{x}_i^{N+1} = \mathbf{x}^{sc}$ or $\mathbf{x}^c$ for $\mathbf{x}_i^N = \mathbf{x}_w^N$
$\mathbf{x}_i^{N+1} = \mathbf{x}_i^N$ if $\mathbf{x}_i^N \neq \mathbf{x}_w^N$
$N = N + 1$
$\mathbf{x}_i^{N+1} = \eta\ \mathbf{x}_i^N + (1 - \eta)\ \mathbf{x}_b^N)$
(shrinkage)
i = 1,..., n+1

Figure 3.14 Flow diagram describing possible simplex operations during one iteration *N* of the Triplet-Choice Multi-Objective Method. Decision flows, corresponding to the (5)-th step of the algorithm, are presented with bold arrows.

Chapter 4. Run-Time Application of a Recipe Adaptation Set to Batch Initialization and Batch Correction

This chapter describes two run-time applications of a recipe adaptation set, namely to batch initialization and to batch correction. As introduced in Chapter 2, a distinction is made between batch initialization and batch correction in order to distinguish between adjustment of a control recipe just before the beginning of the process and adjustment during processing.

4.1. Batch Initialization

Suppose a deviation from the prescribed initial conditions becomes known at the start of the batch. For example, the reactor contains more (or less) material than intended, the available processing time must be shorter than prescribed or, maybe, the prices of feedstocks, products or energy differ from the values holding at the moment the master recipe was created. In that case it is worthwhile to compensate for these known deviations by adjustment of available recipe items, which may be adjustable initial conditions as well as time-dependent profiles during subsequent batch execution. A control recipe generated in this manner leads to efficient *batch initialization*, see Figure 4.1.

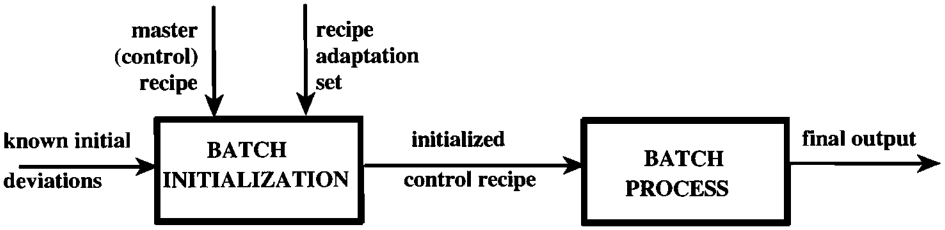


Figure 4.1 Batch initialization scheme

4.1.1 Batch-Initialization Procedure

In the framework of the FRIS-approach, the batch initialization procedure searches for the best control recipe by optimizing the performance criterion with regard to the accepted process model, defined constraints and known disturbances. The model used for batch initialization is either the nominal process model or an auxiliary model (mentioned in Section 2.3.2 as the sixth, respectively seventh component of a recipe adaptation set), depending on which model is valid under the disturbed conditions. Section 3.2 describes how such a model can be obtained.

Batch initialization comes down to the following simple, but effective, procedure:

1) detect initial deviations in process conditions, and changes in feedstocks or prices;
2) define the goal in the form of a performance criterion, which may be the same as used during generation of the corresponding master recipe;
3) add new constraints, if necessary;
4) choose adjustable initialization variables;
5) choose a valid initialization model;
6) optimize the performance criterion subject to the initialization model, constraints and known deviations;
7) present the optimized settings to the user for acceptance/rejection;
8) accept or reject the presented settings;
9) if the settings are accepted, then prepare an initialized control-recipe, else go to 2 or exit;
10) start the batch according to the initialized control-recipe.

The steps 1 ÷ 4, 8 and 10 are done by a user in Production, and the steps 5 ÷ 7 and 9 are done by the Initialization Module in the FRIS-package, which will be described in Chapter 5.

4.1.2 Batch-Initialization Example

The example fermentation-process described in Sections 3.1.2.÷ 3.2.2 is used here to illustrate how batch initialization works (see also Appendix B). The corresponding process model was developed in Section 3.2.2.

Suppose that the initial temperature Tb does not correspond to the master recipe (it being 28°C instead of 29.4°C), and that the temperature slope Sl can be adjusted, while the other recipe items, including sugar dosage Sd, have to remain fixed at their nominal levels. The initialized control recipe presented in the last column of Table 4.1 is obtained by maximization of the final amount of desired product Pf:

$$\max_{Sl} J = \max_{Sl} Pf$$

subject to the accepted process model of Pf, shown in Formula (3.25):

$$Pf = -748.031 + 60.878\ Tb + 273.756\ Sl + 29.807\ Sd - 11.150\ Tb \cdot Sl - 1.225\ Tb^2 - 25.926\ Sl^2 \qquad (3.25)$$

and the constraints:

$Sd = 1.15$	:	sugar dosage at the nominal level
$Tb = 28$	:	known disturbed initial temperature
$-1 \leq Sl \leq -0.5$	:	constraints on the temperature slope, which is here the optimization variable.

Table 4.1 The results of batch initialization applied to the fermentation process. The disturbed recipe item (initial temperature) is presented in italics, the initialized item in bold letters.

	Optimal settings for undisturbed situation (Table 3.10)	**"Disturbed" control recipe**	**Initialized control recipe**
Initial temperature Tb	29.4 °C	*28.0 °C*	*28.0 °C*
Temperature slope Sl	-1.0 °C/h	- 1.0 °C/h	**- 0.74 °C/h**
Sugar dosage Sd	1.15 kg	1.15 kg	1.15 kg
Performance index: final amount of the product Pf	45.73 penicillin units	43.30 penicillin units	45.02 penicillin units

The expected final product amount according to the initialized control recipe is 45.02. In the optimal undisturbed situation Pf would have been 45.73, whereas without any adjustment it would have been only 43.30.This simple example shows what profit can be expected with batch initialization. Moreover, the efficient batch initialization helps avoid production out of specs. Our experience is, however, that in present industrial practice little attention is paid to the appropriate batch initialization, also when it is very easy to realize.

4.2. Batch Correction

Batch initialization, performed at the beginning of the batch phase, takes into account known initial deviations only. Often there are also unmeasured deviations at the moment the batch begins, further disturbances which occur during processing and deviations between process model and actual behaviour. Provided that continuous (e.g temperature measurements) and/or sample measurements are available, it may be possible, to a certain degree, to compensate for the effects of unknown disturbances. As soon as the deviation is detected, the control recipe can be adapted, as presented in Figure 4.2, for the remaining period of time in order to optimally approximate the desired final process conditions. This is called *batch correction* [RIJ92].

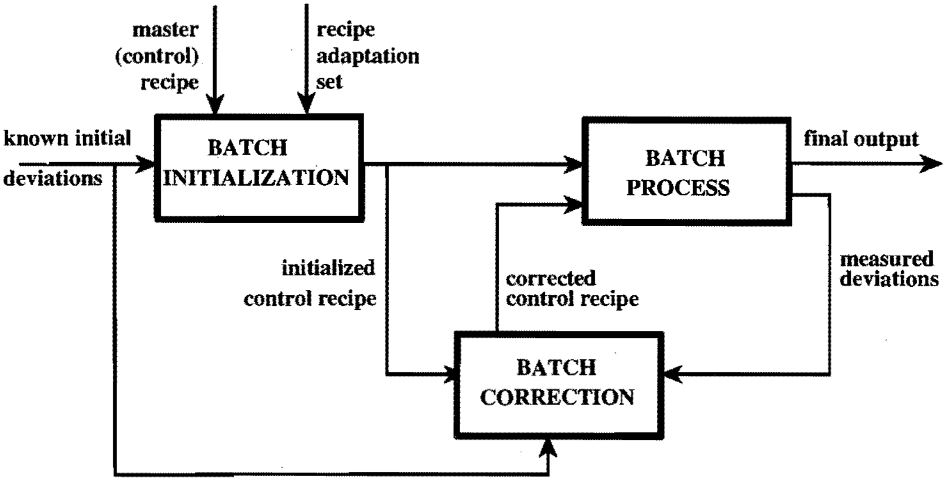


Figure 4.2 Batch initialization and correction scheme.

Whereas batch initialization is a fairly if not completely new approach to improving batch processes, batch correction has received a great attention from control engineers in the area of predictive control and optimal control. The

FRIS-approach, thus far mainly based on black-box transition models, proposes another way to obtain a corrected recipe. How a corrected recipe can be obtained on the basis of white models is described in [THI94].

As pointed out in Section 2.3.2, apart from the performance criterion and the knowledge of additional plant, process and economic constraints, the so-called correction information, included in a recipe adaptation set, is necessary for correcting a batch during processing. The elements of the correction information are:

a) sample and correction moments, t_S and t_C, respectively;
b) process outputs measured at the sample moment(s);
c) correction variables;
d) at least two correction models.

The expected final product amount according to the initialized control recipe is 45.02. In the optimal undisturbed situation Pf would have been 45.73, whereas without any adjustment it would have been only 43.30.For ease of explanation, let us restrict the following to the case that only measurements of one single sample become available during the run.

4.2.1 Sample and Correction Moments

Sample and correction moments, denoted further by t_S and t_C respectively, have to be chosen carefully. They will be used not only for determining t_S and t_C, but also during development of the correction models.

Due to the time needed for the analysis, the correction can only be made some time after the sample moment t_S, at moment t_C. The timing of t_S and t_C is not quite known a priori. If chosen too late, then there might be not enough time for the correction to take effect, whereas if chosen too early, then sample information about deviations might be too uncertain. The first choice can be made on basis of the experience and opinion of the production staff. Usually, the production staff will already have chosen a convenient sample time, but that may not be the best choice if model-based correction are to be the most effective. Therefore, it makes sense to try by experimentation whether a shifting of the chosen moments is advantageous.

4.2.2 Process Outputs Measured at the Sample Moment

The user must decide what process outputs, denoted by $\mathbf{y}^S$, will be measured by the analysis of a sample. Note that here the output variables correspond to measured composition quality, or other process variables, so they are not necessarily the same as the process outputs involved in the performance criterion (3.26) or (3.27).

4.2.3 Correction Variables

Correction variables, for example: extra ingredients, which may be added at t_C, or the processing time, or the slopes of time-dependent temperature or pressure profiles, provide degrees of freedom for batch correction. They are denoted by $\mathbf{x}^C$.

4.2.4 Correction Models

Batch correction requires at least two correction models, that will be further called *Model_C1* and *Model_C2*.

Model_C1
This model describes the operating conditions at the sample moment as a function of initial process conditions. Its general form is as follows:

$$\mathbf{y}^S = \mathbf{f}_S(\mathbf{x}(t_0)) \tag{4.1}$$

where:

$\mathbf{y}^S$	:	process output vector at the sample time t_S;
$\mathbf{f}_S$	:	correction-model function describing all components of $\mathbf{y}^S$ as a function of the recipe items $\mathbf{x}(t_0)$
$\mathbf{x}(t_0)$	:	vector with recipe items at t_0.

The model *Model_C1* may be obtained in the same manner as the nominal or an auxiliary process model, e.g. by factorial experimentation. To reduce the number of test runs, it is recommendable to estimate this model simultaneously with the nominal process model.

Model_C2
This model describes the ultimate effect of the values measured at t_S as well as any run-time corrections applied during the remainder of the processing time:

$$\mathbf{y}^F{}_C = \mathbf{f}_C(\mathbf{y}^S{}_m, \mathbf{x}^C) \tag{4.2}$$

where:

$\mathbf{y}^F{}_C$: process output vector at t_F due to the correction at t_C; this vector corresponds to the process output vector involved in the performance criterion used for recipe generation, improvement or batch initialization;

t_C : beginning of the correction $t_S < t_C < t_F$;

$\mathbf{y}^S{}_m$: measured process output at t_S

$\mathbf{f}_C$: correction-model function;

$\mathbf{x}^C$: vector of correction variables; if $\mathbf{x}^C = 0$, then *Model_C2* tells what the expected final process output is due to zero correction, i.e. without any subsequent control action.

To obtain *Model_C2*, the correction variables $\mathbf{x}^C$ should be varied according to a chosen experiment design method. Because $\mathbf{y}^S{}_m$ cannot be varied freely, the initial factors $\mathbf{x}(t_0)$ may be varied, if possible, so that the variations in $\mathbf{y}^S{}_m$ are visible, in other cases the natural variations in $\mathbf{y}^S{}_m$ are given by the process.

4.2.5 Batch-Correction Procedure

If the correction information is available, it can be decided during each batch whether a correction is needed and if so, what kind. The procedure is as follows:

1) choose sample and correction moment, correction variables and outputs to be measured at the sample moment;
2) define the goal in the form of a performance criterion used for batch correction, denoted further by J^C; it will often be the same criterion as for batch initialization;
3) define constraints on $\mathbf{x}^C$ and $\mathbf{y}^F{}_C$, if necessary;
4) predict $\mathbf{y}^S$ according to *Model_C1* and the known initial conditions $\mathbf{x}(t_0)$;
5) take and analyze a sample at t_S to obtain $\mathbf{y}^S{}_m$;
6) use *Model_C2* with $\mathbf{x}^C = 0$ to predict the final value of the process output $\mathbf{y}^F{}_C$ without any correction,
7) **if** ($\mathbf{y}^F{}_C$ is not satisfactory or $|\ \mathbf{y}^S - \mathbf{y}^S{}_m\ | \geq \boldsymbol{\delta}_{COR}$)
   where $\boldsymbol{\delta}_{COR}$ denotes the allowed deviation from the expected values $\mathbf{y}^S$,
   then compute the corrections $\mathbf{x}^C$ by solving the following optimization problem:

$$\max_{\mathbf{x}^C} J^C$$

```
        with regard to *Model_C2* and the constraints
    else
        no correction is suggested;
8)  prepare a corrected control recipe;
9)  replace the previous control recipe by the corrected control recipe.
```

The steps 1 ÷ 3, 5 and 9 are done by a user in Production, the steps 4 and 6 ÷ 9 are done by the Correction Module in FRIS. Of course, batch correction can also be an actual issue in R&D.

4.2.6 Batch-Correction Example

The example of the fermentation process illustrates the procedure of batch correction.

To establish the deviation between the measured and the expected process condition at t_S = 4h, the following models of type *Model_C1* for $\mathbf{y}^S$ = [M4h, P4h] have been determined:

$$M4h = -20.675 + 1.898\ Tb + 9.780\ Sl - 5.929\ Sd - 0.323\ Tb \cdot Sl + 0.337\ Tb \cdot Sd - 0.039\ Tb^2 \tag{4.3}$$

$$P4h = -160.056 + 12.648\ Tb + 27.401\ Sl + 22.660\ Sd - 0.960\ Tb \cdot Sl - 0.735\ Tb \cdot Sd - 3.018\ Sl \cdot Sd - 0.244\ Tb^2 \tag{4.4}$$

where:

M4h : total biomass in the reactor at t_S= 4h
P4h : amount of product in the reactor at t_S= 4h.

The models (4.3) and (4.4) are obtained on the basis of the same 21 test runs (see Tables 3.1 ÷ 3.5) used for the estimation of the initialization model (3.25), but with interim measurements performed at the sample moment t_S = 4h. The statistical information about the corresponding scaled models is presented in Table 4.2 and in Table 4.3.

Table 4.2 Process model and the analysis of variance for the response M4h

```
Least Squares Coefficients, Response M4h, Model MODEL_C1_M4h
1 Term          2 Coeff.   3 Std. Error  4 T-value  5 Signif.  6 Transformed Term
----------------------------------------------------------------------------------
1 1              4.605636    0.018401
2 ~Tb            0.342700    0.019299                            ((Tb-2.85e+01)/1.5)
3 ~Sl            0.146100    0.019299                            (Sl+7.5e-01)/2.5e-01)
4 ~Sd            0.549800    0.019299                            ((Sd-1)/1.5e-01)
5 ~Tb*Sl        -0.121000    0.021577     -5.61      0.0001
6 ~Tb*Sd         0.075750    0.021577      3.51      0.0035
7 ~Tb**2        -0.088736    0.026665     -3.33      0.0050
No. cases = 21        R-sq. =  0.9888    RMS Error = 0.06103
Resid. df = 14   R-sq-adj. =  0.9840      Cond. No. = 2.335
~ indicates factors are transformed.
                     R-PRESS  = 0.978
==================================================================================
Least Squares Summary ANOVA, Response M4h,  Model MODEL_C1_M4h
Source             df     Sum Sq.    Mean Sq.   F-Ratio  Signif.
----------------------------------------------------------------------------------
1 Total(Corr.)     20    4.667107
2  Regression       6    4.614963    0.769161   206.50   0.0000
3   Linear          3    4.410685    1.470228   394.70   0.0000
4   Non-linear      3    0.204278    0.068093    18.28   0.0000
5  Residual        14    0.052144    0.003725
6   Lack of fit     8    0.021307    0.002663     0.52   0.8087
7   Pure error      6    0.030837    0.005139
Model obeys hierarchy. The sum of squares for linear terms
is computed assuming nonlinear terms are first removed.
F(8,6) as large as 0.5182 is not a rare event =>
    no evidence of lack of  it.
```

Table 4.3 Process model and the analysis of variance for the response P4h

```
Least Squares Coefficients, Response P4h, Model MODEL_C1_P4h
1 Term          2 Coeff.    3 Std. Error  4 T-value  5 Signif.  6 Transformed Term
----------------------------------------------------------------------------------
1 1              6.280873    0.026076
2 ~Tb           -1.900150    0.027349                            (Tb-2.85e+01)/1.5)
3 ~Sl           -0.744950    0.027349                            (Sl+7.5e-01)/2.5e-01)
4 ~Sd            0.594820    0.027349                            ((Sd-1)/1.5e-01)
5 ~Tb*Sl        -0.360037    0.030577      -11.77    0.0001
6 ~Tb*Sd        -0.165463    0.030577       -5.41    0.0001
7 ~Sl*Sd        -0.113188    0.030577       -3.70    0.0027
8 ~Tb**2        -0.548663    0.037788      -14.52    0.0001
No. cases = 21        R-sq. =  0.9980    RMS Error = 0.08648
Resid. df = 13   R-sq-adj. =  0.9969      Cond. No. = 2.335
~ indicates factors are transformed.
                     R-PRESS  = 0.995
==================================================================================
Least Squares Summary ANOVA, Response P4h,  Model MODEL_C1_P4h
Source              df     Sum Sq.     Mean Sq.   F-Ratio  Signif.
----------------------------------------------------------------------------------
1 Total(Corr.)      20    48.22590
2  Regression        7    48.12867    6.87552    919.30   0.0000
3   Linear           3    45.19331   15.06444   2014.00   0.0000
4   Non-linear       4     2.93536    0.73384     98.11   0.0000
5  Residual         13     0.09723    0.00748
6   Lack of fit      7     0.03332    0.00476      0.45   0.8420
7   Pure error       6     0.06391    0.01065
Model obeys hierarchy. The sum of squares for linear terms
is computed assuming nonlinear terms are first removed.
F(7,6) as large as 0.4469 is not a rare event =>
    no evidence of lack of fit.
```

Table 4.4 presents the predicted process outputs at t_S: M4h and P4h, together with their 95% confidence bounds for the nominal values of the recipe items Tb, Sl and Sd. The correction rule for this example is: if the measured values of M4h or P4h are not inside their 95% confidence intervals, then correction should be carried out.

Table 4.4 Predicted process outputs at t_S: M4h and P4h, with their 95% confidence bounds according to the models (4.2) and (4.3)

Process output at t_s	Lower bound	**Predicted process outputs at t_s**	Upper bound
M4h	5.136	**5.301**	5.466
P4h	6.225	**6.513**	6.801

In this example, the means for coping with detected deviations happen to be rather limited for practical reasons: the deviations can only be compensated for by decreasing or increasing the temperature during the rest of the batch phase, so the only correcting variable is the slope of the temperature Sl_cor after the correction moment t_C= 5h.

To predict the final product amount in the case of measured deviations, and to calculate the influence of the correction variable on the final output, the following model of type *Model_C2* for $\mathbf{y}^F_c$ = [Pf_cor] is found:

$$\begin{aligned} \text{Pf\_cor} = \; & 0.990 + 2.493\ \text{P4h} + 0.952\ \text{M4h} - 14.743\ \text{Sl\_cor} \\ & + 1.205\ \text{P4h} \cdot \text{M4h} + 2.211\ \text{P4h} \cdot \text{Sl\_cor} - 0.397\ \text{P4h}^2 \end{aligned} \qquad (4.5)$$

where:

Pf_cor : final amount of product after a correction of the temperature slope at t_C= 5h.

For the estimation of the correction model (4.5), 25 process measurements presented in Table 4.5 are used: 11 measurements without any correction, - already presented in Table 3.1 as factorial experiments and here marked with asterisk "*", and 14 additional measurements with a corrected temperature slope Sl_cor. For comparison, Table 4.5 presents the initial temperature slope Sl in Column 4. Table 4.6 shows the statistical information corresponding to the model (4.5).

Table 4.5 Worksheet with the test runs used for the estimation of the correction model (4.5)

0 Case	1 M4h	2 P4h	3 Sl_cor (SlC)	4 Sl initial	5 Pf_cor	6 Fitted Pf_cor	7 Raw Resid.
1*	4.355	2.411	-0.50	-0.50	26.08	26.20	-0.12
2	4.355	2.411	-0.75	-0.50	28.12	28.56	-0.44
3*	3.845	6.513	-0.50	-0.50	34.87	34.41	0.46
4	3.845	6.513	-0.75	-0.50	35.66	34.49	1.17
5*	5.401	2.929	-0.50	-0.50	33.48	33.23	0.25
6	5.401	2.929	-0.75	-0.50	35.28	35.30	-0.02
7*	4.610	6.266	-0.75	-0.75	40.77	40.90	-0.13
8	4.610	6.266	-0.50	-0.75	40.00	40.68	-0.68
9	4.610	6.266	-1.00	-0.75	40.60	41.12	-0.52
10*	4.389	8.849	-1.00	-1.00	38.15	38.14	0.01
11	4.389	8.849	-0.75	-1.00	39.72	39.35	0.37
12*	5.355	5.426	-1.00	-1.00	45.73	45.70	0.03
13	5.355	5.426	-0.75	-1.00	45.31	45.01	0.30
14*	4.849	7.912	-0.50	-0.50	44.70	45.35	-0.65
15	4.849	7.912	-0.75	-0.50	45.51	44.67	0.84
16*	4.610	6.266	-0.75	-0.75	40.65	40.90	-0.25
17	4.610	6.266	-1.00	-0.75	41.11	41.12	-0.01
18	4.610	6.266	-0.50	-0.75	40.56	40.68	-0.12
19*	3.505	7.310	-1.00	-1.00	30.05	30.81	-0.76
20	3.505	7.310	-0.75	-1.00	30.99	31.21	-0.22
21*	4.610	6.266	-0.75	-0.75	40.06	40.90	-0.84
22	4.610	6.266	-1.00	-0.75	41.37	41.12	0.25
23	4.610	6.266	-0.50	-0.75	40.82	40.68	0.14
24*	4.219	4.373	-1.00	-1.00	35.91	35.63	0.28
25	4.219	4.373	-0.75	-1.00	35.02	34.36	0.66

To follow the progress of the process and to decide whether a run-time correction is needed or not, a sample of the reaction mixture is taken at the moment t_S = 4h. The analysis reveals that the total biomass is smaller and the total product amount is higher than expected: the measured value of M4h is 3.8 instead of 5.3, and that of P4h is 7.7 instead of 6.5. The adjustment of the control recipe is found by calculating the control action that compensates for the ultimate effect of the disturbance and, at the same time, optimises the performance criterion.

Therefore, the following non-linear optimization problem with linear constraints is solved:

$$\max_{Sl} \; J^C = \max_{Sl} \; \mathrm{Pf\_cor}$$

subject to the constraints:

$$\mathrm{M4h} = 3.8$$
$$\mathrm{P4h} = 7.7$$
$$-1 \leq \mathrm{Sl\_cor} \leq -0.5$$

where:

Pf_cor : final product amount after correction, for which model (4.5) was found
M4h : total biomass at t_S, for which model (4.3) is found
P4h : product amount at t_S, for which model (4.4) is found.

Table 4.6 Correction model and the analysis of variance for the response Pf_cor

```
Least Squares Coefficients, Response Pf_cor, Model MODEL_C2_Pf

1 Term          2 Coeff.    3 Std. Error  4 T-value  5 Signif. 6 Transformed Term
----------------------------------------------------------------------------------
1 1             39.595577     0.155736
2 ~M4h          12.274859     0.364200                              ((M4h-4.5)/1.5)
3 ~P4h           5.984360     0.309405                              ((P4h-6)/4)
4 ~S1C          -0.368744     0.157909                              (S1C+7.5e-01)/2.5e-01)
5 ~M4h*P4h       7.230754     0.854764       8.46     0.0001
6 ~P4h*S1C       2.211310     0.464793       4.76     0.0002
7 ~P4h**2       -6.349276     0.576840     -11.01     0.0001

No. cases = 25         R-sq. =   0.9914     RMS Error = 0.5753
Resid. df = 18     R-sq-adj. =   0.9885      Cond. No. = 3.306
~ indicates factors are transformed.

                     R-PRESS   = 0.984

==================================================================================

 Least Squares Summary ANOVA, Response Pf_cor  Model MODEL_C2_Pf

Source              df    Sum Sq.     Mean Sq.    F-Ratio  Signif.
-----------------------------------------------------------------------------------
1 Total(Corr.)      24    691.0764
2  Regression        6    685.1180    114.1863    345.00   0.0000
3   Linear           3    549.4985    183.1662    553.30   0.0000
4   Non-linear       3    135.6195     45.2065    136.60   0.0000
5  Residual         18      5.9583      0.3310
6   Lack of fit     12      5.0114      0.4176      2.65   0.1207
7   Pure error       6      0.9469      0.1578

Model obeys hierarchy. The sum of squares for linear terms
is computed assuming nonlinear terms are first removed.
F(12,6) as large as 2.646 is not a rare event =>
    no evidence of lack of fit.
```

In this example it was found that the temperature should be decreased by -0.5 °C/h after the correction moment instead of by -1 °C/h. Table 4.7 shows the performance criterion values in the disturbed situation without correction and with optimized control. The profit is evident.

Table 4.7 The results of batch correction applied at t_C to the fermentation process. The expected total biomass at t_S = 4h is M4h = 5.3 and the amount of product is P4h = 6.5. The measured values are: M4h_meas = 3.8 and P4h_meas = 7.7.

	"Disturbed" recipe without correction	**"Disturbed" recipe with correction**
Correction variable Sl_cor	- 1 °C/h (Sl_cor = Sl)	- 0.5 °C/h (Sl_cor ≠ Sl)
Expected final amount of the product Pf_cor	33.26 penicillin units	34.40 penicillin units

4.3 Conclusions

This chapter described the application of a recipe adaptation set to batch initialization and correction, for achieving the best possible processing performance. The examples illustrated the main ideas of both strategies and gave an impression about the expected profits.

For batch correction, it was assumed that only one sample could be taken and analyzed. Sometimes the practical situation is different: more than one sample is taken and more corrections can be done. In such a case, more than one pair of correction models is needed, namely one pair (*Model_C1*, *Model_C2*) for every sample/correction moment.

Chapter 6 describes more comprehensive examples of the proposed strategies, when applied to an industrial process producing alkyd resins and to a laboratory process producing epoxy resins. Further, batch initialization applied to the production of benzylalcohol in a pilot plant is described in [KEE93], and batch correction applied to a simple irreversible first-order $A \rightarrow B$ reaction is described in [STE92].

5. The FRIS-Package

The flexible recipe-approach, to be used in an R&D and production environment of a large variety of batch processing industry, has to be supported by the FRIS software package (the Flexible Recipe Improvement System). The implementation of the package is done by the TNO/TPD Institute of Applied Physics. This chapter describes my contribution to the software development, in particular the definition of the functional model and the software requirements. The determination of system structure, the choice of the environment for the implementation, and the definition of possible system developments in the future, resulting from team discussions, are also briefly outlined in this chapter.

5.1 The Functional Model

During problem analysis leading to software development, making a functional model of the software system turned out to be very useful. A functional model describes what the system has to do, without pointing out how and when it is done [MAR79]. It consists of a network of processing units, called processes or functions. Data-flow diagrams are used to provide the functional model with a graphical representation. The structure of a data-flow diagram is a directed graph with three kinds of nodes: processes, terminators and data stores. Nodes are connected by arrows representing an information flow from the source node to the destination node.

Usually, the following symbols are used [BRU95]:
- circle: describes a process which transforms data;
- rectangle: represents a terminator, which is either a sender
  or a receiver of data outside the system;
  together they indicate the environment of the system;
- pair of parallel horizontal segments: depicts a data store for later use,
- arrow with a name: describes type of data and the direction of flow.

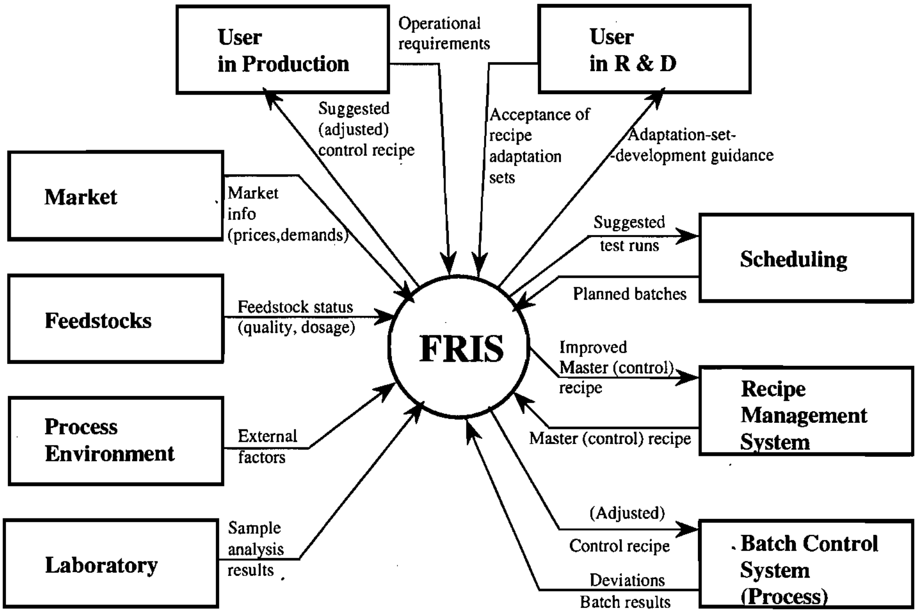


Figure 5.1 Data-context diagram of the FRIS package

The first data-flow diagram is called the context diagram. It shows the surroundings of the software, i.e. just one process, the system itself, and all the terminators and all environment data flows. The context diagram of the FRIS-package, among other things with the two types of users: in R&D and in Production, is presented in Figure 5.1. When the top layer, i.e. the context diagram, is split into sub-processes and next into sub-sub-processes and so on, a top-down specification of the software can be done. In Figure 5.2, which is the first sub-layer of the data-context diagram, the two processes, earlier called the two FRIS activity-domains (see Section 2.3.2), are presented, the first one: development of a recipe adaptation set and recipe improvement, and the second: run-time application of a recipe adaptation set. The refinement of the levelled data-flow diagrams can go on in successive "layers" until each process may be considered to be "atomic". How it is done for batch initialization, which is a sub-process of run-time applications of a recipe adaptation set, is described in [VERD95].

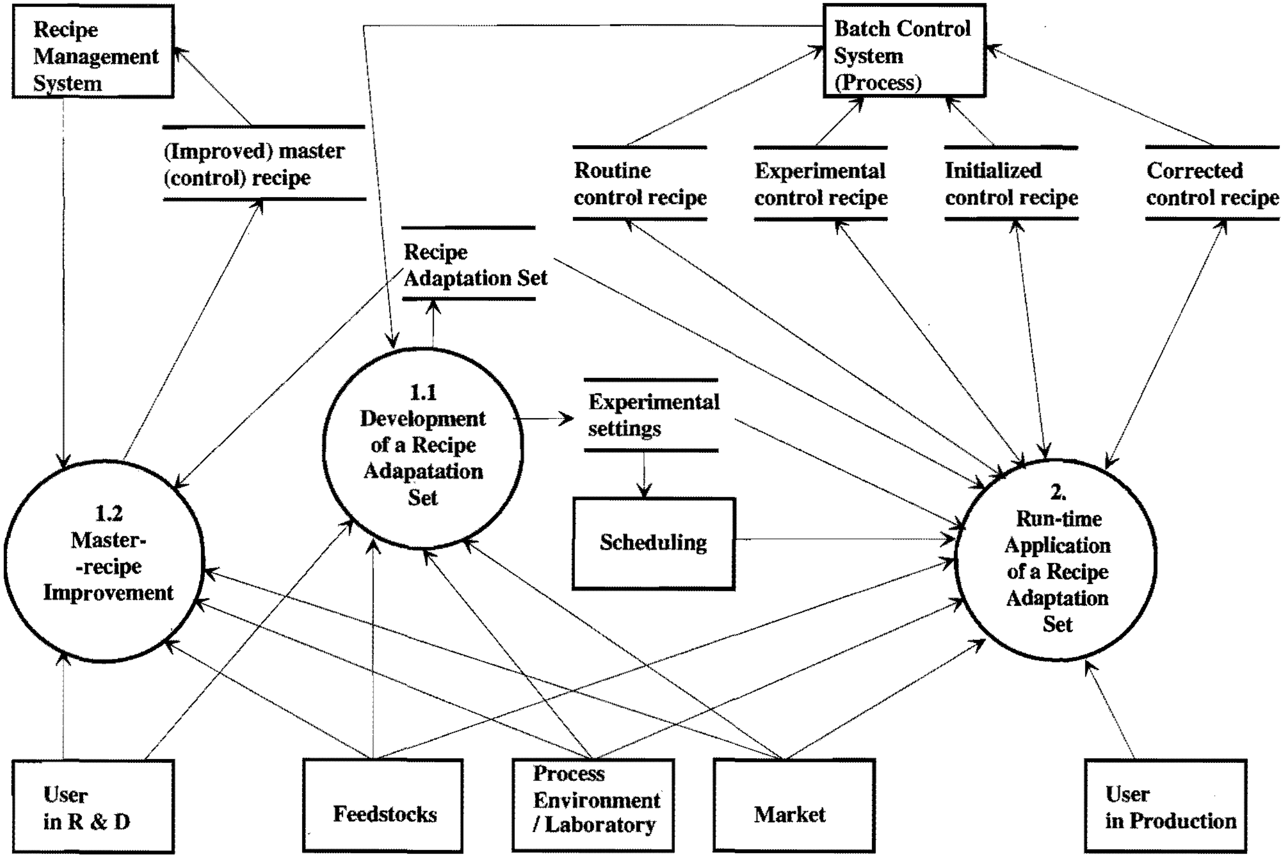

Recipe Management System
(Improved) master (control) recipe
Batch Control System (Process)
Routine control recipe
Experimental control recipe
Initialized control recipe
Corrected control recipe
Recipe Adaptation Set
1.1 Development of a Recipe Adapatation Set
Experimental settings
Scheduling
1.2 Master- -recipe Improvement
2. Run-time Application of a Recipe Adaptation Set
User in R & D
Feedstocks
Process Environment / Laboratory
Market
User in Production

Figure 5.2 The second layer of the flow diagram of the FRIS-package

5.2 Software Requirements

According to Jalote, the basic issues which software requirements must address are [JAL91]:

1) functionality;
2) system performance;
3) design constraints imposed on an implementation;
4) external interfaces.

The general requirements of the FRIS system can be summarized as follows:

1.1 the package must fully support the experiment design and the model development procedure;

1.2 the package must support recipe improvement, also when more than one objective are involved;

1.3 in the case of deviations, FRIS must provide an adjusted control recipe for batch initialization; it must also do so for batch correction based on run-time measurements, including sample analyses;

1.4 the search for an improved/adjusted recipe must take **all** specified constraints into account;

1.5 the package should support process monitoring;

2.1 the system must be sufficiently fast;

2.2 it must be easy in starting up a module and in modification of the input variables during its use;

3.1 the terms used by the system should not be conflicting with the ISA S88 standard;

4.1 FRIS must serve as an "open" system which allows communication with any statistical package preferred by the user;

4.2 the user interface must be suitable for users with experience in statistics, and for users without experience in statistics;

4.3 the presentation of results must be self-explanatory.

5.3 The Structure of the FRIS-Package

At present, the FRIS software-system, as presented in Figure 5.3, consists of the following parts [REN95]:

1) FRIS Monitor ("Main Program")
2) Recipe-adaptation-set Database (in short called "RECIPES")
3) Batch Planning/Scheduling and Results Database (in short called "BATCH DATA")

4) Recipe-adaptation-set Editor
5) Experiment Design, Modelling and Experimental Optimization
6) Model-based Recipe Improvement
7) Interface to external (statistical and other) packages
8) Batch Initialization
9) Batch Correction
10) Process Monitoring.

They are described below. It should be mentioned that the parts 5÷7 are collections of modules, which are embedded in one coherent structure.

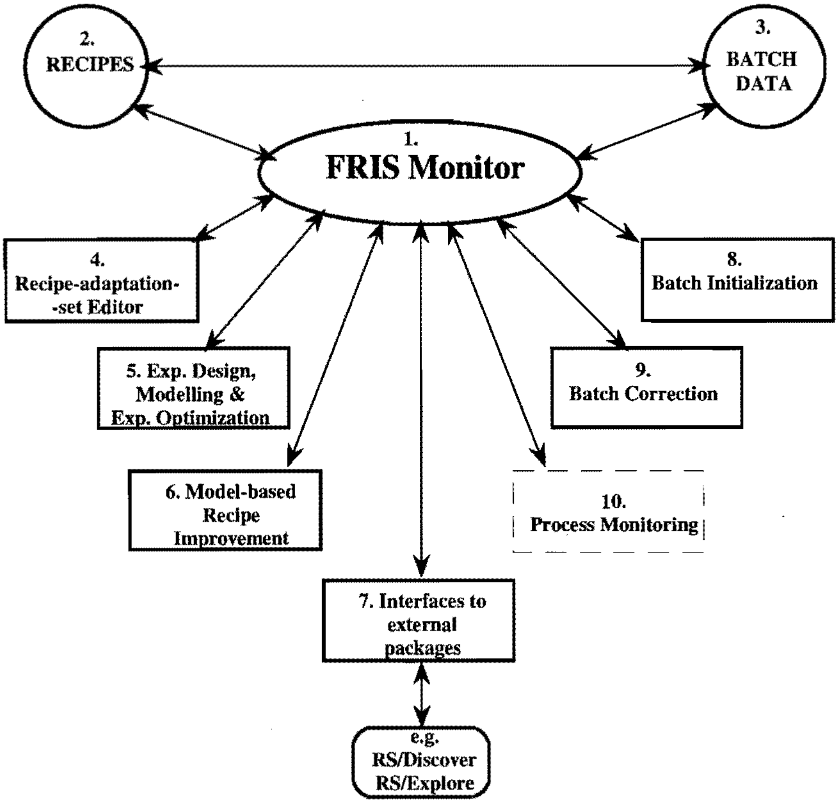


Figure 5.3 FRIS system functions. The modules 4÷7 correspond to the development activity-domain, and the modules 8÷10 to the application domain. The module for process monitoring must still be realized. The database "RECIPES" contains master (control) recipes and recipe adaptation sets, "BATCH DATA" contains routine, experimental, initialized, corrected and accomplished control recipes.

5.4 Principal FRIS Components

5.4.1 FRIS Monitor

The FRIS Monitor is the main component of the package that implements the central menu-based user interface, database management and module management.

5.4.2 Recipe-Adaptation-Set Database "RECIPES"

The total FRIS database is composed of two parts, namely: the Recipe-Adaptation-Set Database (RECIPES) and the Batch Planning/Scheduling and Results Database (BATCH DATA).

The former part contains a number of recipe-adaptation-set "objects".
Each recipe-adaptation-set object contains the following information:

0) a reference to the corresponding master recipe;
1) a list of recipe items and their properties;
2) a list of process outputs;
3) a list of relevant constraints;
4) at least one performance criterion;
5) values (often prices of ingredients and products) of variables used in performance criteria;
6) a nominal process model with its validity constraints;
7) a collection of auxiliary process models;
8) the correction information;
9) a description of the processing unit-configuration;
10) a list of **experiment worksheets**;
11) other information, e.g. notes.

The elements 1÷9 correspond to the components of a recipe adaptation set as defined in Section 2.3.2; the tenth element is added to link corresponding experiments with the recipe adaptation set.

An **experiment worksheet** is a complete list of planned, at that moment active or accomplished test runs corresponding either to model-based recipe improvement with one specific model or to experimental optimization. The planned and active test runs can also be found in BATCH DATA under the mode "EXPERIMENT" (see below).

It should be mentioned that RECIPES part of the database is the domain of the user in R&D, who develops a recipe adaptation set during the interactive session according to the procedure described in Chapter 3, and next uses it for the generation or improvement of a master (control) recipe.
Once defined, the RECIPES part of the FRIS database has 'read-only' status for the operational part of the package.

5.4.3 Batch Planning/Scheduling and Results Database "BATCH DATA"

This part of the FRIS database contains information about all individual batches, namely: batches which have been, are being, or have to be carried out. For each batch the following information is stored:

1) "mode" describing the kind of control recipe (see below);
2) "recipe": name of the corresponding recipe-adaptation-set object;
3) "recipe items": the actual recipe-item values used and the measured process outputs.

Batches can be carried out in one of four possible modes, depending on the kind of the corresponding control recipe. The names of these modes are:

- "ROUTINE": production run according to a routine control recipe; its results are important to the FRIS system for the comparison of the process performance and for model estimation, validation and/or updating;
- "EXPERIMENT": test run in the framework of an experiment design or experimental optimization according to an experimental control recipe;
- "INIT": production run according to an initialized control recipe;
- "CORRECT": production run according to a corrected control recipe.

In planning the mode of batches is either "ROUTINE" or "EXPERIMENT". It can be changed at the start of a batch in "INIT", or during processing in "CORRECT". When a batch is finished, the results are entered into BATCH DATA. The results of a batch labelled with "EXPERIMENT" are copied to the corresponding experiment worksheet and the batch is removed from the list of planned batches. If the user so wishes, he can also add results of batches with an other mode to the experiment worksheet.

5.4.4 Recipe-Adaptation-Set Editor

The Recipe-adaptation-set Editor is a module for the creation and modification of a recipe adaptation set. It offers facilities for specifying all parts of a recipe-adaptation-set object as defined in Section 5.4.2.

5.4.5 Experiment Design, Modelling and Experimental Optimization

Experiment design modules, e.g. factorial design modules, start with a recipe-adaptation-set object and generate a worksheet with planned test runs, which are next copied into BATCH DATA. The results of each completed batch are copied back to be used for process modelling or refinement/extension of the experiment design task.

Modelling modules use the data of a worksheet to fit a model, that explains the batch results as a function of the recipe item settings. If the fitted process model is accepted, then the FRIS Monitor stores it in the reçipe-adaptation-set object. If more than one worksheet is defined in the object, the FRIS system will ask for the name of the worksheet before beginning modelling.

To the group of **experimental optimization** belong the Nelder-Mead, Multiplex-Fitting and Line Optimization modules. They start with a flexible recipe object and an input from the user with additional information about stopping criteria and direction of search, with which the chosen module generates a worksheet with planned test runs. The main program inserts a copy of the worksheet into BATCH DATA, so that the test runs can taken care of the batch execution part of the FRIS package.

As soon as a batch is executed, its results are copied back into the worksheet and the module can compute new test-run settings. This procedure is repeated until the optimized settings are found. The found settings can next be used as nominal settings of a master control recipe but also as initial settings for an factorial design module intended for process modelling.

5.4.6 Model-Based Recipe Improvement

The model-based recipe improvement module, starting with a recipe-adaptation-set object, optimizes the performance criterion subject to model, process/plant and economical constraints.

The user is presented with an elaborate user interface for "playing" with the optimization process, such as fixing certain recipe items or adding/removing constraints. The progress of the criterion value can be monitored, e.g. by means of a plot of the criterion value versus the optimization run number or of the hilly landscape.

5.4.7 Interfaces to External (Statistical and Other) Packages

This modules give the user the possibility of using external packages, e.g packages for design of experiments and process modelling, or Batch Information & Management systems. For each external package, a separate software module must be written that converts data from the FRIS database to a format to be read by the external package, and vice versa.

Currently, an interface with the RS/Discover and RS/Explore packages is available. Design of experiments and process modelling are done in this RS-environment.

5.4.8 Batch Initialization

The batch initialization module presents a user interface, with which the user can specify known deviations in initial conditions, set additional operational limits on recipe items and process outputs, choose adjustable recipe items, process models and the desired performance criterion, and next accept or reject the suggested adjusted recipe item values. If the suggested values are accepted, then the main program copies them into the BATCH DATA labelled with the mode "INIT" and prepares the initialized control recipe. If they are not accepted, then the module can be started again with modified input information.

5.4.9 Batch Correction

Batch correction is to be used, one or more times, during the batch run. The corrected control recipe gives the adjusted recipe settings for the remainder of the processing time.
If the corrected recipe settings suggested by the module are accepted by the user, then the main program inserts them into BATCH DATA under the label "CORRECT", and the batch correction can be carried out according to the prepared corrected control recipe.

5.4.10 Process Monitoring

Process monitoring uses models to detect disturbances which may be cased by changes in feedstocks, fouling etc, and subsequently to improve recipes. This module must still be realized.

5.5 Technical Implementation

The prototype of the FRIS-package has been implemented as a Windows-NT application in the PRIMACS package for real-time data acquisition, intelligent data processing & analysis, process modelling and control design of continuous processes [LIN90, VERH95]. The FRIS modules are implemented in C++ making use of the standard PRIMACS tools for graphical presentation.

The FRIS package can be seen as the extension of PRIMACS to batch processes.

5.6 Desired Further System Developments

At present, the TNO/TPD Institute of Applied Physics defines with a number of industrial partners a project, in the framework of which the development of a commercial version of the FRIS package is to be realized. Various improvements and extensions of the actual FRIS implementation are important [RAD95]:

- further development of robust modules for experiment design, process modelling, batch initialization and correction;
- realization of the process-monitoring module;
- realization of a framework for adding modules for recipe generation, improvement and adjustment on the basis of white process models;
- linkage with Windows packages for use of the data in reports and spread sheets;
- interfacing with other data bases of the Open Data Base Connectivity Standard for SQL data bases;
- interfacing with Batch Information & Management (planning/scheduling/control) systems and DCS/SCADA-systems;
- adding an expert system for better support of users without experience or with little experience in statistical analysis.

Chapter 6. Case studies

The flexible recipe-approach described in this thesis has been successfully applied to two simulated and five real processes. In this chapter in details two cases will be discussed:

a) production of an alkyd resin (industrial application at Akzo Nobel).
b) production of an epoxy resin (in the laboratory of our University);

For both cases, after a short process description, the results of recipe improvement, batch initialization and correction will be presented.
Other industrial cases will be mentioned in Section 6.3.

6.1 Alkyd Resin Production

This application of the flexible recipe-approach was in the production of "Setal" (an alkyd resin) at Akzo Nobel Resins in Bergen op Zoom. Three problems were investigated:

1) process improvement by the use of new raw materials;
2) optimization of process duration;
3) process correction for deviations from the desired acid value - viscosity band.

All problems concern the same production process, which will be described in brief before the three problems are examined. Because this application contains confidential business information, some terms are replaced by blank codes, and all models are presented only in a scaled form without transformation rules.

6.1.1 Reaction Mechanism and Process Description for the Production of the Alkyd Resin Setal

The alkyd resin Setal is produced in a so-called fatty acid process, i.e. an esterification of a polybasic fatty acid, the molecules of which contain two or more carboxyl groups ($-COOH$), and a polyhydric alcohol, the molecules of which contain two or more hydroxyl groups ($-OH$) [PAT62]. The reaction between a carboxyl group of an acid and a hydroxyl group of an alcohol, which releases water molecules, is called esterification. Broadly speaking, this type of esterification produces compounds of the general class of polyesters.

The key feature that distinguishes the alkyd resins, or alkyds, from other polyesters is the presence of fatty acid as a major part of their composition.

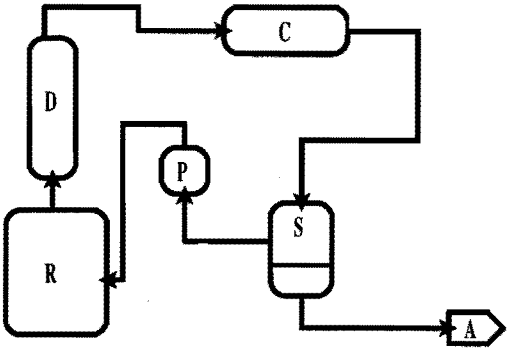


Figure.6.1 Schematic presentation of the X-cycle in the Setal process

The reaction mechanism is an equilibrium reaction. By withdrawing the water formed during esterification, the equilibrium "shifts to the right" in favour of the formation of alkyds. For this purpose, a volatile component X with azeotropic properties is used in an evaporation-distillation-reflux cycle, as shown in Figure 6.1: evaporation in the reactor (R) takes care of the removal of the X+water mixture from the reactor via the dephlegmator (D); next the X+water mixture is condensed (C) and after it settles in a separation tank (S), the X component is carried back into the reactor through the pump (P), whereas the water is drawn off (A).
After loading the raw materials into the reactor, the process is started by heating the vessel. During this step inert gas is led through the mixture to prevent the alkyd from becoming too dark.

The second step is called the **solvent process**, which is the actual reaction phase. Normally, no inert gas passes through the reactor during this process phase, in which the progress is followed by periodical checking of the acidity and viscosity of the alkyd. The acid number furnishes an indication of the proportion of the reaction completed, and the viscosity furnishes an indication of the size and complexity of the alkyd polymers that have been formed. It is common practice to plot the acid value versus the viscosity. Figure 6.2 shows a characteristic diagram with the ideal progress represented by the so-called "heart line" (HL) and the tolerance boundaries. Periodically, a sample is taken from the reactor mixture and after analysis its results can be marked by a point in the acid-viscosity diagram. If this point does not lie between the allowed boundaries, a correction is necessary: if the point lies above the allowed band (point H in the diagram), the correction is done with an alcohol, and if it is below (point L in the diagram) an acid is used. The reaction is terminated when the viscosity and the acid number lie in the final destination area, which is represented by a dotted rectangle in Figure 6.2. Next, after the mixture has been cooled, it is transported to a thinning tank to bring the product on the final specifications. Finally, the alkyd resin is filtered and pumped into storage tanks.

The critical phase in FRIS-terms is the solvent process, for which the three application problems mentioned in Section 6.1 will be examined.

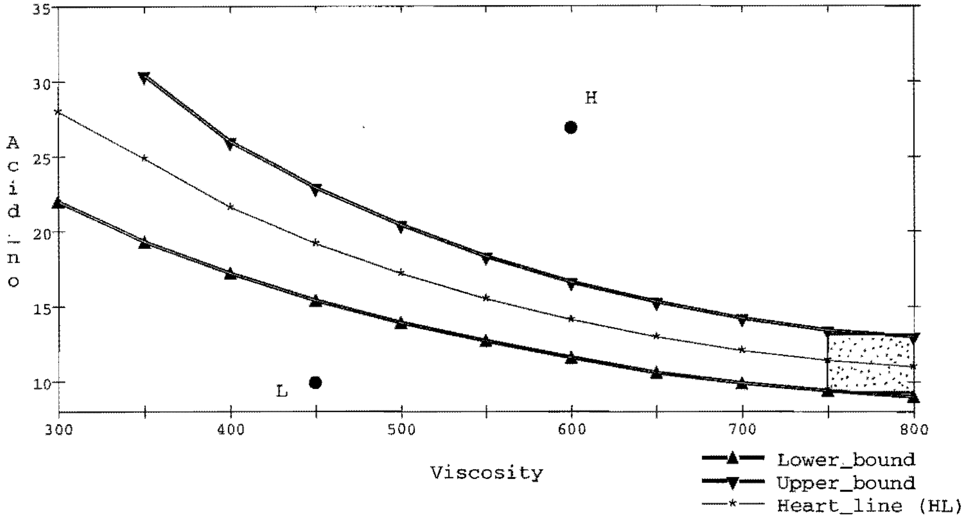


Figure 6.2 Acid - viscosity trajectory used for the monitoring of the Setal process

6.1.2 Process Improvement by the Use of New Raw Materials

This application concerns the improvement of the Setal process, in which the old recipe is replaced by a cheaper one: four ingredients of the old recipe are replaced by two new feedstocks.

The question is: how should the dosage of these two new ingredients be chosen so as to obtain the final specifications:

$$9 < AC\_end < 14 \tag{6.1}$$

$$750 < VIS\_end < 850; \tag{6.2}$$
the desired value of VIS_end is 800;

$$85 < OHv\_no < 120; \tag{6.3}$$
the desired value of OHv_no is 105;

in which:

AC_end	:	final value of the acid number;
VIS_end	:	final value of the viscosity;
OHv_no	:	hydroxyl number of the product.

The conditions (6.1) and (6.2), which must be satisfied, require that the final conditions of the solvent process lie inside the rectangle shown in Figure 6.2. Further, it is desirable that no correction takes place during the batch operation, in other words:

$$H\_cor = 0 \tag{6.4}$$
$$L\_cor = 0 \tag{6.5}$$

where:

H_cor : amount of correction with an alcohol in case the sample indicates that the process conditions are above the allowed acid-viscosity band;

L_cor : amount of correction with an acid in case the sample indicates that the process conditions are below the allowed acid-viscosity band.

The problem of finding an appropriate recipe for the new ingredients can be seen as an end-specifi-cation problem in accordance with definition (3.82) given in Section 3.6.3.2.

6.1.2.1 Still 30% of Batches Needs Corrections

After switching over to the new feedstocks, none of the batches progressed within the desired acid-viscosity band, i.e. the requirements (6.4) ÷ (6.5) were violated. To improve this situation, Akzo Nobel found a production point that was expected to result in the desired end specification. This was done with the help of the available process knowledge and 33 experimental test runs, which, however, had not been done in accordance with the methods from the field of "experiment design", as used in the FRIS-approach. Still, for about 30% of the batches produced under these new process conditions, corrections were needed to ensure that the process remained within the acid-viscosity bound.

6.1.2.2 Could the FRIS-Approach Lead to an Improvement of the Setal Process?

The aim of the FRIS application was to investigate whether the flexible recipe-approach could lead to an adequate improvement of the Setal process.
Because the experiments had already been done, it was impossible to follow the model-development procedure described in Section 3.2.2. Therefore, and because of the typical character of the end-specification problem, the following procedure was chosen:

1) become conversant with the available information and the objectives, and consider the need of additional experiments;
2) estimate the transition model for each response y_i:

$$y_i = f_i(\mathbf{x}), \quad i = 1 \div 5 \tag{6.6}$$

where:

$$\mathbf{y} = [AC\_end, \quad VIS\_end, \quad OHv\_no, \quad H\_cor, \quad L\_cor]; \tag{6.7}$$
$$\mathbf{x} = [x_1, \ x_2, \ x_3, \ x_4, \ x_5] \qquad \text{recipe items (ingredient dosages);}$$

Note that because the end product is not only affected by the two new ingredients but also by the three old ingredients, their dosages are also treated as recipe items. It should be mentioned that the total charge is constrained by the reactor capacity.
3) conduct statistical tests in order to judge the models;
4) investigate the models and decide upon their usefulness; if they are not satisfactory, then define additional experiments and do them, if possible, and return to 2;
5) determine the area for the recipe items $x_1 \div x_5$, from which the required end-specification of $\mathbf{y}$ can be reached without corrections, and develop one or more intelligent methods for investigation of that area;
6) investigate the scope of and the need for defining one or more performance criteria; search for the best recipe for each criterium, or define and solve a multi-objective problem.

Ad 1). *Investigation of the available information*

After the examination of the available data the following observations were made:

- The five recipe items were varied during the 33 experimental test runs as follows:
  x_1, x_2 and x_3 were varied at five levels, x_4 and x_5 at two levels. At the end of the process, VIS_end and AC_end, and in most cases also OHv_no, were measured. In all cases the corrections, if carried out, were recorded too.
- The variations occurring in x_4 were correlated with those in x_5: an increase in x_4 was always coupled with an increase in x_5, so that one of these, it does no matter which one, had to be left out of consideration. It was chosen to proceed with x_4.
- The hydroxyl number OHv_no was unknown for nine batches. Upon a suggestion by Akzo, it was decided to ignore this response, also because the OHv_no value is corrected in the final product thinning tank.
- During the experiments there were no batches with too low an acid number, so L_cor was always zero.
- The test runs were not performed according to any customary design of experiments scheme, therefore it had to be expected that the conditions for model estimation are less favourable.
- There turned out to be no possibilities for further experimentation, so that the recipe improvement had to be done on the basis of the available information.

Ad 2). *Model estimation*

From the five transition models expected by (6.6) only three models remain, namely:

- MOD_AC: $\text{AC\_end} = f_1(x_1, x_2, x_3, x_4)$ (6.8)
- MOD_VIS: $\text{VIS\_end} = f_2(x_1, x_2, x_3, x_4)$ (6.9)
- MOD_H_COR: $\text{H\_cor} = f_3(x_1, x_2, x_3, x_4).$ (6.10)

To estimate them sensitivity analysis was done: first, a correlation matrix was built for each response. This matrix was then used to choose the recipe items with the strongest linear influence upon the response. After that, a second-order model in terms of the selected items was estimated for each response. Next, it was tried to determine whether the addition of the items initially left out, and any combination with other items, improves the model. As a criterion for model judgement the coefficients R^2, R^2_{adj} and R^2_{PRESS} defined in

Section 3.2.2.6 were used. The model with the highest values of these coefficients was taken into consideration and subsequently reduced by removing any non-significant terms. This resulted in the following models:

$$\begin{aligned} \text{AC\_end} = \; & 32.227\cdot(\sim x_1) + 31.701\cdot(\sim x_2) \\ & + 32.080\cdot(\sim x_1\cdot\sim x_2) + 2.015\cdot(\sim x_2\cdot\sim x_3) \\ & - 1.381\cdot(\sim x_3^2) + 42.961\cdot(\sim x_4^2) \end{aligned} \tag{6.11}$$

$$\begin{aligned} \text{VIS\_end} = \; & 3484.61\cdot(\sim x_1) + 3542.05\cdot(\sim x_2) \\ & + 6293.80\cdot(\sim x_1\cdot\sim x_2) + 1213.51\cdot(\sim x_1\cdot\sim x_3) \\ & + 1220.381\cdot(\sim x_2\cdot\sim x_3) + 7073.64\cdot(\sim x_1^2) \end{aligned} \tag{6.12}$$

$$\begin{aligned} \text{H\_cor} = \; & - 885.81\cdot(\sim x_2) + 1948.64\cdot(\sim x_3) \\ & + 1431.14\cdot(\sim x_1\cdot\sim x_3) + 1434.92\cdot(\sim x_1\cdot\sim x_4) \\ & + 2602.39\cdot(\sim x_2\cdot\sim x_3) - 2476.57\cdot(\sim x_2\cdot\sim x_4) \\ & - 5111.09\cdot(\sim x_2^2) + 1125.54\cdot(\sim x_3^2) \end{aligned} \tag{6.13}$$

The corresponding Tables 6.1 ÷ 6.3 with the statistical information show that the values of the coefficients R^2, R^2_{adj} and R^2_{PRESS} are higher than 0.8, which in first instance is an indication of good fit. That the condition numbers are quite high indicates some collinearity among recipe items in each of these three models.

Table 6.1 Model of the response AC_end

```
Least Squares Coefficients, Response AC_end, Model MOD_AC

Term          Coeff.          Std. Error    T-value    Signif.
-----------------------------------------------------------------
1 ~x1         32.227440       7.378336       4.37      0.0004
2 ~x2         31.701115       7.299435       4.34      0.0004
3 ~x1*x2      32.080279       7.232632       4.44      0.0003
4 ~x2*x3       2.014662       0.318834       6.32      0.0001
5 ~x3**2      -1.380763       0.474617      -2.91      0.0094
6 ~x4**2      42.960733       7.172196       5.99      0.0001

No. cases = 24        R-sq. =   0.9986      RMS Error = 0.4967
Resid. df = 18    R-sq-adj. =   0.9982      Cond. No. = 212.7
~ indicates factors are transformed.

                R-sq-PRESS =   0.997
```

Table 6.2 Model of the response VIS_end

```
Least Squares Coefficients, Response VIS_end, Model MOD_VIS

Term            Coeff.              Std. Error       T-value    Signif.
--------------------------------------------------------------------------
1 ~x1           3484.613019          1015.175490        3.43      0.0056
2 ~x2           3542.045108          1015.664207        3.49      0.0051
3 ~x1*x2        6293.796896          1227.758987        5.13      0.0003
4 ~x1*x3        1213.508375           563.103665        2.16      0.0542
5 ~x2*x3        1220.460837           569.617962        2.14      0.0554
6 ~x1**2        7073.639850          1225.851233        5.77      0.0001

No. cases = 17          R-sq. =   0.9986        RMS Error = 37.31
Resid. df = 11     R-sq-adj. =   0.9978        Cond. No. = 374.1
~ indicates factors are transformed.

                   R-sq-PRESS =   0.997
```

Table 6.3 Model of the response H_cor

```
Least Squares Coefficients, Response H_cor, Model MOD_H_COR

Term            Coeff.              Std. Error       T-value    Signif.
--------------------------------------------------------------------------
1 ~x2            -885.804834          246.705002       -3.59      0.0017
1 ~x3            1948.643247          457.917819        4.26      0.0004
3 ~x1*x3         1431.139501          287.290963        4.98      0.0001
4 ~x1*x4         1434.920795          287.225995        5.00      0.0001
5 ~x2*x3         2602.388494          638.423855        4.08      0.0005
6 ~x2*x4        -2476.565672          636.526172       -3.89      0.0008
7 ~x2**2        -5111.085652         1210.555409       -4.22      0.0004
8 ~x3**2         1125.543292          156.116938        7.21      0.0001

No. cases = 29          R-sq. =   0.9118        RMS Error = 31.23
Resid. df = 21     R-sq-adj. =   0.8782        Cond. No. = 432.6
~ indicates factors are transformed.

                   R-sq-PRESS =   0.816
```

Ad 3). *Conduction of statistical tests in order to judge the models*

Tables 6.4 ÷ 6.6 present the analysis of variance of the three models. In no case a lack of fit has to be concluded.

Table 6.4 Analysis of variance for the model of the response AC_end

```
Least Squares Summary ANOVA, Response AC_end Model MOD_AC

Source             df    Sum Sq.     Mean Sq.    F-Ratio    Signif.
------------------------------------------------------------------------
1 Total            24    3210.990
2  Regression       6    3206.548     534.425   2166.00    0.0000
3   Linear          2       4.889       2.445      9.91    0.0013
4   Non-linear      4     755.951     188.988    765.90    0.0000
5  Residual        18       4.442       0.247
6   Lack of fit     5       1.099       0.220      0.86    0.5357
7   Pure error     13       3.342       0.257

F(5,13) as large as 0.8551 is not a rare event =>
    no evidence of lack of fit.
```

Table 6.5 Analysis of variance for the model of the response VIS_end

```
Least Squares Summary ANOVA, Response VIS_end Model MOD_VIS

Source             df    Sum Sq.     Mean Sq.    F-Ratio    Signif.
------------------------------------------------------------------------
1 Total            17    10978825
2  Regression       6    10963516     1827253   1313.00    0.0000
3   Linear          2       49771       24885     17.88    0.0003
4   Non-linear      4     8155744     2038936   1465.00    0.0000
5  Residual        11       15309        1392
6   Lack of fit     1        1200        1200      0.85    0.3781
7   Pure error     10       14109        1411

F(1,10) as large as 0.8506 is not a rare event =>
    no evidence of lack of fit.
```

Table 6.6 Analysis of variance for the model of the response H_cor

```
 Least Squares Summary ANOVA, Response H_cor Model MOD_H_COR

Source             df    Sum Sq.     Mean Sq.     F-Ratio  Signif.
------------------------------------------------------------------------
1 Total            29    232186.0
2  Regression       8    211706.5     26463.3     27.14    0.0000
3   Linear          2    133405.8     66702.9     68.40    0.0000
4   Non-linear      6    211629.1     35271.5     36.17    0.0000
5  Residual        21     20479.5       975.2
6   Lack of fit     5      3688.3       737.7      0.70    0.6294
7   Pure error     16     16791.2      1049.4

F(5,16) as large as 0.7029 is not a rare event =>
    no evidence of lack of fit.
```

When one takes a look at the probability plots, like Figure 6.3, or at the histograms of the estimated responses, like Figure 6.4, one may conclude that the residues do not appear to be normally distributed. This can mean that the presumptions of the least square estimation method are not fulfilled. To be sure, the goodness-of-fit test with χ^2 statistic was done: there is no reason to reject the null hypothesis about the normality presumption, so that it was decided to tentatively accept the models.

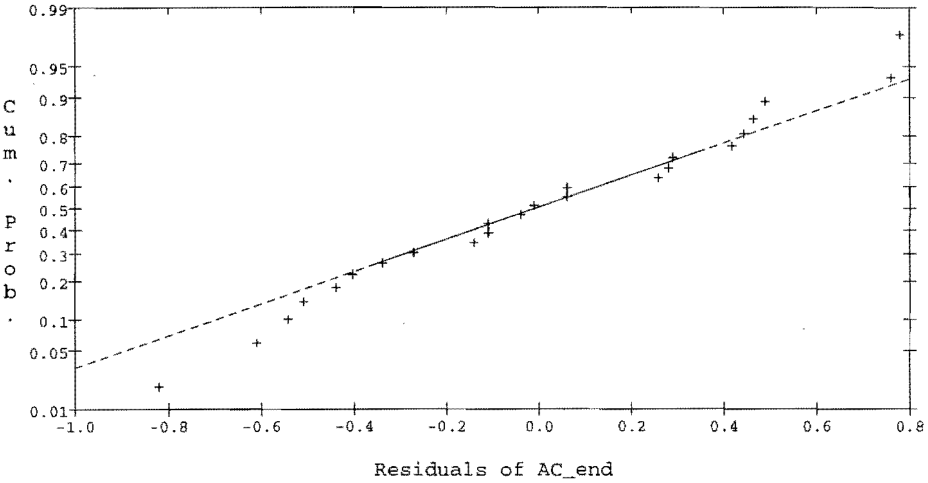


Figure 6.3 Probability plot for the process model of the response AC_end

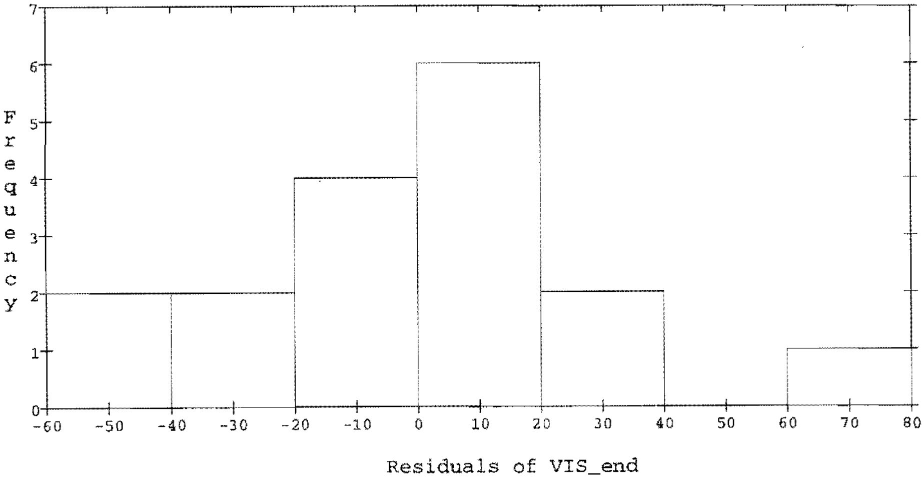


Figure 6.4 Histogram for the process model of the response VIS_end

Ad 4). *Investigation of the process models*

The visualisation of the modelled responses was found to be very helpful in the investigation of the models. If one looks at Figure 6.5, which presents the contour plots of the estimated responses as a function of x_1 and x_2, with the other recipe items fixed at a certain level, one can see that the experiments, presented as "*", were not very well balanced: they were done for recipe settings, which differ very little from each other. The contour plots for other recipe items fixed at another level show the same badly balanced distribution of the experiments. Since validation experiments and new test runs were impossible, it was decided to use the models 6.11 ÷ 6.13 in the search for an improved recipe.

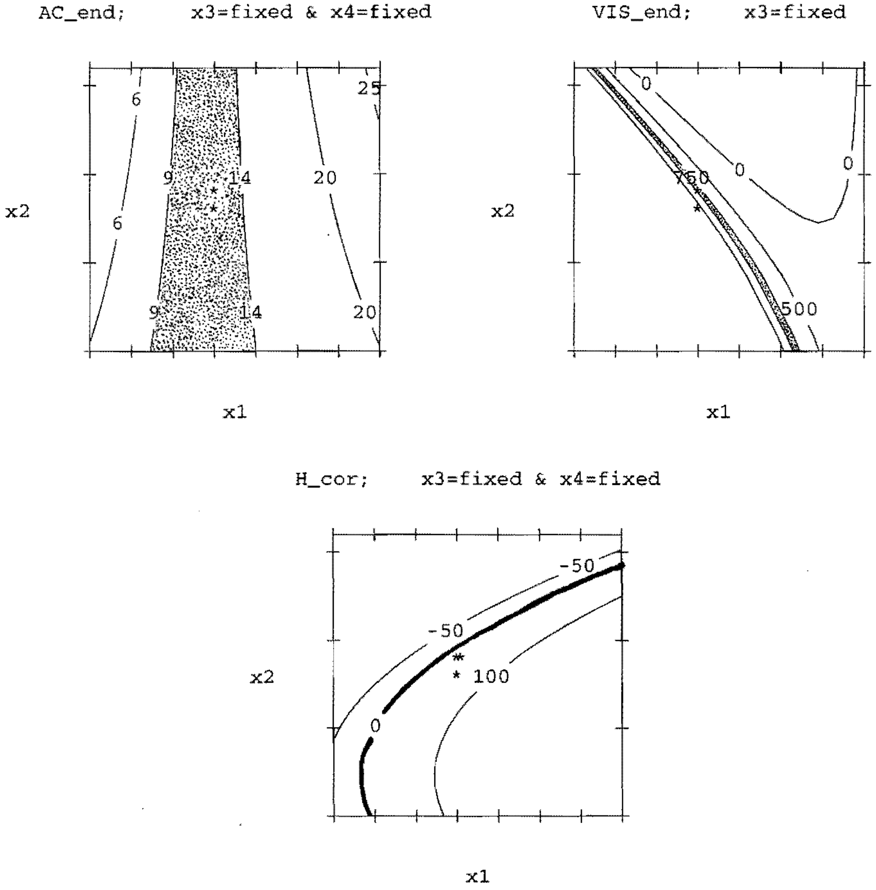


Figure 6.5 Contour plots of the three estimated responses with the allowable area's (shaded)

Ad 5). *Determination of the allowable area X^{spec} for the recipe items*

Contrary to the method for recipe improvement described in Chapter 3, the circumstances have led us to consider a new view of the problem, namely with which beginning dosage the desired end specification can be reached without any corrections.

If the desired end specification is to be reached, the allowable area X^{spec} for the selected four recipe items is described as follows:

$$X^{spec} = \{(x_1, x_2, x_3, x_4):\ 9 < AC\_end < 14 \wedge 750 < VIS\_end < 850 \wedge H\_cor = 0\ \} \qquad (6.14)$$

It looks nice, but what can you do with this? For practical applications such a four-dimensional area may be somewhat awkward. In order to arrive at useful results the four possibilities were considered.

The first possibility was to give an answer to the question whether a particular combination of the recipe items would lead to the required end conditions. If the models are known, a simple computation can answer this question with "yes" or "no". However, this method is not satisfactory enough because of its limited usefulness. If the answer is "no", the method gives no indication of the direction in which to search for a good recipe.

The second possibility was to make a set of contour plots of the modelled responses in one figure, like Figure 6.5, which may give a good impression about this allowable area. In this case, it is like a "snake" in a four-dimensional space. This method can be used as a tool to define a good search direction. Moreover, it gives some insight in the process.

The other possibility is to trace whether an important criterion can be formulated, which involves, for example, the most expensive ingredient, the minimal cost or the highest added value. If such criterion is defined, the optimization results in such a combination of recipe item settings, that it not only realizes the desired end specification, but moreover does so in an optimal fashion. This possibility will be discussed below.

The fourth possiblity, the finest solution of the dilemma of how to obtain a recipe in the four-dimensional space, lies in the problem of complicating the previous choice of one criterion. In practice, it is difficult, if not impossible, to choose one criterion, especially in situations when there are more requirements than just the degree of freedom. If there is more than one criterion in play, one of the current multi-objective methods described in Section 3.6 can be applied. However, according to our experience, these methods as far from being perfect. Therefore, an other method was developed: the Triplet-choice Multi-Objective Method, already described in Section 3.6.3.2, for the generation of the permitted combinations of the recipe items, which realize the required final specification. The user has to choose at each iteration step, from a set of allowed recipes the best, the worst and the second worst one. The procedure converges to the best recipe realizing the specification requirements. It must be mentioned that at each iteration step, additional responses can be added to or removed from the analyzed set of responses. Such an extension was also done at Akzo: i.e. hydroxyl number, left out of consideration during modelling, oil-length and average molecular weight of the alkyd resin were easily analyzed together with the three responses, which are modelled. Although there were no possibilities for further experimentation, only one iteration step according to the method showed its successfulness.

The same method can easily be used not only for end-specification, but also for solving multi-objective problems involving more than one separate objectives. Because of its particular usefulness in the industrial environment this working method has attracted special attention from Akzo Nobel.

Ad. 6) *Definition of the performance criterion and searching for the best recipe*

To investigate how a definition of a performance criterion can help to arrive at the best recipe, a criterion yielding a recipe ensuring the cheapest feedstock cost was formulated. Akzo Nobel was interested in the minimization of the feedstock cost per kg end product. Every time the prices are changed, the FRIS-approach makes it possible to compute a new recipe adjusted to the changed prices. To be more specific, we have compared the real prices in May and in November, which the concern was facing. It has resulted in the conclusion that the cheapest recipe in May is not the most economical one for November and that both recipes are different from the process conditions found by Akzo [KOO95].

Table 6.7 shows the percentage of profit increasing which can be reached by applying the FRIS-approach. It is 100 % for the most profitable recipe in May and in November, respectively. Also the most unprofitable recipes in both months were studied by maximization of the criterion. In this manner the possible range of improvement is computed: it is 0% for the most unprofitable recipe, 94 % for the actual Akzo recipe for the May prices, and 62% for the November prices. Thus, the actual Akzo recipe, is not optimal according to prices in May as well as in November.

Table 6.7 Percentage of profit increasing, computed for the feedstock cost per kg end product, for recipe settings according to the actual, best and worst recipes in May and November, respectively.

Settings according to the:	Percentage of profit increasing in May [%]	Percentage of profit increasing in November [%]
actual Akzo recipe	94	62
recipe, which minimizes the criterion in May	100	95
recipe, which minimizes the criterion in November	99	100
recipe, which maximizes the criterion in May	0	27
recipe, which maximizes criterion in November	17	0

6.1.3 Optimization of Process Duration

Another application of the flexible-recipe approach was to investigate whether it was possible to speed-up the solvent process. The manner in which three selected recipe items (other than in the previous case), say v_1, v_2 and v_3, influence the reaction time was examined. For this purpose, these three recipe items were varied as presented in Figure 6.6, with respect to their nominal values denoted with a bolder point. The numbers near the points denote how many test runs have been made, and the numbers in parentheses give the numbers of runs useful for further analysis. As the remaining 14 batches were not taken into consideration because of abnormal process situations, only 26 of the 40 runs could be used for model estimation.

The modelled response, the duration of the solvent process, denoted as TIME_SOLV, was computed for each batch as the time after the heating had finished and until the cooling begun. On the basis of the available 26 test runs the estimated model for the duration of the solvent process is presented in Table 6.8.

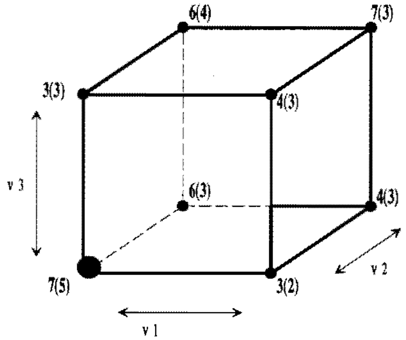


Figure 6.6 Schematic presentation of the experiments according to 2^3 factorial design aimed at the reaction time reduction.

The modelled response, the duration of the solvent process, denoted as TIME_SOLV, was defined for each batch as the time that elapsed between the moment that the heating had finished and the cooling begun. On the basis of the available 26 test runs, the estimated model for the duration of the solvent process is presented in equation (6.15) and in Table 6.8.

$$\begin{aligned}\text{TIME\_SOLV} = 3.622 - 0.213\cdot(\sim v_1) - 0.377\cdot(\sim v_2) - 0.202\cdot(\sim v_3)\\ - 0.197\cdot(\sim v_2 \cdot \sim v_3)\end{aligned} \tag{6.15}$$

Table 6.8 Model of the response TIME_SOLV

```
Least Squares Coefficients, Response TIME_SOLV, Model MOD_TIME_SOLV

Term            Coeff.          Std. Error    T-value    Signif.
----------------------------------------------------------------------
1 1             3.621770       0.096306
2 ~v1          -0.213304       0.099757       -2.14      0.0444
3 ~v2          -0.377169       0.096171
4 ~v3          -0.201790       0.097005
5 ~v2*v3       -0.197002       0.097407       -2.02      0.0560

No. cases = 26          R-sq. =  0.5805      RMS Error = 0.4805
Resid. df = 21      R-sq-adj. =  0.5006       Cond. No. = 1.332
~ indicates factors are transformed.

                  R-sq-PRESS  =  0.359
```

Obviously, this model has quite a small explanatory ability (with R^2 = 0.5805 and R^2_{adj} = 0.5006) as well as a very small predictive ability (R^2_{PRESS} is only 0.359). Therefore, this model is not acceptable. However, upon further reflection, it becomes clear, that the modelled response was not properly chosen. The solvent process was finished when the viscosity and the acid number were expected to lie in the desired rectangle of Figure 6.2. This means that the end moments of the solvent step were subjected to non-negligible fluctuations. As a consequence, another, more consistent, response was preferred: the time between the start time of the solvent phase and the moment the acid number reaches approximately the value of 12.

This new modelled response is denoted by TIME_AC12. Equation (6.16) and Table 6.9 show the estimated model of this response. It is obvious that the explanatory and predictive ability of this model is improved considerably in comparison to the model MOD_TIME_SOLV. This, together with no evidence of lack of fit, as presented in Table 6.10, lays the foundation of tentative acceptation of the new model MOD_TIME_AC12.

$$\begin{aligned}\text{TIME\_AC12} = 3.259 &- 0.123\cdot(\sim v_1) - 0.521\cdot(\sim v_2) - 0.280\cdot(\sim v_3) \\ &+ 0.208\cdot(\sim v_1 \cdot \sim v_2) + 0.135\cdot(\sim v_1 \cdot \sim v_3)\end{aligned} \qquad (6.16)$$

Table 6.9 Model of the response TIME_AC12

```
Least Squares Coefficients, Response TIME_AC12, Model MOD_TIME_AC12

Term            Coeff.           Std. Error    T-value    Signif.
-----------------------------------------------------------------------
1 1             3.258728        0.070866
2 ~v1          -0.123158        0.072274
3 ~v2          -0.520802        0.071278
4 ~v3          -0.279912        0.072340
5 ~v1*v2        0.208417        0.072483        2.88      0.0094
6 ~v1*v3        0.135023        0.073792        1.83      0.0822

No. cases = 26          R-sq. =   0.8433      RMS Error = 0.3487
Resid. df = 20    R-sq-adj. =   0.8041         Cond. No. = 1.5
~ indicates factors are transformed.

               R-sq-PRESS   =   0.739
```

Table 6.10 Analysis of variance for the process model of the response TIME_AC12

```
Least Squares Summary ANOVA, Response AC_end Model MOD_AC

Source              df   Sum Sq.     Mean Sq.    F-Ratio   Signif.
-----------------------------------------------------------------------
1 Total(Corr.)      25   15.51696
2  Regression        5   13.08501     2.61700     21.52    0.0000
3   Linear           3   11.72256     3.90752     32.13    0.0000
4   Non-linear       2    1.36244     0.68122      5.60    0.0117
5  Residual         20    2.43195     0.12160
6   Lack of fit      4    0.33164     0.08291      0.63    0.6471
7   Pure error      16    2.10032     0.13127

Model obeys hierarchy. The sum of squares for linear terms
is computed assuming nonlinear terms are first removed.
F(4,16) as large as 0.6316 is not a rare event =>
    no evidence of lack of fit.
```

To validate the model MOD_TIME_AC12, three test runs were planned in the centre of the factorial design in Figure 6.5. Unfortunately, one of them could not be used for validation because of an unforeseen, and undesirable correction, which prolonged the processing time. The results of the remaining two valid experiments are presented in Table 6.11. For comparison, the expected values and the confidence intervals are also presented for the rejected model MOD_TIME_SOLV. The results of both validation test-runs lie in the 95% confidence interval of the model MOD_TIME_AC12, whereas one result does so for the model MOD_TIME_SOLV.

Table 6.11 Validation test runs and the predicted values of the responses TIME_SOLV and TIME_AC12 in the centre point of the design according to the models MOD_TIME_SOLV and MOD_TIME_AC12

	TIME_SOLV	TIME_AC12
Batch V1	3.50	3.00
Batch V2	2.83	2.99
Lower pred. 95%	3.269	2.979
Predicted value	3.621	3.259
Upper pred. 95%	3.975	3.539

On the basis of these results, the model MOD_TIME_AC12 was accepted. The minimization of the modeled response TIME_AC12 results in a rather surprising minimum duration of 2.24 hours, at the boundary of the factorial cube of Figure 6.6, namely the point 4(3) at the bottom level. That means: shortening the solvent process duration by about 44%! However, such a shortening of the solvent step is, unfortunately, not really acceptable in practice, because in case a correction is needed, there would be not enough time for it to have effect. However, if one decides to aim at a reduction of the reaction time to only 3 hours, the model helps to find the appropriate process conditions, which may be done by solution of the following optimization problem:

$$\min_{\mathbf{v}} J(\mathbf{v}) = \min_{\mathbf{v}} v_1, \tag{6.17}$$

subject to:

$$\text{TIME\_AC12}(\mathbf{v}) = 3.0, \tag{6.18}$$

which means the minimization of the most expensive recipe item v_1 under the desired condition (6.18). Next, if the process turns out to be better under control, in the sense that corrections are unnecessary, the value in (6.18) may be replaced by a new one with shorter processing time.
In this manner the FRIS-approach can be used to generate an optimal master control recipe.

6.1.4 Process Correction for Deviations from the Desired Acid Value - Viscosity Band

This section shows what contribution the flexible recipe approach can make to the correction rules in cases in which the solvent phase does not progress inside the desired acid-viscosity band.

Although, as described in Section 6.1.2, a recipe was found which should achieve the end-specification requirements without any interim correction, about 30% of the batches still needed corrections. This was a clear-cut improvement with respect to the situation just after switching over to new ingredients, in which all batches did not progress inside the allowed acid-viscosity band. At present, however, there still are batches processing in an undesirable manner, and, moreover, if a correction is carried out, often (in more than 50% cases) it does not lead to the desired result. As a consequence, a second and sometimes a third correction is necessary.

6.1.4.1 The Correction Rule Used in the Factory

The progress of the reaction phase is tested by periodically taking a sample from the reactor and, if considered necessary, adding a corrective substance. Figure 6.7 presents the time scheme for taking a sample and making corrections. The first sample is taken about half an hour after the beginning of the solvent process, the analysis takes about 20 minutes, and the rule is that after half an hour after the first sample and just after the second sample, a correction can be made. The third sample, taken another half an hour later, shows the effect of the correction. If it is not sufficient, a second correction is carried out. The size of the correction does not depend on the correction moment, but only on the size of the measured deviation in the acid number with respect to the heart line: for 1 point of deviation on the acid number scale, 1 unit of a corrective substance must be added. In this manner, the process is monitored and corrected during the whole phase, but in the last hour no correction is allowed. In practice, the correction rule is more a guideline than a norm, and the operators, depending among other things upon their own experience, administer less or more than prescribed, so that an addition of 0.51 or 1.22 unity per 1 point deviation on the acid number scale is not uncommon. What is striking furthermore is that the same correction rule is applied irrespective of the position of the heart line and the desired acid value - viscosity area; the specification band as a whole shifted upwards, later the upper bound shifted downwards, narrowing the band.

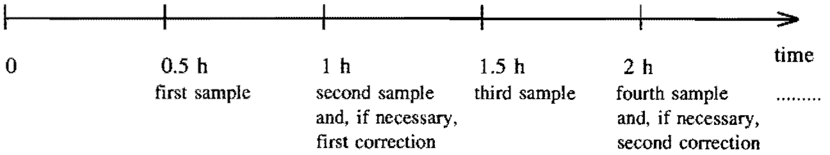


Figure 6.7 Time scheme for taking a sample and performing corrections

6.1.4.2 The Procedure Followed with FRIS

As the process data at my disposal did not provide information about corrections in case the acid number is lower than expected, the analysis was restricted to cases of too high an acid number, the correction being done with an alcohol.

Although the process is sampled periodically, it was found impossible to estimate other models than transition models, for example ARMA- or impulse-response models. The dynamics apparently play a part in a time interval shorter than half an hour and therefore could not be identified. Hence, continuous control, for keeping the reaction as closely as possible to the heart line, may be very effective. Therefore, it is interesting to examine what solution the flexible recipe-approach can offer.

To estimate the correction models, information about 22 corrected batches, i.e. batches with corrections, were at our disposal. The corrections were made at various sample moments. Therefore there were not enough data to estimate models like (6.19) and (6.20) for each sample moment separately, which was our primary aim:

$$ac_{n+1} = f_{a,n}(ac_{n-1}, vis_{n-1}, corr_n) \tag{6.19}$$
$$vis_{n+1} = f_{v,n}(ac_{n-1}, vis_{n-1}, corr_n) \tag{6.20}$$

where:

n : number of sample moment, n=1,2,3,...
ac_n : acid number measured at the n^th sample moment;
vis_n : measured viscosity corresponding with the n^th sample moment;
$corr_n$: correction at the n^th sample moment on the basis of the measured conditions at the n-1^th moment;
$f_{a,n}$: function describing the correction model for the acid number at the n^th moment, for example a second-order polynomial in the variables in parentheses;
$f_{v,n}$: function describing the correction model for the viscosity at the n^th moment, for example a second-order polynomial in the variables in parentheses.

An idea to answer the question, what insight into the correction procedure can be given by the flexible-recipe approach, was given by the standard Akzo correction rule: for 1 point of deviation on the acid number scale, 1 unit of alcohol must be added independently of the correction moment.

The solution was dealt with in the following way:

1) as reference for all batches the current specification band and the current heart line was taken;
2) at the moment that a deviation in the solvent process was detected, the following data were recorded (see Figure 6.8):

a) Delta_AC (D_A) — the difference in the acid number with respect to the heart line (measured vertically) at the sample moment when the conditions are outside the band;

b) Delta_VIS (D_V) — the difference in the viscosity with respect to the heart line (measured horizontally) at the sample moment when the conditions are outside the band;

c) Corr_alc (COR) — added amount of the corrective alcohol half an hour later than D_A and D_V are measured;

d) New_Delta_AC (N_DA) — the new difference in the acid number with respect to the heart line, half an hour after the correction is done, thus one hour after D_A is measured;

e) New_Delta_VIS (N_DV) — the new difference in the viscosity with respect to the heart line half, an hour after the correction is done, thus one hour after D_V is measured;

3) for a number of batches without corrections the same data were kept as in 2); here COR = 0 for all these batches. Provision had been made to assure that approximately as many data without correction as with corrections is available;

4) on the basis of the collected data, as described in 2) and 3), the following two transition models were estimated:

$$\text{N\_DA} = f_{\text{A}}\,(\text{D\_A, D\_V, COR}) \tag{6.21}$$
$$\text{N\_DV} = f_{\text{V}}\,(\text{D\_A, D\_V, COR}) \tag{6.22}$$

where:

f_{A}, f_{V} : model functions, in first instance second order polynomials in D_A, D_V and COR;

5) the estimated models were used for the generation of new correction rules.

With the definition of the model structure for f_{A} and f_{V}, the starting point was that the place with respect to the acid-viscosity band (in other words: the moment of the correction) also has an effect on N_DA and N_DV. Due to the non-linear shape of the heart line, D_A and D_V uniquely provide the position of a point in the acid-viscosity plane. In this manner one can more or less take into account the moment of sample taking.

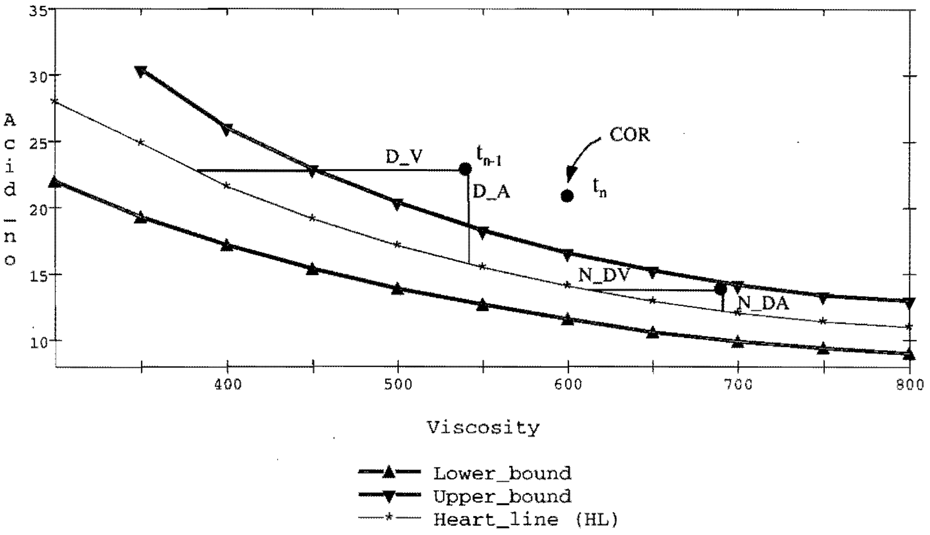


Figure 6.8 Acid - viscosity band with the indication of the modelled variables

6.1.4.3 The Correction Models

On the basis of 66 data sets from batches with corrections and without them, models as presented in (6.21) and (6.22) were estimated. The starting structure for both models was a second order polynomial in D_A, D_V and COR. After the non-significant terms were removed, linear models without a constant remained in both cases.

Equation (6.23) and Table 6.12 show the tentatively accepted correction model, called MOD_COR_AC, of the new difference in the acid number N_DA.

$$N\_DA = 0.792 \cdot D\_A - 0.212 \cdot COR \tag{6.23}$$

Table 6.12 Correction model of the response N_DA

```
Bisquare Coefficients, Response N_DA, Model MOD_COR_AC

Term            Coeff.        Std. Error    T-value    Signif.
------------------------------------------------------------------
1 D_A            0.792673      0.048397      16.38      0.0001
2 COR           -0.212526      0.043527      -4.88      0.0001

No. cases = 66        R-sq. =   0.8686      RMS Error = 0.7722
Resid. df = 64    R-sq-adj. =   0.8645       Cond. No. = 2.821
```

Figure 6.9 shows a graph of residuals of N_DA according to the model MOD_COR_AC as a function of case number. All residuals lie between -2 and +2 except for two outliers marked with a circle. The measurements of the acid number were done with an accuracy of 0.5 point. On the basis of the fact, that about 80% of the cases lie between -1 and +1, and taking into account the poor reproducibility of the process, it was decided to accept the model MOD_COR_AC.

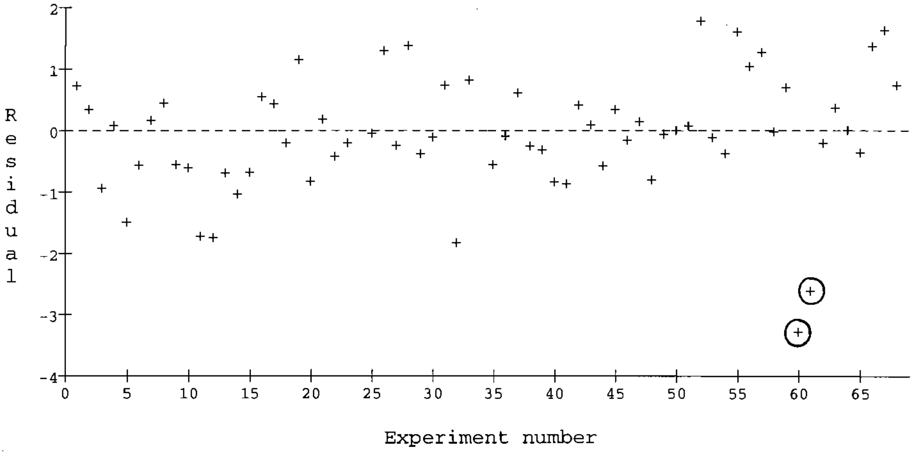


Figure 6.9 Case order graph of residuals of N_DA according to the model MOD_COR_AC

Similar calculations were done for the response N_DV. Equation (6.24) and Table 6.13 present the estimated coefficients of the model MOD_COR_VIS.

$$\text{N\_DV} = 1.115 \cdot \text{D\_V} - 10.622 \cdot \text{COR} \qquad (6.24)$$

Table 6.13 Correction model of the response N_DV

```
Least Squares Coefficients, Response N_DV, Model MOD_COR_VIS

Term            Coeff.       Std. Error    T-value    Signif.
-----------------------------------------------------------------
1 D_V          1.114872       0.089457      12.46      0.0001
2 COR        -10.622187       2.384012      -4.46      0.0001

No. cases = 66       R-sq. =   0.7789       RMS Error = 43.85
Resid. df = 64   R-sq-adj. =   0.7720       Cond. No. = 2.837

                 R-sq-PRESS  =   0.768
```

The residuals computed on the basis of this model are shown in Figure 6.10. Most of them lie between -80 and +80. The viscosity measurement had an accuracy of 10% of the measured value. All this gives sufficient confidence to tentatively accept the model MOD_COR_VIS.

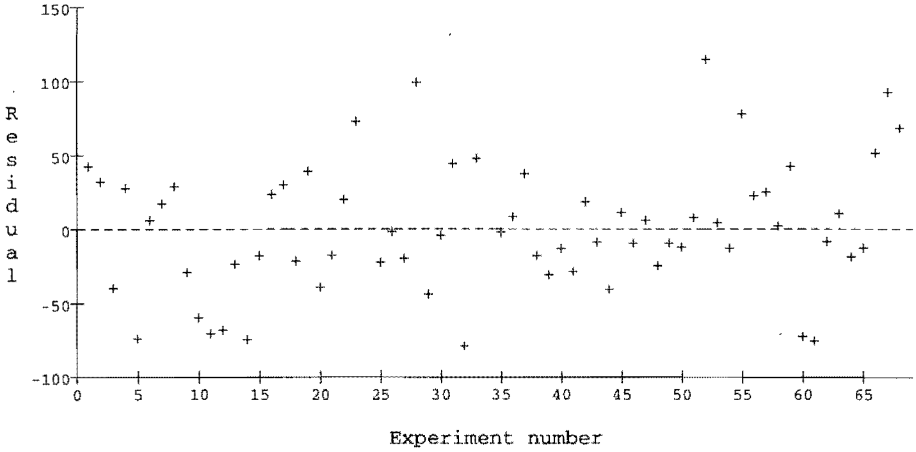


Figure 6.10 Case order graph of residuals of N_DV according to the model MOD_COR_VIS

6.1.4.4 Discussion of Both Correction Models

Before discussing the use of the estimated model for the computation of a correction rule, it is useful to place a number of notes.
According to both models, the size of the correction does not depend on time, and the estimated responses are linearly dependent on the size of the correction and the distance (measured vertically or horizontally) from the heart line.

In essence, the models are contradictory: if N_DA is only dependent on D_A and COR, it is impossible that N_DV is dependent only on D_V and COR, because of the non-linear character of the heart line. This can be seen in Table 6.14, which shows correction sizes, according to the model MOD_COR_AC, respectively MOD_COR_VIS, needed to arrive within half an hour at the heart line. However, only one model can be used for the computation of the correction rule.

Table 6.14 The expected deviations N_DA and N_DV from the heart line after corrections COR according to the models MOD_COR_AC and MOD_COD_VIS; cases 1÷ 4 are in the beginning of the solvent step and cases 5 ÷ 8 in the final part.

No	D_A	D_V	COR	N_DA according to MOD_COR_AC with 95% confidence interval			N_DV according to MOD_COR_VIS with 95% confidence interval		
1	1	20	3.73	-0.32	**0.00**	0.32	-36.32	**-17.32**	1.68
2	1	20	2.10	0.19	**0.35**	0.50	-9.48	**0.00**	9.48
3	2	30	7.46	-0.64	**0.00**	0.64	-85.35	**-45.80**	-6.24
4	2	30	3.15	0.70	**0.92**	1.13	-14.21	**0.00**	14.21
5	1	100	3.73	-0.32	**0.00**	0.32	57.01	**71.87**	86.73
6	1	100	10.50	-2.49	**-1.44**	-0.38	-47.38	**0.00**	47.38
7	2	150	7.46	-0.64	**0.00**	0.64	60.02	**87.99**	115.96
8	2	150	15.75	-3.30	**-1.76**	-0.22	-71.07	**0.00**	71.07

The correction computed on the basis of the one model does not lead to the heart line according to the other model. This discrepancy is particularly pronounced in the beginning and near the end of the acid-viscosity trajectory. However, it should be taken into account that the data used for model estimation mainly provide information about corrections in the middle of the solvent step, where, as shown in Table 6.15, the contradiction is not pronounced.

Table 6.15 The expected deviations N_DA and N_DV from the heart line after corrections COR according to the models MOD_COR_AC and MOD_COD_VIS; all cases are in the middle of the solvent step

No	D_A	D_V	COR	N_DA according to MOD_COR_AC with 95% confidence interval			N_DV according to MOD_COR_VIS with 95% confidence interval		
1	1	40	3.73	-0.32	**0.00**	0.32	-11.33	**4.97**	21.27
2	1	40	4.20	-0.47	**-0.09**	0.27	-18.95	**0.00**	18.95
3	2	70	7.46	-0.64	**0.00**	0.64	-35.01	**-1.20**	32.61
4	2	70	7.35	-0.61	**0.02**	0.66	-33.17	**0.00**	33.17

Because the model MOD_COR_AC is a bit better than the model MOD_COR_VIS (see R^2 and R^2_{adj}), it was included in the recipe adaptation set as the correction model. Thus, it became easy to compare the correction results according to the flexible-recipe approach with the results according to the standard Akzo method.

6.1.4.5 Batch Correction According to the Model MOD_COR_AC

If the selected model MOD_COR_AC is used for the computation of the correction rule with the intention that N_DA for the present heart line must be zero, then it yields:

$$\text{COR} = 3.73\ \text{D\_A} \qquad (6.25)$$

This means that instead of one unit of the corrective substance per point deviation on the acid number scale, 3.73 units of the correction substance should be used !

The correction model can be used every time the specification band changes, for example, when the heart line is shifted +0.5 point in the ACID-direction. On the other hand, if the aim is to arrive at +0.5 ACID-point above the heart line after correction, the rule would become:

$$\text{COR} = -2.35 + 3.73\ \text{D\_A} \qquad (6.26)$$

These results are clearly different from the strategy used by Akzo, which may explain the number of batches not successfully terminated after correction according to their method. As of now, the rule (6.25) will be the correction rule used by Akzo.

6.1.5 Conclusions

This section described an application where data from the industrial polymerisation process were processed and exploited following the flexible recipe-approach, so as to find an improved master control recipe, firstly in case the recipe ingredients were changed, and secondly, in case the processing time was to be shortened. Thirdly, our findings showed what contributions can be made by the FRIS-approach to the improvement of the correction rule in the event the process does not progress according to the acid-viscosity band. For all three cases, reasonably satisfactory models were found for the responses: AC_end, VIS_end, H_cor, TIME_AC12 and N_DA.

Looking back upon the modelling exercise in case of new recipe ingredients, it may be concluded that better and more complete information about the process behaviour would have been obtained, if it had been possible to plan experiments according to a customary experiment design scheme. The available process data were ill-conditioned and rather scarce. Notwithstanding that, it turned out to be possible to find process conditions that came closer in the desired end specification, and it was found to be profitable to adapt the recipe every time the prices of feedstocks are changed.

Looking back at this case study, the following conclusions may be drawn:

1) the FRIS-approach used in an industrial environment proved a number of its advantages: without upsetting normal production, information can be collected and then be used to derive (in a simple fashion) improved recipes or adjusted recipes in case corrections are needed;
2) once one has obtained estimated models, they can be used to find a recipe in case the end specification, or the prices or the processing time are changed;
3) the quality of the estimated models, and therefore also of the obtained results, would be better if the data were better conditioned and the measurements were more accurately performed; this emphasizes how important it is to apply the flexible-recipe approach from the very start

6.2 Epoxy Resin Production

The second application of the flexible recipe approach described in this chapter concerns the production of epoxy resin Epikote 1001 in the laboratory of our University. The function of this case was to test the most important aspects of the FRIS-approach: development of a recipe adaptation set, model-based process optimization, batch initialization and batch correction.
The production of epoxy resin was chosen as a test process because of its reasonable reproducibility, simple requirements regarding instrumentation and relatively short processing time.

6.2.1 Reaction Mechanism

Epoxy resins all have the epoxy group in common [LEE82]. This consists of a triangle with one oxygen and two carbon atoms, see Figure 6.11. To produce epoxy resins two main processes are used in industry: the *taffy* and the *advancement process*. The advancement process, using the Diglycidylether of Bisphenol A (DGEBA) and Bisphenol A (BA), is the most widely applied process because of the relative simplicity of the reaction.

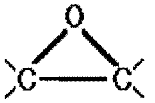


Figure 6.11 The epoxy group

The know-how of this process was available from Sheby, a sister company of Scado (both companies were part of the Polymer Division of Unilever).

The process starts after BA and DGEBA have been mixed and heated in a reaction vessel under nitrogen until the mixture becomes homogeneous (≈115°C). At this point a basic catalyst is added. At a temperature of about 140°C the exothermic reaction starts (see Figure 6.12). This raises the temperature to 180-190°C. Next, the mixture will be held at a temperature of 180°C for about two hours to complete the reaction.

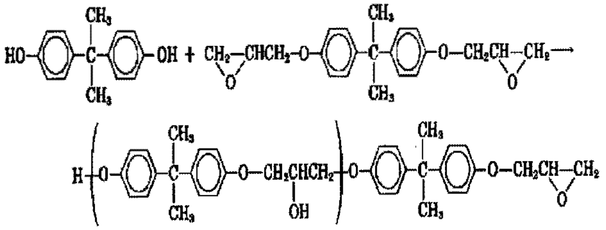


Figure 6.12 The reaction of BA with pure DGEBA

The mole ratio DGEBA:BA governs the degree of polymerization.
By means of a titration, the Weight per Epoxy group (WPE) can be measured. This tells us something about the average chain length of a resin molecule. This WPE will be used as the product quality parameter during the creation of the recipe adaptation set for this process.

6.2.2 Process Modelling Helps to Find Unexpected Variations in Feedstock Quality

As already mentioned, one of the important final qualities of the formed epoxy resin Epikote 1001 is the WPE-number. For this application it was assumed, that the final WPE-number, denoted by WPE_1001, has to satisfy the following specification:

$$480 \leq \text{WPE\_1001} \leq 520; \tag{6.27}$$
the desired value is 500.

To find process conditions which result in the desired end quality specification, a process model of the final WPE-number was estimated. However, it was not quite clear which recipe items should be included in the flexible recipe. For this reason recipe development took place in a number of steps: it started with the variation of three factors, then their number was enlarged to four and next to five. Finally, one external factor, the quality of one ingredient, was involved too. Reason for these enlargements was the increasing process knowledge.

The following process inputs (1 ÷ 3), process parameters (4 ÷ 5) and one external factor (6) were eventually chosen as recipe items and then varied during the experimentation:

1) amount of Bisphenol A (BA);
2) amount of Diglycidylether of Bisphenol A (DGEBA);
3) amount of catalyst (KAT);
4) reaction time (TIME);
5) reaction temperature (TEMP)
6) WPE-number of Diglycidylether of Bisphenol A (WPE_828).

The measured process output was the WPE-number of Epikote 1001 (WPE_1001).

The worksheet with the test runs carried out, intended for the model estimation of WPE_1001, is presented in Table 6.16. The experiments are divided into four groups.

In the first group (exp. 1 ÷ 17) only three factors BA, DGEBA and KAT, were varied according to the Central Composite Circumscribed Design for a fixed value of TIME and TEMP. The last three test-runs of this series were carried out in the centre of the design.

Table 6.16 Worksheet with experiments

Exp.	BA	DGEBA	KAT	TIME	TEMP	WPE_ 828	WPE_ 1001
1	96	355	0.29	108	180	194	458
2	118	355	0.29	108	180	194	577
3	96	433	0.29	108	180	194	388
4	118	433	0.29	108	180	194	459
5	96	355	0.37	108	180	194	458
6	118	355	0.37	108	180	194	599
7	96	433	0.37	108	180	194	401
8	118	433	0.37	108	180	194	459
9	88	394	0.33	108	180	194	388
10	126	394	0.33	108	180	194	554
11	107	328	0.33	108	180	194	548
12	107	460	0.33	108	180	194	398
13	107	394	0.26	108	180	194	459
14	107	394	0.40	108	180	194	470
15	107	394	0.33	108	180	194	470
16	107	394	0.33	108	180	194	469
17	107	394	0.33	108	180	194	468
18	96	355	0.29	132	180	194	472
19	118	355	0.29	132	180	194	590
20	96	433	0.29	132	180	194	405
21	118	433	0.29	132	180	194	461
22	96	355	0.37	132	180	194	474
23	118	355	0.37	132	180	194	594
24	96	433	0.37	132	180	194	410
25	118	433	0.37	132	180	194	474
26	107	394	0.33	96	180	194	456
27	107	394	0.33	144	180	194	462
28	118	355	0.29	108	150	194	534
29	96	433	0.29	108	150	194	412
30	96	355	0.37	108	150	194	456
31	118	433	0.37	108	150	194	457
32	96	355	0.29	132	150	194	451
33	118	433	0.29	132	150	194	454
34	118	355	0.37	132	150	194	587
35	96	433	0.37	132	150	194	388
36	107	394	0.33	120	135	194	453
37	107	394	0.33	120	195	194	461
38	118	355	0.29	108	180	190	543
39	118	355	0.29	108	150	190	548
40	96	433	0.29	108	150	190	378
41	96	355	0.37	108	150	190	452
42	118	433	0.37	108	150	190	449
43	88	394	0.33	108	180	190	380
44	126	394	0.33	108	180	190	520
45	107	328	0.33	108	180	190	542
46	107	460	0.33	108	180	190	391
47	107	394	0.33	120	195	190	446
48	118	430	0.37	165	108	190	456
49	96	355	0.33	120	150	190	446
50	118	433	0.33	120	150	190	446

After the first series was completed TIME was included as a new factor in the recipe adaptation set and the full factorial design was applied for four factors. Because the experiments for lower value of TIME were already performed, only eight experiments were needed (exp. 18 ÷ 25) for the high value of TIME. Experiments 26 and 27 were the star points for TIME of this design; see Appendix A.

Similarly, the next group of experiments (exp. 28 ÷ 37) was defined after TEMP was included in the set of recipe items.

The last group of experiments (exp. 38 ÷ 50) was defined after unexpected variations were discovered in the final WPE-number of Epikote 1001 with respect to the values predicted by the model estimated after 37 test runs. Subsequent investigation revealed, that these variations were caused by the quality, especially the WPE-number of DGEBA that was used, called WPE_828. The second portion of Diglycidylether of Bisphenol A was from another batch than the first one. The specification of that product, which affects the quality of the resin Epikote 1001, was not the same. For the old resin the number WPE_828 was 194, for the new one it was 190 [VER94c].

The results of all 50 test runs were used in the estimation of the process model presented by equation (6.28) and in Table 6.17:

$$\begin{aligned}\mathrm{WPE\_1001} = {} & 474.204 + 76.762\cdot(\sim\mathrm{BA}) - 76.027\cdot(\sim\mathrm{DGEBA}) \\ & + 7.824\cdot(\sim\mathrm{KAT}) + 6.585\cdot(\sim\mathrm{TEMP}) \\ & + 8.242\cdot(\sim\mathrm{TIME}) + 7.183\cdot(\sim\mathrm{WPE\_828}) \\ & - 40.756\cdot(\sim\mathrm{BA}\cdot\sim\mathrm{DGEBA}) + 11.532\cdot(\sim\mathrm{BA}\cdot\sim\mathrm{KAT}) \\ & - 10.235\cdot(\sim\mathrm{DGEBA}\cdot\sim\mathrm{KAT}) + 10.156\cdot(\sim\mathrm{DGEBA})^2 \\ & - 25.530\cdot(\sim\mathrm{TEMP})^2 - 18.787\cdot(\sim\mathrm{TIME})^2 \end{aligned} \qquad (6.28)$$

where:

$$\sim\mathrm{BA} = (\mathrm{BA} - 107) / 19 \qquad (6.29)$$

$$\sim\mathrm{DGEBA} = (\mathrm{DGEBA} - 394) / 66 \qquad (6.30)$$

$$\sim\mathrm{KAT} = (\mathrm{KAT} - 0.33) / 0.07 \qquad (6.31)$$

$$\sim\mathrm{TEMP} = (\mathrm{TEMP} - 165) / 30 \qquad (6.32)$$

$$\sim\mathrm{TIME} = (\mathrm{TIME} - 120) / 24 \qquad (6.33)$$

$$\sim\mathrm{WPE\_828} = (\mathrm{WPE\_828} - 192) / 2 \qquad (6.34)$$

Table 6.17 Model of the final WPE-number. The non-significant terms are not shown.

```
Least Squares Coefficients, Response WPE_1001, Model WPE_RED

Term               Coeff.      Std. Error   T-value  Signif.  Transformed Term
----------------------------------------------------------------------------------
 1 1             474.204058     4.724451    100.37    0.0001
 2 ~BA            76.762004     2.263975     33.91    0.0001   ((BA-1.07e+02)/1.9e+01)
 3 ~DGEBA        -76.027227     2.231006    -34.08    0.0001   ((DGEBA-3.94e+02)/6.6e+01)
 4 ~KAT            7.824398     2.529443      3.09    0.0038   ((KAT-3.3e-01)/7e-02)
 5 ~TEMP           6.584537     2.527179      2.61    0.0131   ((TEMP-1.65e+02)/3e+01)
 6 ~TIME           8.242292     2.705849      3.05    0.0043   ((TIME-1.2e+02)/2.4e+01)
 7 ~WPE_828        7.183309     1.501066      4.79    0.0001   ((WPE_828-1.92e+02)/2)
 8 ~BA*DGEBA     -40.756365     4.552465     -8.95    0.0001
 9 ~BA*KAT        11.532203     4.790017      2.41    0.0212
10 ~DGEBA*KAT    -10.234964     4.712417     -2.17    0.0363
11 ~DGEBA**2      10.155850     5.110679      1.99    0.0543
12 ~TEMP**2      -25.530343     7.813849     -3.27    0.0023
13 ~TIME**2      -18.786691     8.242733     -2.28    0.0285

No. cases = 50         R-sq.  =   0.9848       RMS Error = 8.534
Resid. df = 37     R-sq-adj.  =   0.9799       Cond. No. = 9.093

~ indicates factors are transformed.

               R-sq-PRESS  =   0.972
```

6.2.3 Statistical Analysis of the Model

Because the values of R^2, R^2_{adj} and R^2_{PRESS} were satisfactory, we continued with the analysis of variance of the estimated model, the results of which are presented in Table 6.18. The lack of fit is quite large in comparison with the pure error. This might suggest that some significant terms may be missing from the model, but on the other hand, the pure error is estimated on the basis of only three replications (see Table 6.16. exp. 15, 16 and 17); the measured values of WPE_1001 are almost the same and therefore the pure error is actually very small.

Nevertheless, it was decided to go on with this model and to decide after residual analysis whether it will be accepted or not. Figure 6.13 shows the histogram and Figure 6.14 the plot of residuals versus experiment number. Both figures do not indicate model inadequacy or deviation from a normal distribution as presumed in regression analysis. Therefore, in spite of the lack of fit, it was decided to tentatively accept the model and to pass on to the investigation of the main effects of each recipe item. Table 6.19 shows these main effects with their 95% confidence intervals.

Table 6.18 The variance analysis of the estimated model

```
Least Squares Summary ANOVA, Response WPE_1001, Model WPE_RED
Source              df    Sum Sq.      Mean Sq.    F-Ratio   Signif.
-------------------------------------------------------------------
1 Total(Corr.)      49    177146.6
2  Regression       12    174451.9     14537.7     199.60    0.0000
3   Linear           6    163767.8     27294.6     374.80    0.0000
4   Non-linear       6      7895.4      1315.9      18.07    0.0000
5  Residual         37      2694.8        72.8
6   Lack of fit     35      2692.8        76.9      76.94    0.0129
7   Pure error       2         2.0         1.0

F(35,2) as large as 76.94 is a rare event =>
    likely that significant terms are missing from model.
```

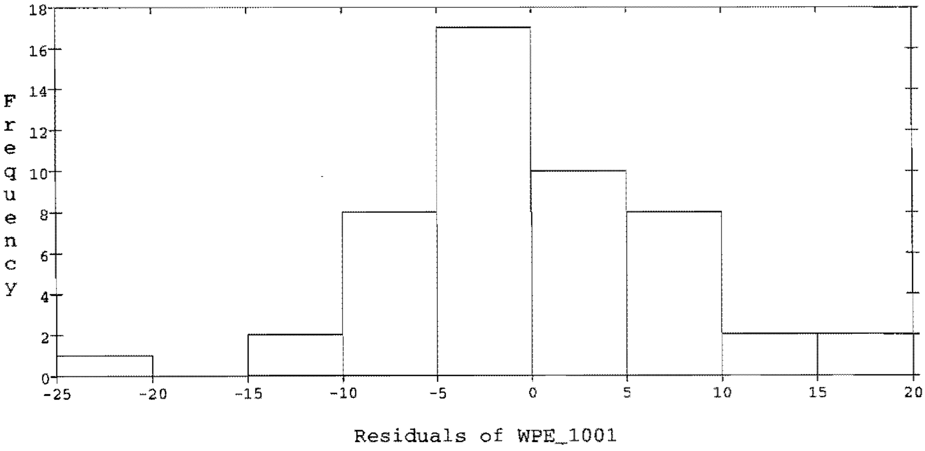


Figure 6.13 Histogram

It should be noted, that BA and DGEBA have nearly the same effect in the chosen experimentation area, albeit the first one is positive and the second one negative. All remaining recipe items have positive main effects on the response, the effects are smaller than those of first two and their 95% confidence intervals are relatively large, which means that the corresponding effects are estimated less accurately. This information will next be used in the interpretation of batch initialization results.

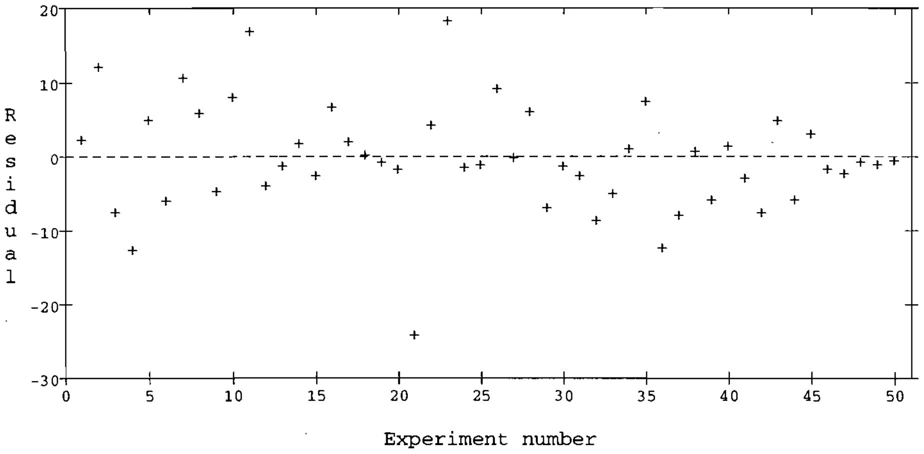


Figure 6.14 Graph of residuals versus experiment number

Table 6.19 Main effects with 95% confidence intervals on the response WPE_1001

Predictor	Settings	Lower Bound	**Estimated Effect**	Upper Bound
BA	88 to 126	144.42	**153.60**	162.77
DGEBA	328 to 460	-165.32	**-153.94**	-142.57
KAT	0.26 to 0.4	5.96	**16.20**	26.44
TEMP	135 to 168.87	10.70	**32.54**	54.38
TIME	96 to 125.26	5.95	**27.93**	49.91
WPE_828	190 to 194	8.28	**14.37**	20.45

6.2.4 Performance Criterion

To examine the "economic side" of the production, a performance criterion defined as the average added value per unit of time was chosen. It is formulated here as the difference between component values at the end of the reaction and the costs of energy and ingredients used for the production, divided by the total time of the batch cycle:

$$J = (\$_{E\_1001}\ E\_1001 - \$_{BA}\ BA - \$_{DGEBA}\ DGEBA - \$_{ENERG}\ ENERGY) / t_{tot} \tag{6.35}$$

where:

$\$_{E\_1001}$, $\$_{BA}$, $\$_{DGEBA}$: benefits and costs of product, respectively reactants;

E_1001 : amount of produced Epikote 1001;

$\$_{ENERG}$: costs of the used energy;

ENERGY : amount of the consumed energy;

t_{tot} : the total batch time; it is the sum of the reaction time TIME and the remaining time needed to complete the batch cycle, that is the time of ingredients dosage, heating and reactor cleaning.

The following prices were used:

$\$_{E\_1001}$ = 1.0 \$/gr
$\$_{BA}$ = 0.528 \$/gr
$\$_{DGEBA}$ = 0.655 \$/gr
$\$_{ENERGY}$ = 0.02 \$/MJ
E_1001 = BA + DGEBA
ENERGY = const · TIME · (TEMP-19)
t_{tot} = TIME + 20 min
const = 0.05 MJ/min·°C; constant for the reactor, which indicates how much energy is needed to keep the reactor at 20°C during one minute.

6.2.5 Constraints

Besides the following model-validity constraints:

$$96 \leq BA \leq 118 \tag{6.36}$$

$$355 \leq DGEBA \leq 433 \tag{6.37}$$

$$0.29 \leq KAT \leq 0.37 \tag{6.38}$$

$$108 \leq TIME \leq 132 \tag{6.39}$$

$$150 \leq TEMP \leq 180 \tag{6.40}$$

It was decided to include in the recipe adaptation set two additional process constraints: bounds on the amount of the used catalyst, and bounds on the final WPE-number.

To avoid the resin becoming muddy, what happened occasionally during experimentation if more than 0.33 ml catalyst was added into reactor, we restricted the catalyst amount:

$$KAT \leq 0.33 \tag{6.41}$$

Because of model inaccuracy and to be sure that the final WPE-number does not exceed the allowed specification bounds mentioned in (6.27), it was decided to add a tighter constraint on the process output:

$$499 \leq \text{WPE\_1001} \leq 501. \qquad (6.42)$$

6.2.6 Search for the Master Recipe

To obtain the master recipe, the following model-based recipe optimization must be done:

$$\mathit{max}\ J, \quad \textit{subject to:} \quad 499 \leq \mathit{WPE\_1001} \leq 501; \quad \mathit{KAT} \leq 0.33. \qquad (6.43)$$

This is an optimization problem of the non-linear type with non-linear constraints, that is:

$$\max_{x} \ (0.472 \cdot x_1 + 0.345 \cdot x_2 - 0.001 \cdot x_4 \cdot (x_5 - 19))/(20. + x_4), \quad x \in \mathbb{R}^6 \qquad (6.44)$$

subject to:

$$\begin{array}{llll} 96 & \leq x_1 & \leq 118 & (6.45) \\ 355 & \leq x_2 & \leq 433 & (6.46) \\ 0.29 & \leq x_3 & \leq 0.33 & (6.47) \\ 108 & \leq x_4 & \leq 132 & (6.48) \\ 150 & \leq x_5 & \leq 180 & (6.49) \\ 190 & \leq x_6 & \leq 190 & (6.50) \end{array}$$

$$\begin{aligned} 499 \leq & - 25.40 \cdot 10^2 + 13.98x_1 + 1.22x_2 - 56.85x_3 + 8.17x_4 + 9.58x_5 \\ & + 3.59x_6 - 32.50 \cdot 10^{-3}x_1x_2 + 86.71 \cdot 10^{-1}x_1x_3 - 22.15 \cdot 10^{-1}x_2x_3 \\ & - 23.31 \cdot 10^{-4}x_2^2 - 32.62 \cdot 10^{-3}x_4^2 - 28.37 \cdot 10^{-3}x_5^2 \leq 501 \qquad (6.51) \end{aligned}$$

where x_1 corresponds to BA, x_2 to DGEBA, x_3 to KAT, x_4 to TIME, x_5 to TEMP and x_6 to WPE_828.

It results in the recipe setting presented in Table 6.21. Before these settings can be accepted as nominal ones it was decided to do two validation test-runs; these resulted in WPE_1001 values of 496.9 and 491.6, which gave no cause for rejection of the model.

Table 6.21 Optimized recipe settings of the master recipe

BA DGEBA KAT TIME TEMP WPE_828	118 395 0.33 108 160 190
The **expected** WPE_number **WPE_1001**	**499**
its 95% confidence interval	[479; 519]
Performance index	1.36

6.2.7 Batch Initialization. Why It Is Occasionally not as Successful as Desired?

As presented in Chapter 4, in those cases where initial process conditions are disturbed, or the prices of energy, used materials or products have unexpectedly changed, it makes sense to search for better process conditions, which can compensate as well as possible for these deviations.

Table 6.22 shows results of six batch initialization cases with changed initial process conditions. All these runs are performed for KAT = 0.33 and WPE_828 = 190, and with the rule: perform initialization if the expected value of WPE_1001 without initialization is lower than 490 or larger than 510.

In the first four cases the profit of batch initialization is evident: the all measured values of WPE_1001 do not differ from 500 more than 10. The last two cases, with identical recipe items, cause some disappointment: the measured number WPE_1001 deviates more from the expected vale than without initialization. However, it is quite logically, because in these two batches no ingredients are used as initialized variables, but only TIME and TEMP. As presented in Table 6.19, it are the ingredients that determine the response, and not the other recipe items, which have quite small main effects with relatively large confidence intervals.

Table 6.22 Results of batch initialization. The disturbed recipe items are presented in italics, the initialized ones are double underlined.

	1	2	3	4	5	6
BA	*114*	*126*	118	*108*	*113*	*113*
DGEBA	383.3	412	386	363	*388*	*388*
TIME	108	108	108	108	118	118
TEMP	*158.9*	156	*145*	162	168	168
expected WPE_1001 without initialization	482.7	531.4	485.4	461.0	488.8	488.8
expected WPE_1001 with initialization	499.1	499.6	499.1	500.0	499.0	499.0
measured WPE_1001 after batch initialization	495.5	504.5	508.4	499.0	484.0	484.4

The initialization results as a whole show that it is possible to achieve the final WPE-number according to the end specification (6.27). The performed batches clearly indicate that the best results can be obtained using BA and/or DGEBA as initialized variables.

6.2.8 Batch Correction

Batch correction, intended to compensate for the effects of (unknown) disturbances, is applied to the production process of Epikote 1001 when some deviation in process conditions with respect to the expected values are measured during a run.

In the examples discussed below, a sample of the reaction mixture is taken at the moment t_S, and the WPE-number is measured. If it is other than expected, then correction is done some time later, at t_C. Because the time needed for sample analysis was about 20 minutes, and some time was needed for the correction to yield a result, it was decided that t_S = 50 min and t_C = 70 min.

Here, the correction variables are:

- amount of BA added at the correction moment:	BA_COR;
- amount of DGEBA added at the correction moment:	DG_COR;
- reaction time, which can be reduced or prolonged:	TIME;
- temperature, which can be changed after t_C:	TEMP_COR.

The response, the measured WPE-number of the sample, is denoted by WPE_50.

6.2.8.1 *Process Models Required for Correction*

As defined in Chapter 4, besides the performance criterion and the information about the detected deviations, for performing batch correction two correction models, are necessary. These models, called *Model_C1* and *Model_C2*, will be described now.

Model_C1

This model predicts the value of the WPE-number measured at the sample moment t_S as a function of the initial values of BA, DGEBA, TEMP and WPE_828. The amount of catalyst KAT is assumed to be fixed at the nominal value, therefore it does not occur in the model.

The WPE-number at t_S = 50 min, called WPE_50, was estimated by *Model_C1* presented in equation (6.52) and Table 6.23.

Table 6.23 First correction model and the analysis of variance of the response WPE_50

```
Least Squares Coefficients, Response WPE_50, Model MODEL_C1_WPE_50

Term            Coeff.          Std. Error   T-value  Signif. Transformed Term
---------------------------------------------------------------------------------
1 1               452.008333     1.158315    390.23   0.0001
2 ~BA              40.541667     1.234768     32.83   0.0001  ((BA-1.07e+02)/1.1e+01)
3 ~DGEBA          -41.666667     1.234768    -33.74   0.0001  ((DGEBA-3.94e+02)/3.9e+1)
4 ~BA*DGEBA        -5.550000     1.209821     -4.59   0.0004
5 ~WPE_828          8.091667     1.158315      6.99   0.0001  ((WPE_828-1.92e+2)/2)
6 ~WPE_828*BA       5.458333     1.234768      4.42   0.0005
7 ~WPE_828*DGEBA -2.583333     1.234768     -2.09   0.0538

No. cases = 22        R-sq. =  0.9936       RMS Error = 5.41
Resid. df = 15   R-sq-adj. =  0.9910      Cond. No. = 1.225
~ indicates factors are transformed.

                  R-sq-PRESS = 0.984
=================================================================================
  Least Squares Summary ANOVA, Response WPE_50 Model MODEL_C1_WPE_50

Source           df    Sum Sq.     Mean Sq.    F-Ratio  Signif.
-----------------------------------------------------------------
1 Total(Corr.)    21    68176.36
2  Regression      6    67737.26    11289.54    385.70    0.0000
3   Linear         3    66319.51    22106.50    755.20    0.0000
4   Non-linear     3     1316.22      438.74     14.99    0.0001
5  Residual       15      439.10       29.27
6   Lack of fit   10      438.60       43.86    438.60    0.0000
7   Pure error     5        0.50        0.10

F(10,5) as large as 438.6 is a very rare event =>
    highly likely that significant terms are missing from model.
```

As in the case of the nominal process model (6.28), a lack of fit had to be concluded also here. The very small pure error may explain this fact.

$$\begin{aligned}\mathrm{WPE\_50} = {} & 452.204 + 40.542\cdot(\sim\mathrm{BA}) - 41.667\cdot(\sim\mathrm{DGEBA}) \\ & + 8.092\cdot(\sim\mathrm{WPE\_828}) - 5.550\cdot(\sim\mathrm{BA}\cdot\sim\mathrm{DGEBA}) \\ & + 5.458\cdot(\sim\mathrm{BA}\cdot\sim\mathrm{WPE\_828}) \\ & - 2.583\cdot(\sim\mathrm{DGEBA}\cdot\sim\mathrm{WPE\_828})\end{aligned} \tag{6.52}$$

where:

$$\sim\mathrm{BA} = (\mathrm{BA} - 107) / 11 \tag{6.53}$$

$$\sim\mathrm{DGEBA} = (\mathrm{DGEBA} - 394) / 39 \tag{6.54}$$

$$\sim\mathrm{WPE\_828} = (\mathrm{WPE\_828} - 192) / 2 \tag{6.55}$$

Model_C2

According to the definition given in Chapter 4, the second correction model *Model_C2* should give the final WPE-number, WPE_1001, as a function of the measured value of WPE_50 and the correction variables. However, to be more precise in this case, two correction models were used to represent the above-mentioned relation. On the basis of the measurements intended to choose the best sample moment, it was possible to estimate the WPE-number at the correction moment t_C, called WPE_70, as a function of WPE_50. This model, called *MODEL_C2_1* is presented by (6.56) and in Table 6.24. Because no test run was repeated here, it was not possible to estimate pure error. Therefore the analysis of variance is omitted here.

$$\mathrm{WPE\_70} = 473.908 + 102.136\cdot(\sim\mathrm{WPE\_50}) + 3.079\cdot(\sim\mathrm{WPE\_50})^2 \tag{6.56}$$

where:

$$\sim\mathrm{WPE\_50} = (\mathrm{WPE\_50} - 470) / 97 \tag{6.57}$$

Table 6.24 Second correction model of the response WPE_70

```
Least Squares Coefficients,  Response WPE_70, Model MODEL_C2_1

Term            Coeff.       Std. Error  T-value  Signif.  Transformed Term
-------------------------------------------------------------------------------
1 1             473.908210    0.670563   706.73    0.0001
2 ~WPE_50       102.136100    0.884617   115.46    0.0001   ((WPE_50-4.7e+02)/9.7e+01)
3 ~WPE_50**2      3.078959    1.285879     2.39    0.0271

No. cases = 22         R-sq. =   0.9987       RMS Error = 2.299
Resid. df = 19    R-sq-adj. =   0.9985       Cond. No. = 2.496
~ indicates factors are transformed.

                 R-sq-PRESS =   0.988
```

Next, the third correction model, called *Model_C2_2* was estimated. It gives the relation between the estimated WPE-number at $t_C = 70$ min, the corrective variables and the final WPE-number WPE_1001. This model is presented by (6.58) and in Table 6.25.

$$\begin{aligned}\text{WPE\_1001} = & - 0.207 + 0.117\cdot(\sim\text{BA\_COR}) - 0.193\cdot(\sim\text{DG\_COR}) \\ & + 0.030\cdot(\sim\text{TIME}) + 0.631\cdot(\sim\text{WPE\_70}) \\ & + 0.014\cdot(\sim\text{BA\_COR}\cdot\sim\text{TIME}) \\ & + 0.020\cdot(\sim\text{BA\_COR}\cdot\sim\text{WPE\_70}) \\ & - 0.118\cdot(\sim\text{DG\_COR}\cdot\sim\text{WPE\_70}) \\ & + 0.118\cdot(\sim\text{DG\_COR}\cdot\sim\text{WPE\_70}) \\ & + 0.020\cdot(\sim\text{TIME}\cdot\sim\text{WPE\_70}) \end{aligned} \qquad (6.58)$$

where:

$$\sim \text{BA\_COR} = (\text{BA\_COR} - 5.87) \,/\, 5.87 \qquad (6.59)$$
$$\sim \text{DG\_COR} = (\text{DG\_COR} - 20.145) \,/\, 20.145 \qquad (6.60)$$
$$\sim \text{TIME} = (\text{TIME} - 120) \,/\, 20 \qquad (6.61)$$
$$\sim \text{WPE\_70} = (\text{WPE\_70} - 456) \,/\, 81.6 \qquad (6.62)$$

Table 6.25 Third correction model of the response WPE_1001

```
Least Squares Coefficients, Response WPE_1001, Model MODEL_C2_2

0 Term                Coeff.    Std. Error  T-value  Signif.  Transformed Term

-------------------------------------------------------------------------------
1 1                 -0.206745   0.005343    -38.70   0.0001
2 ~BA_COR            0.117359   0.005736     20.46   0.0001    ((BA_COR-5.87)/5.87)
3 ~DG_COR           -0.192873   0.005829    -33.09   0.0001   ((DG_COR-20.145)/20.145)
4 ~TIME              0.029884   0.006491      4.60   0.0001    ((TIME-1.2e+02)/2e+01)
5 ~WPE_70            0.630750   0.007894     79.90   0.0001   ((WPE_70-4.56e+02)/81.6)
6 ~BA_COR*TIME       0.014169   0.006982      2.03   0.0455
7 ~BA_COR*WPE_70     0.019773   0.008357      2.37   0.0202
8 ~DG_COR*WPE_70    -0.118319   0.008710    -13.58   0.0001
9 ~TIME*WPE_70       0.019770   0.009525      2.08   0.0409

~ indicates factors are transformed.


No. cases = 96         R-sq. =   0.9903    RMS Error = 0.05115
Resid. df = 87     R-sq-adj. =   0.9894      Cond. No. = 1.399
~ indicates factors are transformed.

                   R-sq-PRESS =   0.988
```

6.2.8.2 *Results: Always Successful*

Table 6.26 shows the results of four cases with batch correction. The recipe presented in Table 6.21 was used to start each batch. At the sample moment the expected value of WPE_50 according to *Model_C1* was 483.42. The applied correction rule was: if the measured value of WPE_50 differs more than 10 from the expected value, then perform batch correction based upon optimization of the following performance criterion:

$$J = \{ \$_{E\_1001}\, E\_1001 - \$_{BA}\,(BA+BA\_COR) - \$_{DGEBA}\,(DGEBA + DG\_COR) - \$_{ENERG}\, ENERGY \} / t_{tot} \qquad (6.63)$$

according to (6.36) and the model validity constraints 6.59 ÷ 6.62 on the correction variables.

Table 6.26 The results of batch correction. The expected value of WPE_50 is 483.42.

	1	**2**	**3**	**4**
WPE_50 measured at t_S = 50 min	457.63	516.25	466.9	473.24
Correction at t_C = 70 min	BA_COR = 12	DG_COR = 22.9	BA_COR = 7.1	BA_COR = 3.83
expected WPE_1001 without correction	470.7	542.7	481.9	489.6
expected WPE_1001 with correction	499.0	499.0	499.1	499.0
measured WPE_1001 after batch correction	499.9	487.6	494.1	504.0

In all cases the corrections were realized by adding BA or DGEBA at t_C. The other possible corrective variable, TIME, was not used, as could have been expected because of its small effect on the response. As a whole, recipe correction brought about improvement of the final results.

6.2.9 Conclusions

Section 6.2 showed an application of the FRIS-approach to the production of epoxy resin. After the recipe adaptation set was developed, the recipe settings of the master recipe were found. Next, the recipe adaptation set was used to compensate for measured deviations in process conditions before and during batch operations by means of batch initialization and correction.

Because only two factors, BA and DGEBA, had large estimated main effects on the response, and the other factors had quite small effects with wide 95% confidence intervals, batch initialization did not in two cases lead to the expected adjustment. Nevertheless, the final WPE-number after initialization was on specifications.

The batch correction showed that the deviations in the WPE-number, measured during processing, can best be compensated by an extra dosage of ingredient BA or DGEBA. The ultimate effect of such corrections was satisfactory every time.

6.3 Other Applications

Another industrial application of the flexible recipe-approach was done in the area of semiconductor device modelling and optimization in cooperation with the Electrical Engineerig Department of our Univeristy and with Philips Electronics Ltd. [OTT93]. Two types of electronic integrated circuits, that is the Inverse-T LDD (ITLDD) and the Oblique Rotated Implanted LDD (OLDD) MOSFETs, were studied there and compared with each other in terms of bulk current, lateral electric field, source-drain series resistance and drain current [OKU91]. Process models of both types of MOSFETs were estimated after simulated experiments. Then, end specification problems were solved by choosing one output variable (lateral electric field) as a performance criterion which was to be minimized, and by bounding the other outputs within desirable limits. The outcome was that the OLDD MOSFET has the fabrication simplicity of the "normal" LDD MOSFET and the advantages of the ITLDD MOSFET and therefore it is probably most promising for mass production. This work was further extended into recipe initialization with a more sophisticated performance criterion [OTT93].

In Section 2.2.2.3 an industrial FRIS-application regarded an industrial production of powder resins was mentioned. Not only was the best overall performance achieved, but also the development of a recipe adaptation set for one production step made it possible to reduce the concentration of an undesirable by-product from 0.28% to 0.08% [SME95].

In [KEE93] model-based recipe improvement was compared with simplex process-optimization applied to a specific industrial application: the production of benzylalcohol on a pilot plant scale. It was concluded there that for more complex response surfaces, e.g. saddle, the model-based approach is preferable to the simplex strategy.

Chapter 7. Final Considerations

7.1 Conclusions

This thesis focused attention on a practical approach, called the flexible recipe-approach, intended for efficient generation and improvement of a master (control) recipe, and for model-based recipe adjustment in the batch processing industry. The application of this approach was illustrated on the basis of three examples. The first one: the simulated fermentation process, was used to explain the concepts introduced. The two real processes, one real industrial process and one accomplished in the laboratory of our group, demonstrated what benefits can be expected when using the FRIS-approach.

In spite of the fact that the presented applications did not cover the entire flexible recipe-approach, it was shown that the FRIS-strategy indeed leads to process improvement and/or more profit. Further support of this conclusion was provided by a number of simulated applications and a variety of industrial applications carried out during the project, which were mentioned in Section 6.3.

All applications showed that the flexible recipe-approach offers a systematic and fast way to recipe generation and improvement as well as to batch initialization and correction, and therefore also to better performance. The general conclusion, that the FRIS-approach provides invaluable support for the purposes of control and improvement of product quality and quantity, as well as for improving the economy of process operation, is supported by the following more detailed findings:

1) In the process industries, many current batch processes are not modelled at all, but are operated using heuristic process understanding. The first-principle models are only rarely available. The black-box transition models, proposed in this thesis, can be estimated relatively quickly and, what is maybe somewhat surprising, black-box modelling may yield valuable process insight;
2) After suitable approximation of time-dependent recipe items and application of an experiment design method, as described in Section 3.1.3,

near-optimal time-dependent profiles can easily be found;

3) The positive effect of the proposed methods of batch initialization and correction is especially apparent in cases where without them the end product would not meet the specifications;

4) For the purpose of the experimental optimization, the multiplex fitting method, based on a local approximation of the response surface, has been developed. This method is a valuable tool for quickly locating a process optimum, especially in cases when the optimum is expected to lie rather far away from the starting point;

5) Because of its effectiveness and simplicity, the developed Triplet-choice Method for solving multi-objective optimization problems is very useful in the industrial environment for solving end-specification problems involving more than one response;

6) Once well-tried and accepted, a recipe adaptation set can be utilized for process monitoring, so that the need for corrections or the recognition of process drift may be readily established. Process monitoring may also be helpful in improving models and recipes, and in augmenting plant and process knowledge;

7) Most of the existing industrial approaches for achieving consistent and reproducible results from batch processes are based on built-up experience or Statistical Process Control (SPC) analysis [KEA91]. That strategy, which is actually a kind of monitoring, is mainly used for the "stabilization" of the process, that is, for the detection of special causes of process deviations (contrary to common causes of deviations, which are always present), and next for making and keeping the process stable. Examples of the causes of such special process deviations given by SPC-practitioners are: variations in the quality of used materials or in used machines, equipment defects and differences in operating practices between various shifts. It is evident that a number of these causes, e.g. differences between shifts, can and must be eliminated to make the process as reproducible as possible. However, not all causes of deviations, e.g. varying quality of feedstocks, can be permanently eliminated. By accounting for them in a recipe adaptation set, the FRIS-approach achieves the reduction of process variance.

It should be mentioned that for the proposed approach, to be successful, a number of prerequisites is necessary with reference to the process and to the user [RAD95]:

- reasonable process reproducibility;
- adequate, well-calibrated process instrumentation;
- management support;

- sufficient operator involvement and discipline;
- insight into process, plant operation and safety;
- appreciation of statistics and insight in production economics on the plant floor.

7.2 The FRIS-Methodology

My study did not pretend to result in a ready-made, directly-applicable theory concerning the improvement of batch processing recipes. Rather, the described approach serves as a way of thinking and looking at industrial reality and thus as a guide in empirical investigations.

I dare say that the FRIS-approach can be seen as a methodology of industrial process optimization. Here, the term "methodology" should not be understood as the science of science in general. Methodology here, has a more restricted meaning. There are several interpretations of this term. The first and most important one is that it is a part of the philosophy of science, the another one has a praxiological character, that is the way of approaching problems towards a goal [KRA77].

Table 7.1 shows the praxiological view on methodology according to the ideas of the Polish philosopher Kotarbinski [KOT82].

The work described in this thesis can be seen as a **methodology of research actions** towards a practical goal. For the user of the flexible recipe-approach it will be more a **methodology of practical skills**. When I talk about the FRIS-methodology I mean the intersection of both these meanings.

7.3 Prospects

Up till now, my research has mainly employed black-box transition models of one, critical, production phase. Important further steps can be the investigation of first-principle models and light grey models especially, and application of recipe adaptation to more than one process phase.

Recipe transposition, that is the preparation of a similar recipe adaptation set for another plant, forms another interesting subject for further research.

Table 7.1 The praxiological view on methodology with its domains presented in brackets.

<table>
<tr><td colspan="3"><b>General methodology
describing the way of approaching problems towards various goals</b></td></tr>
<tr><td><b>Methodology of theoretical research</b>
(theoretical research questions)</td><td colspan="2"></td></tr>
<tr><td colspan="2"><b>Methodology of research actions</b>
(theoretical and practical research questions)</td><td></td></tr>
<tr><td></td><td colspan="2"><b>Methodology of practical skills</b>
(practical research questions and practical non-research problems)</td></tr>
</table>

Apart from the development of new and the improvement of existing methods much effort has to be put into further software development. As pointed out in Section 5.6 the TNO/TPD Institute is busy defining a new project to develop the FRIS-prototype into a powerful commercial package.

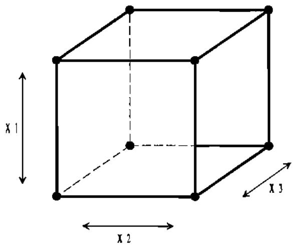
Appendix A Design of Experiments

A.1 Design of Experiments Used for Response Surface Modelling

By way of illustration, a very short description of the factorial design method will be given below.

Table A.1 Full factorial design scheme for 3 factors at two levels

	factor_1	factor_2	factor_3
	x_1	x_2	x_3
1	-1	-1	-1
2	-1	-1	+1
3	-1	+1	-1
4	-1	+1	+1
5	+1	-1	-1
6	+1	-1	+1
7	+1	+1	-1
8	+1	+1	+1



In a so-called 2^k factorial design for k factors, each can have 2 values, around the selected or nominal value, so that 2^k different conditions are created. For 3 factors, the 2^3 dsign of experiments can be simply presented as in Table A.1, where "-1" indicates that, compared to the reference value, this factor is smaller, and "+1" indicates that the factor is larger. It may also be mentioned here that the scaling of factors between -1 and +1 considerably simplifies the computation of model coefficients.

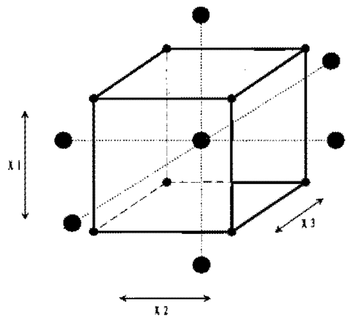
Together all these experiments give sufficient information for the estimation of the following so-called interaction process model (without power terms), which is implicitly assumed by the choice of design of experiments method:

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \sum_{j>i}^{k} \beta_{ij} x_i x_j + \varepsilon \qquad (A.1)$$

where β_0 is the model constant, β_i a model main effect coefficient, β_{ij} a model interaction effect coefficient, x_i the experimental factor, y the response, k is the number of factors selected for experimentation and ε is a stochastic normally-distributed error. Thus, for three factors seven coefficients have to be estimated on the basis of eight experiments.

Table A.2 Extension of full factorial design into central composite design for 3 factors

	factor_1 x_1	factor_2 x_2	factor_3 x_3
1	1.6818	0	0
2	-1.6818	0	0
3	0	1.6818	0
4	0	-1.6818	0
5	0	0	1.6818
6	0	0	-1.6818
7	0	0	0
8	0	0	0
9	0	0	0



If a limited model like the interaction model (A.1) is not satisfactory, then it recommended to estimate a full second-order model with power terms:

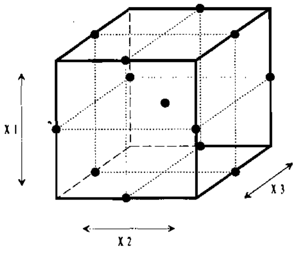
$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \sum_{j\geq i}^{k} \beta_{ij} x_i x_j + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \varepsilon \qquad (A.2)$$

For this purpose the so-called central composite design can be applied to expand factorial design with so-called star points and with a number of experiments in the middle of the design. Table A.2 shows the scheme for this extension for 3 factors, whereby "0" indicates the reference value. Experiments 1÷6 are the star points. Experiments 7÷9 must be done according to the reference values. They are needed to estimate experimental error. As one can see, the star points are situated outside the original experimentation area. When they are not acceptable to run, the star points will be located on the permissible low and high levels. Such experiments form the so-called central composite faced design.

When in the beginning of experimentation there is enough process knowledge to assume that the second-order process model is adequate, it is recommendable to at once use the Box-Behnken design presented in Table A.3.

Table A.3 Box-Behnken design for 3 factors at three levels

	factor_1	factor_2	factor_3
	x_1	x_2	x_3
1	-1	-1	0
2	-1	+1	0
3	+1	-1	0
4	+1	+1	0
5	-1	0	-1
6	-1	0	+1
7	+1	0	-1
8	+1	0	+1
9	0	-1	-1
10	0	-1	+1
11	0	+1	-1
12	0	+1	+1
13	0	0	0
14	0	0	0
15	0	0	0



In principle, all these factorial-like methods work well if the experimentation area is not constrained. In case of constraints, optimal designs [ATK82] give a better solution. In this category a process model structure is explicitly assumed. Experiments are defined by optimization of a specific criterion that in some way comprises the covariance matrix of the estimated coefficients in the assumed model. This clearly differs from factorial designs. Here a process model determines experiments, there the quite reverse is valid: there experiments determine what kind of a model may be estimated. This explicit model assumption can sometimes be a disadvantage of the optimal design method, because the correct structure of the process model is not always known a priori and as consequence the defined experiments are not optimal any more.

A.2 Other Design of Experiments Methods

The factorial and optimal designs are mostly used during the development of a recipe adapter, because they lead to a process model. Sometimes other types of design will be very useful too, for example, sequential or Taguchi design.

The sequential design, as proposed by Nelder and Mead, does not rely on a specific process model, but starts with a number of prespecified experiments, after which new experiments are defined on the basis of the comparison of the results of experiments already performed. It is worthwhile mentioning that this method can easily cope with constraints [SPE62, NEL65]. Because the Nelder-Mead method is very useful in experimental optimization it is in more detail described in Section 3.5 and in Appendix D.

Taguchi designs, often called off-line quality control designs, search for the operating regions where the production is least sensitive to the sources of various "noise factors" (disturbances) [TAG87]. If the effect of such noise factors are rather costly or impossible to control, it may often be preferable to avoid such control if a production region can be found where the natural variations of the noise factors have an insignificant effect on the response of interest. Taguchi design uses its own design matrices and methods of analysis, which are not model-based. However, it is still possible to estimate process models on the basis of experiments planned according to Taguchi's method.

Appendix B Fermentation Process

B.1 White Model for the Fermentation Process

Consider a fed-batch fermentation process, as described in [OVE92]. In the reactor a micro-organisms grow for the production of penicillin. Sugar is used as the culture medium for bacteria.
The process is described by the following differential equations:

$$dMb(t)/dt = c_1(T(t)) \cdot Sd \tag{B.1}$$

$$dPf(t)/dt = Qp(T(t)) \cdot Mb(t) \tag{B.2}$$

with the initial conditions:

$$Mb(t=0) = 1.0 \tag{B.3}$$

$$Pf(t=0) = 0.0 \tag{B.4}$$

where:

$Mb(t)$	:	amount of the micro-organisms [kg]
$Pf(t)$	:	amount of the produced product [penicillin unit]
Sd	:	sugar dosage [kg/h]
$T(t)$	:	temperature as a function of time [°C]
$c_1(T(t))$	:	yield factor [-]
$Qp(T(t))$	:	specific production velocity [penicillin unit / kg · h]

$$c_1(T(t)) = -8.0 + 0.6T(t) - 0.01T^2(t) \tag{B.5}$$

$$Q_P(T(t)) = -3.0 + 0.4T(t) - 0.01T^2(t) \tag{B.6}$$

This model is used to simulate measurements, which are then used for estimation of black-box transition models. To make the simulation more realistic, with each measurement a random number is added from a normal distribution $\mathcal{N}(0,\sigma_m)$, with various σ_m for each measured response.

B.2 Dynamic Optimization of the Fermentation Process

Suppose, one is interested in the maximal fermentation yield. It is known that the temperature favourable for the growing of the micro-organisms is higher than the temperature favourable for product forming. But it is not known, however, what kind of temperature profile in the batch reactor has to be chosen for the optimal yield of this product.

The performance criterion J, which has to be maximized, with temperature T(t) as a control variable, is:

$$\max_{T(t)} J = \max_{T(t)} Pf(t_F = 10h) \tag{B.7}$$

Such an optimization problem is a special kind of the following optimal control problem:

$$\underset{\mathbf{u}(t)}{\text{maximize}} \quad \int_{t_0}^{t_F} f(\mathbf{x}(t),\mathbf{u}(t),t)dt + \Phi(\mathbf{x}(t_F)) \tag{B.8}$$

subject to:

$$d\mathbf{x}(t)/dt = \mathbf{g}(\mathbf{x}(t),\mathbf{u}(t),t) \tag{B.9}$$

$$\mathbf{x}(t_0) = \mathbf{x0} \tag{B.10}$$

$\mathbf{x}(t_F)$ to be chosen optimally

where:

$\mathbf{x}(t)$: vector of state variables;

$\mathbf{u}(t)$: vector of degrees of freedom for dynamic optimization (vector of control variables);

t : time, $t \in [t_0, t_F]$;

f : function;

$\mathbf{g}$: vector function with the right side of the differential equations (B.9)

Φ : final value function.

It can be solved by Maximum Principle of Pontryagin defining the necessary optimality conditions:

$$\partial \mathcal{H}(t)/\partial \mathbf{u} \quad = 0 \tag{B.11}$$
$$d\boldsymbol{\pi}(t)/dt \quad = -\partial \mathcal{H}(t)/\partial \mathbf{x} \tag{B.12}$$
$$\boldsymbol{\pi}(t_F) \quad = \partial \Phi/\partial \mathbf{x}(t_F) \tag{B.13}$$
$$d\mathbf{x}(t)/dt \quad = \partial \mathcal{H}(t)/\partial \boldsymbol{\lambda} \tag{B.14}$$

where:

$\mathcal{H}(t)$: Hamiltonian
$\boldsymbol{\pi}(t)$: adjoint variable vector

The adjoint-variable vector $\boldsymbol{\pi}(t)$ is used for the definition of the Hamiltonian $\mathcal{H}(t)$:

$$\mathcal{H}(t) = f(\mathbf{x}(t),\mathbf{u}(t),t) + \boldsymbol{\pi}^T(t)\cdot \mathbf{g}(\mathbf{x}(t),\mathbf{u}(t),t) \tag{B.15}$$

It can be mentioned that the adjoint-variable vector can be interpreted as the sensitivity of the objective function to changes in the state vector effected between moments of time *t* and *t+dt*.

$$\boldsymbol{\pi}(t) = \{ J(\mathbf{x}(t+dt)) + d\mathbf{x}(t+dt)) - J(\mathbf{x}(t))\} \ / \ d\mathbf{x}(t) \tag{B.16}$$

As presented in (B.13) the optimal choice for the final state value $\mathbf{x}(t_F)$ can be obtained by imposing end-point transversality conditions on the so-called adjoint variable vector $\boldsymbol{\pi}(t)$. When the final values of one or more state variables are prescribed, then the corresponding adjoint variables have no prescribed final value.

Going back to the optimization problem of the fermentation process, one can see that:

$$\mathbf{x}(t) = [Mb(t),\ Pf(t)\ ] \tag{B.17}$$
$$\mathbf{u}(t) = T(t) \tag{B.18}$$
$$f(\mathbf{x}(t),\mathbf{u}(t),t) = 0 \tag{B.19}$$
$$\Phi(\mathbf{x}(t_F)) = Pf(t_F) \tag{B.20}$$
$$g_1(t) = c_1(T(t)) \cdot Sd \tag{B.21}$$
$$g_2(t) = Qp(T(t)) \cdot Mb(t) \tag{B.22}$$

According to the Maximum Principle of Pontryagin, T*(t) is optimal when the following set of equations is satisfied:

$$0.6\pi_1(t) - 0.02\pi_1(t)\cdot T^*(t) - (0.4 - 0.02T^*(t))\cdot Mb(t) = 0 \quad (B.23)$$

$$\pi_2(t) = -1 \quad (B.24)$$

$$d\pi_1/dt = -3 + 0.4T^*(t) - 0.01T^{*2}(t) \quad (B.25)$$

$$dMb/dt = -8 + 0.6T^*(t) - 0.01T^{*2}(t) \quad (B.26)$$

$$dPf/dt = (-3 + 0.4T^*(t) - 0.01T^{*2}(t))\cdot Mb(t) \quad (B.27)$$

$$\pi_1(t_F) = 0 \quad (B.28)$$

$$Mb(0) = 1 \quad (B.29)$$

$$Pf(0) = 0 \quad (B.30)$$

Through substitution according to (B.23):

$$T^*(t) = (0.6\pi_1(t) - 0.4Mb(t)) / (0.02\pi_1(t) - 0.02Mb(t)) \quad (B.31)$$

one gets a system of three ordinary differential equations of π_1, Mb and Pf, which form the so-called two-point boundary-value problem.
Figure B1 shows the optimal state and control variables of the fermentation process found with Maximum Principle of Pontryagin.

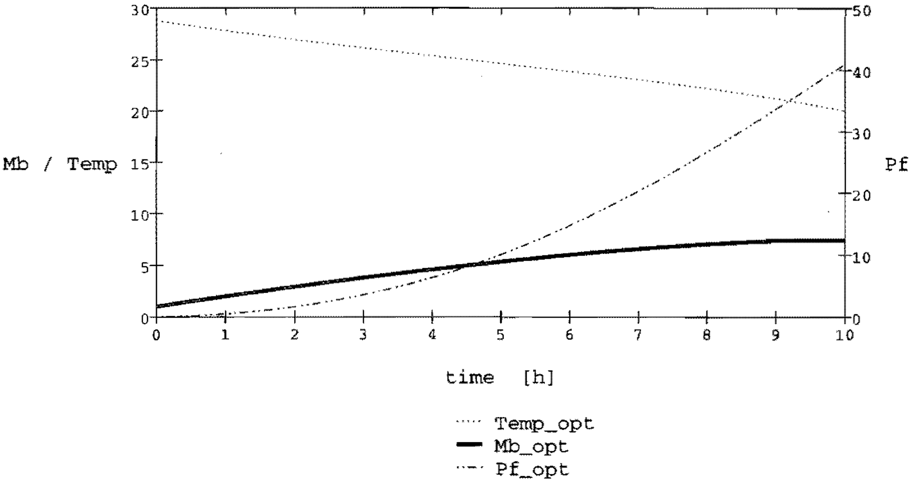


Figure B1 Optimal state and control variables of the fermentation process

Appendix C The Use of Statistical Tests During Process Modelling

C.1 Coefficients of Determination

The **coefficient of determination R²**, with values between zero and unity, measures the proportion of variance in the response explained by the model. It is defined as follows [MON91]:

$$R^2 = (SS_{TOTAL} - SS_{RESID}) \,/\, SS_{TOTAL} = SS_{REGR} \,/\, SS_{TOTAL} \qquad \text{(C.1)}$$

where:

SS_{TOTAL}	:	total sum of squares
SS_{REGR}	:	regression sum of squares
SS_{RESID}	:	residual sum of squares
y_i	:	measured response of the ith experiment
$\bar{y}$	:	mean of n measured responses
$\hat{y}_i$	:	estimated response of the ith experiment
n	:	number of experiments

$$SS_{TOTAL} = \sum_{i=1}^{n} (y_i - \bar{y})^2 = SS_{REGR} + SS_{RESID} \qquad \text{(C.2)}$$

$$SS_{REGR} = \sum_{i=1}^{n} (\bar{y} - \hat{y}_i)^2 \qquad \text{(C.3)}$$

$$SS_{RESID} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \qquad \text{(C.4)}$$

$$\bar{y} = 1/n \sum_{i=1}^{n} y_i \qquad \text{(C.5)}$$

A high value of R^2 indicates a good fit, and a low value a poor fit. But it should be emphasized that the value of R^2 increases as the number of model terms is increased. Therefore, in practice a compromise is needed between a R^2 and the number of model terms. For this purpose, the statistic R^2 is adjusted for the number of degrees of freedom. It results in the so-called **adjusted coefficient of determination R^2_{adj}**, defined as follows:

$$R^2_{adj} = (MS_{TOTAL} - MS_{RESID}) / MS_{TOTAL} = 1 - (1-R^2)(n-c) / (n-p) \quad (C.6)$$

where:

$MS_{TOTAL} = SS_{TOTAL} / (n-c)$: total mean square (C.7)
$MS_{RESID} = SS_{RESID} / (n-p)$: residual mean square (C.8)
$c =$: 1 if the model contains a constant term, otherwise 0
$p =$: number of model terms (including the constant term, if present)

Adding model terms decreases R^2_{adj}, so together with R^2, it provides a better judgement.

C.2 Significance of Model Coefficients

To test the significance of the model coefficient β_i, one has to calculate the t-value (in Table C1, which is the same as Table 3.2, it is denoted by "T-value") [DEM87]:

$$\text{t-value} = \beta_i / SE(\beta_i) \quad (C.9)$$

where:

$SE(\beta_i)$: standard error of the coefficient β_i (in Table C1 column 3).

If this calculated t-value is smaller than the tabulated critical t-value_{crit}, then the null hypothesis

$$H_0: \quad \beta_i \neq 0$$

can be rejected at the taken level of confidence. As the level of confidence for my statistical analysis I have taken 90%.

A number between zero and one in the 5th column of Table C1, denoted by "Signif", indicates the insignificance of the corresponding term, i.e. the risk that the given model estimate will be obtained when the true coefficient value were in fact zero. A low risk (smaller than 0.1) indicates that there is a reason to assume that the coefficient is not zero, so the term should be retained in the model; conversely a high risk (higher than 0.1) indicates that there is no reason to assume that the true coefficient is not zero, and so the term can be removed from the model.

Table C1 Model MOD1 of the response Pf (see Table 3.1)

```
   Least Squares Coefficients, Response Pf, Model MOD1

1 Term          2 Coeff.    3 Std. Error  4 T-value  5 Signif.  6 Transformed Term

-------------------------------------------------------------------------------------
1 1             37.313636      1.001673
2 ~Tb           -0.821250      1.174566                           ((Tb-2.85e+01)/1.5)
3 ~Ts           -1.338750      1.174566                           ((Ts+7.5e-01)/2.5e-01)
4 ~Sd            4.393750      1.174566                           ((Sd-1)/1.5e-01)
5 ~Tb*Ts        -4.181250      1.174566     -3.56      0.0236
6 ~Tb*Sd        -0.088750      1.174566     -0.08      0.9434
7 ~Ts*Sd        -0.086250      1.174566     -0.07      0.9450

No. cases = 11        R-sq. =  0.8768      RMS Error = 3.322
Resid. df =  4   R-sq-adj. =  0.6920          Cond. No. = 1
~  indicates  factors  are  transformed.
```

For example, in Table C1 the risk of the term Tb*Ts is 0.0236, which means that the calculated t-value of -3.56 is significant at the level (1 - 0.0236)*100% = 97.64%; in other words, the risk of falsely removing this term is at most 0.0236, therefore this term should be retained in the model.

C.3 Analysis of Variance

The investigation of sums of squares of residuals and resulting variances is called ANalysis Of VAriance (ANOVA) [BOX78]. Table C2 (the same as Table 3.4) gives an ANOVA example.

As defined before in (C.2), the total sum of squares (row 1 in Table C2) is partitioned into a component due to the regression (row 2 in Table C2) and a components due to the residuals (row 5 in Table C2). If there are higher-order terms in the model (in this case the interaction term Tb*Ts), the regression sum of squares is further partitioned into a component due to the linear model terms (row 3 in Table C2) and a component due to the nonlinear terms (row 4 in Table C2), which enables testing the non-linear effects separately. The F-ratio statistic is used to determine whether the non-linear terms, as a group, make a significant contribution to the model fit. In this case the F-ratio is 139.8628/ 7.3783 = 18.96. As one can see in the adjacent significance column such an F-ratio would occur in about 0.48 percent of the cases if the true fit were a linear model. As this seems a very rare event, it appears likely that the quadratic model fits significantly better than a linear one.

Table C2 The variance analysis of the model MOD1_RED (see Table 3.4)

```
  Least Squares Summary ANOVA, Response Pf, Model MOD1_RED

Source                            df     Sum Sq.     Mean Sq.   F-Ratio  Signif.
-------------------------------------------------------------------------------------
1 Total(Corr.)                    10   358.3067
2  Regression                      4   314.0368     78.5092     10.64    0.0068
3   Linear                         3   174.1739     58.0580      7.87    0.0168
4   Non-linear                     1   139.8628    139.8628     18.96    0.0048
5  Residual                        6    44.2699      7.3783
6   Lack of fit                    4    43.9810     10.9953     76.13    0.0130
7   Pure error                     2     0.2889      0.1444

                        R-sq. =  0.8764
                    R-sq-adj. =  0.7941
Model obeys hierarchy. The sum of squares for linear terms
is computed assuming nonlinear terms are first removed.
F(4,2) as large as 76.13 is a rare event =>
    likely that significant terms are missing from model.
```

If there are replicated observations (i.e. two or more experiments made at an identical set of factors values), the residual sum of squares can be further partitioned into a component due to lack of fit SS_{LOF} (row 6 in Table C2) and a component due to the replicated observations (pure error) SS_{PE} (row 7 in Table C2):

$$SS_{RESID} = SS_{LOF} + SS_{PE} \tag{C.10}$$

where:

SS_{PE} : pure error sum of squares
SS_{LOF} : lack of fit sum square
y_{jk} : kth response measurement for the jth experimental setting
$\bar{y}_j$: response mean for the jth experimental setting
$\hat{y}_j$: estimated response for the jth experimental setting

m : number of different experimental settings
r_j : number of repetitions at j^{th} of the m different experimental settings

$$SS_{PE} = \sum_{j=1}^{m} \sum_{k=1}^{r_j} (y_{jk} - \bar{y}_j)^2 \qquad (C.11)$$

$$SS_{LOF} = \sum_{j=1}^{m} r_j(\bar{y}_j - \hat{y}_j)^2 \qquad (C.12)$$

$$\bar{y}_j = 1/r_j \sum_{k=1}^{r_j} y_{jk} \qquad (C.13)$$

This makes it possible to test the lack of fit between the model and the data. Lack of fit is a measure of the discrepancy between the model prediction and the average of the replicated runs made at the set of m experimental conditions.

The so-called null hypothesis

$$H_0: \quad (SS_{LOF} / (m-p)) \overset{?}{=} (SS_{PE} / (n-m))$$

where:

n : number of experiments
p : number of model terms,

whether there is lack of fit, that is whether there may be terms missing from the model, is tested with the F statistic. This hypothesis will be not rejected when the significance is <u>larger</u> than 0.1. If the significance is between 0.05 and 0.1 I talk about some lack of fit.

It should be mentioned that the pure mean square error MS_{PE}, defined as follows:

$$MS_{PE} = \frac{SS_{PE}}{n - m} \qquad (C.14)$$

gives an estimate of the variance σ^2. See row 7 of Table C2.

C.4 Residual Analysis

If the model takes into account all factors that influence the response correctly, then the residuals for the model should show no discernible patterns (they should be randomly distributed). Any trends in the residuals generally indicate that one or more additional factors are not accounted for by the model. Residual analysis helps to decide whether there are any other factors or systematic errors influencing the response. The principal ways of plotting residuals are:

1. a histogram
2. in time sequence
3. versus fitted values
4. versus each factor
5. a probability plot

A **histogram** shows the frequency distribution of the residuals. In the ideal situation, the peak of the histogram should be at zero and it should be symmetric. If the histogram is extremely skewed a transformation of the response might be necessary to ensure the regression assumption about normal distribution of the random error ϵ.
However, with small number of cases, considerable deviations from a normal distribution often occur and does not necessarily imply a serious violation of the regression assumption.
If some outliers, that is atypical observations with residuals lying three or more standard deviations from the mean, are detected, a robust regression method might be recommended. This method gives less weight to outliers.

Plot of residuals in time sequence might be helpful in detecting a correlation between residuals, that is whether or not the residuals are dependent upon the time or sequence of the experiments. If the pattern of the residuals is not constant, that may mean that the process is "drifting", and that maybe a linear or higher-order term of time may improve the model.

Plot of residuals versus fitted values gives, like a histogram, an indication to use weighted least square, to transform the response or to add some extra (e.g. constant or higher-order) terms in the model. If the model is correct and the regression assumptions are satisfied, this plot should not reveal any obvious pattern.

Plot of residuals versus each factor is not often used when more than two factors are involved, but it can also give an indication about response transformation or to add some terms to the model. Once again an overall impression of a horizontal band of residuals is regarded as satisfactory.

To sum up, for the last three types of residuals' plots abnormality would be indicated if:

a) plot of residuals would look like a megaphone, what means that the variance of the observations increases with time or with the increasing magnitude of the response or of the factor; this indicates need for using weighted least squares;

b) plot of residuals would look like a rising band, that is negative residuals are mainly present in the first part of the plot and positive residuals in the second part; this indicates an error in analysis: a linear term in time or a constant should have been included in the model, or linear effect of a factor should have been removed;

c) plot of residuals has would look like an arc; this indicated a model inadequacy: quadratic and/or cross-product terms should have been included in the model, or the observations should have been transformed before analysis. For more discussion of transformation refer see e.g. [MON91].

A probability plot (the residuals are plotted on a cumulative normal probability scale) is constructed in such a way that if the residuals are normally distributed, the points that follow on the graph should lie close to a straight line. This makes it easy to detect (positive and negative) outliers and that systematic effects are overlooked. In the latter case, a break in the middle of the plot is visible.

C.5 Goodness of Fit for a Normal Distribution with χ^2 Test

Because the residual analysis does not tell the whole story, χ^2 test may be used to check if a sample distribution fits the frequencies to be expected given a normal population distribution [HAY94]. To use a sample of data to test the hypothesis that the population has a normal distribution, one first must decide on a number of class intervals of some given size and then think of both, the sample and the population, as divided into these class intervals. Next, the following value of χ^2 statistic should be computed:

$$\chi^2 = \sum_{k=1}^{K} \frac{(n_k - n \cdot p_k)^2}{n \cdot p_k} \qquad (C.15)$$

where:

K : number of distinct categories
n_k : number of experiments for category k; $n_k \geq 8$
n : number of experiments; $n = \sum n_k$
p_k : the relative frequency for category k as dictated by the normal distribution
$n \cdot p_k$: the expected frequency in category k.

Given that the null hypothesis is true, the sample distribution is approximately χ^2 with K-2-1 degrees of freedom, if the mean and the standard deviation of the population are not known, but they are estimated.
If the calculated value (C.15) is larger than the tabulated critical value χ^2_{crit} for the chosen level of confidence, e.g. 90%, then the null hypothesis should be rejected.

C.6 PRESS-Test

Another, often used, procedure to check a regression model is the so-called PRESS (PREdiction Sum of Squares), which gives a measure of the prediction ability of the model [BOX87]. One obtains the sum of squares of discrepancies, defined as in (C.16), for a given model by leaving out one observation, fitting a given model to the rest of the data, then predicting the one left out and obtaining the square of the discrepancy, and then repeating this for all other omissions:

$$\mathrm{PRESS} = \sum_{i=1}^{n}(y_i - \hat{y}_{model\backslash i})^2 \tag{C.16}$$

where:

y_i : measured response of the ith experiment
$\hat{y}_{model\backslash i}$: response of the ith experiment predicted by a model estimated without ith experiment
n : number of experiments.

Often, in place of PRESS-value, a statistic R^2_{PRESS} defined as in (C.17) is used.

$$R^2_{PRESS} = (SS_{TOTAL} - \mathrm{PRESS}) \,/\, SS_{TOTAL} \tag{C.17}$$

The highest value of this statistic (the maximal value is 1) means the best prediction ability of the model.

Appendix D More About the Nelder-Mead Simplex Method

The Nelder-Mead procedure starts with the computation of the reflected vertex $\mathbf{x}^r$ as defined in Section 3.5.1. If $\mathbf{x}^r$ gives better results than $\mathbf{x}_{w-1}^N$ and worse than $\mathbf{x}_b$, then $\mathbf{x}^r$ replaces $\mathbf{x}_w^N$ as a vertex in the new, $N+1^{th}$, simplex. The remaining vertices are retained. If $\mathbf{x}^r$ gives better results than $\mathbf{x}_b^N$, then the procedure tries to continue the search in the same direction by computing the expanded vertex $\mathbf{x}^e$. It should be emphasized that the expansion will also be accepted when $\mathbf{x}^e$ is worse than $\mathbf{x}^r$. This decision is in accordance however with the idea of the procedure: first the simplex should enclose the optimum and later, by means of shrinkage, approximate the optimum.

The operation of contraction occurs when the reflected vertex $\mathbf{x}^r$ is a worse approximation to a solution than $\mathbf{x}_w^N$. The shadow contraction is applied when $\mathbf{x}^r$ is better than $\mathbf{x}_w^N$ but worse than $\mathbf{x}_{w-1}^N$. If the vertex after contraction of shadow contraction is worse than $\mathbf{x}_w^N$, then the shrinkage operation starts. This operation reduces the size of the simplex by moving all vertices except $\mathbf{x}_b$ towards the best vertex.
When a new $N+1^{th}$ simplex is generated in this manner, again a triplet ($\mathbf{x}_b^{N+1}$, $\mathbf{x}_{w-1}^{N+1}$, $\mathbf{x}_w^{N+1}$) is selected from all vertices and the procedure is repeated until the optimum is reached.

Because usually some constraints on recipe items are present, the presented method must be extended. Nelder and Mead proposed the so-called penalty method to handle constraints: in the case where the proposed new simplex vertex $\mathbf{x}^e$ or $\mathbf{x}^r$ is outside the allowable area, an extremely high penalty will be given for this point. As a consequence the questions 1 ÷ 4 of Figure D1 will be answered with "no" and the unfeasible point will be replaced by $\mathbf{x}^r$ or $\mathbf{x}^c$, respectively. In this manner linear as well as non-linear constraints on factors can be handled. Although the method is safe in the sense that the simplex can never leave the allowable bounded factor area, the danger is, however, that when proceeding in this manner, the simplex will be shrunk unnecessarily and the procedure will stop prematurely instead of searching further along the constraint.

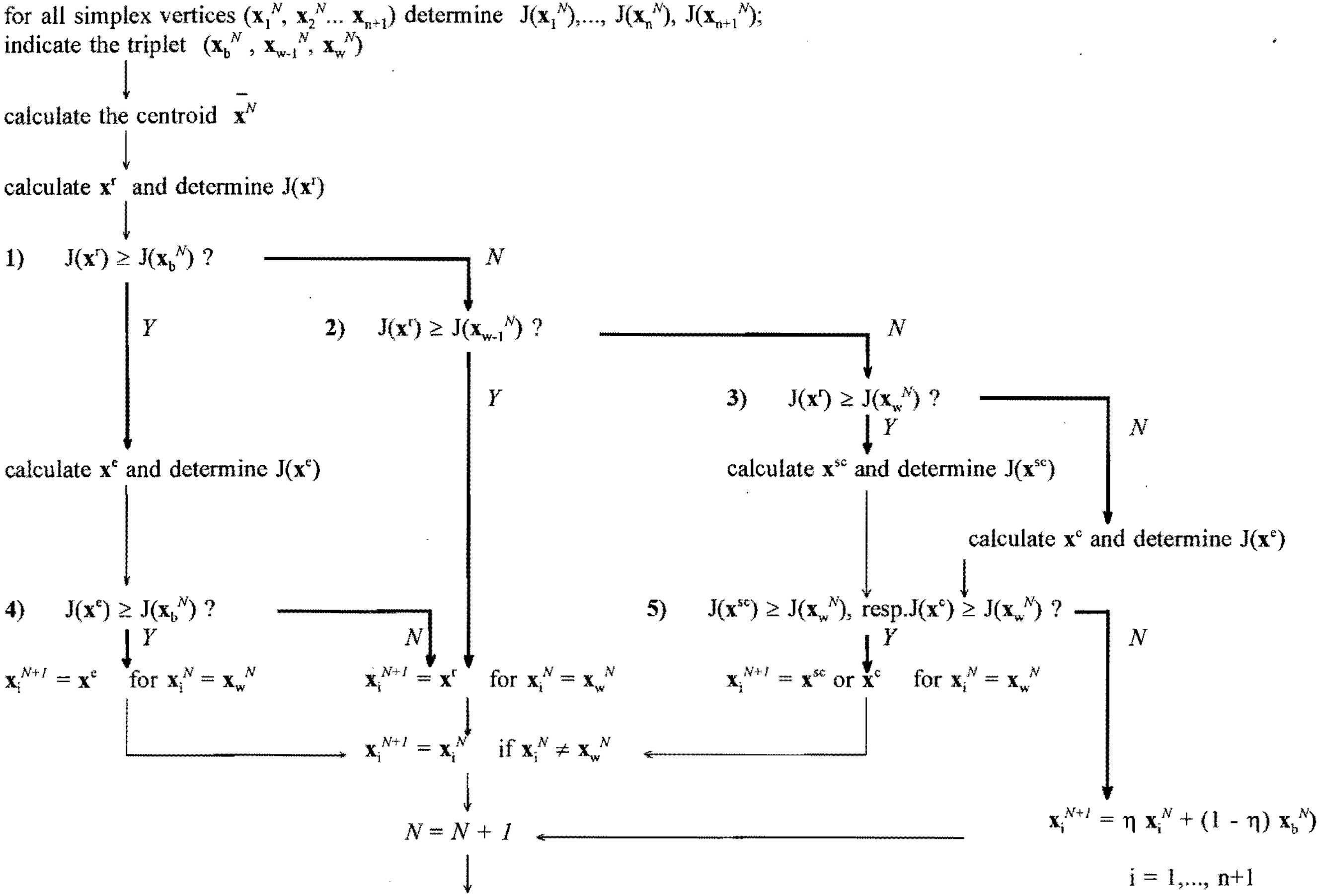

for all simplex vertices $(\mathbf{x}_1^N, \mathbf{x}_2^N ... \mathbf{x}_{n+1})$ determine $J(\mathbf{x}_1^N),..., J(\mathbf{x}_n^N), J(\mathbf{x}_{n+1}^N)$;
indicate the triplet $(\mathbf{x}_b^N, \mathbf{x}_{w-1}^N, \mathbf{x}_w^N)$
calculate the centroid $\bar{\mathbf{x}}^N$
calculate $\mathbf{x}^r$ and determine $J(\mathbf{x}^r)$
1) $J(\mathbf{x}^r) \geq J(\mathbf{x}_b^N)$?
N
Y
2) $J(\mathbf{x}^r) \geq J(\mathbf{x}_{w-1}^N)$?
N
Y
3) $J(\mathbf{x}^r) \geq J(\mathbf{x}_w^N)$?
Y
N
calculate $\mathbf{x}^e$ and determine $J(\mathbf{x}^e)$
calculate $\mathbf{x}^{sc}$ and determine $J(\mathbf{x}^{sc})$
calculate $\mathbf{x}^c$ and determine $J(\mathbf{x}^e)$
4) $J(\mathbf{x}^e) \geq J(\mathbf{x}_b^N)$?
Y
N
5) $J(\mathbf{x}^{sc}) \geq J(\mathbf{x}_w^N)$, resp.$J(\mathbf{x}^c) \geq J(\mathbf{x}_w^N)$?
Y
N
$\mathbf{x}_i^{N+1} = \mathbf{x}^e$ for $\mathbf{x}_i^N = \mathbf{x}_w^N$
$\mathbf{x}_i^{N+1} = \mathbf{x}^r$ for $\mathbf{x}_i^N = \mathbf{x}_w^N$
$\mathbf{x}_i^{N+1} = \mathbf{x}^{sc}$ or $\mathbf{x}^c$ for $\mathbf{x}_i^N = \mathbf{x}_w^N$
$\mathbf{x}_i^{N+1} = \mathbf{x}_i^N$ if $\mathbf{x}_i^N \neq \mathbf{x}_w^N$
$\mathbf{x}_i^{N+1} = \eta\, \mathbf{x}_i^N + (1 - \eta)\, \mathbf{x}_b^N)$
$i = 1,..., n+1$
$N = N + 1$

Figure D1 Flow diagram describing possible operations during one N^{th}iteration of the Nelder-Mead simplex method. Decision flows are presented with bold arrows.

In the FRIS-approach, in the case $\mathbf{x}^r$ is not feasible, I propose to use an additional operation, namely shadow contraction repeated maximally five times. The first time shadow contraction starts with $\delta = 1/2$ (see Definition (3.47)) and next, depending whether $\mathbf{x}^{sc}(\delta)$ is feasible or not, the procedure tries by enlarging or reducing this coefficient to find δ_{good}, so that $\mathbf{x}^{sc}(\delta_{good})$ lies in the allowable area, but as close as possible to the constraint. When no $\delta_{good} < 0.99$ can be found, that is: no feasible point $\mathbf{x}^{sc}$ can be found, the contraction point inside the current simplex will replace the worst vertex. This procedure works as follows:

```
(j)    set:  $\delta_{low} = 0$;   $\delta_{up} = 1$;   $\delta_{good} = 1$;   $\delta = 1/2$;   j = 0;
(jj)   if $\mathbf{x}^{sc}(\delta)$ feasible then
           $\delta_{good} = \delta$;   $\delta_{up} = \delta$;   $\delta = (\delta_{low} + \delta) / 2$;
       else
           $\delta_{low} = \delta$;   $\delta = (\delta_{up} + \delta) / 2$;
       endif;
       j = j + 1;
       repeat (jj) until j = 5.
(jjj)  if $\delta_{good} > 0.99$, then replace $\mathbf{x}^{sc}$ by $\mathbf{x}^c$
       else accept $\mathbf{x}^{sc}( \delta_{good})$ endif.
```

Implemented in the FRIS package, the whole Nelder - Mead simplex algorithm works as follows:

```
(o)    set $N = 0$;
       choose the start simplex $(\mathbf{x}_1^N, \dots, \mathbf{x}_n^N, \mathbf{x}_{n+1}^N)$;
(i)    determine $J(\mathbf{x}_1^N), \dots, J(\mathbf{x}_n^N), J(\mathbf{x}_{n+1}^N)$;
       from the simplex vertices select a triplet $(\mathbf{x}_b^N, \mathbf{x}_{w-1}^N, \mathbf{x}_w^N)$;
(ii)   compute the centroid $\mathbf{P}^N$;
(iii)  if  $N = 0$ then go to (iv);
       else check stopping criteria and if they are satisfied exit
       endif;
(iv)   perform reflection; if $\mathbf{x}^r$ not feasible, then apply (j) ÷ (jjj) endif;
(v)    according to Figure D1 accept expansion, reflection, contraction,
       shadow contraction or perform shrinkage;
```

```
        if $\mathbf{x}^e$ is not feasible, then expansion is skipped and $\mathbf{x}^r$ is accepted
        endif;
(vi)    if shrinkage is performed, then set $\mathbf{x}_i^{N+1} = \eta\ \mathbf{x}_i^N + (1 - \eta)\ \mathbf{x}_b^N$, i = 1,..., n+1
        else set:
            $\mathbf{x}_i^{N+1} = \mathbf{x}_i^N$                                for $\mathbf{x}_i^N \neq \mathbf{x}_w$ , i = 1,..., n+1

            $\mathbf{x}_i^{N+1} = \mathbf{x}^r$   or   $\mathbf{x}_i^{N+1} = \mathbf{x}^e$,   or   $\mathbf{x}_i^{N+1} = \mathbf{x}^c$,   or   $\mathbf{x}_i^{N+1} = \mathbf{x}^{sc}$

                                                     for $\mathbf{x}_i^N = \mathbf{x}_w$ , i = 1,..., n+1
        endif;
(vii)   set $N = N+1$ and go to (i), unless $N \geq Nmax$.
```

Up till now nothing has been said about the stopping criteria of the algorithm. The stopping criterion proposed by Nelder and Mead for function optimization (so: not for empirical optimization) can be presented as follows:

$$1/n+1 \sum_{i=1}^{n+1} \left(J(\mathbf{x}_i^N) \ - 1/n+1 \sum_{i=1}^{n+1} J(\mathbf{x}_i^N) \right)^2 \leq crit3 \tag{D.1}$$

where:

crit3 > 0 i : some preset value.

Thus, Nelder and Mead proposed to stop the function optimization when the variance of the criterion is less than the preset value *crit3*.

However, there are certain cases where the stopping criterion of Nelder Mead is not sufficient, even for function optimization. If all criterion values are close enough, then the algorithm will stop. This may also occur when the simplex is very large. Therefore, I use additional stopping criteria for function optimization, that is:

$$1/n+1 \left| \sum_{i=1}^{n+1} \left(J(\mathbf{x}_i^{N-1}) - J(\mathbf{x}_i^N) \right) \right| \leq crit4 \tag{D.2}$$

$$x_{i,\,MAX} - x_{i,\,MIN} \leq crit5_i, \qquad i=1,...,n \tag{D.3}$$

where:

$x_{i,\,MAX}$, $x_{i,\,MIN}$: maximal, respectively minimal value of the ith factor in the simplex

$crit4 \geq 0$: some preset value

$crit5_i \geq 0$: some preset value, $i=1,\ldots,n$.

The second criterion (D.2) means that the mean function value of the previous and the last simplex does not change more than *crit4*. The criterion (D.3) checkes for each factor whether its range in the simplex is smaller than preset value *crit3*$_i$.

None of these criteria individually guarantees convergence to a local optimum. For this reason all criteria (D.1 ÷ D.3) must be satisfied simultaneously. However, this is only true for noise-free function optimization. In the case of experimental optimization the criteria (D.1) and (D.2) might take an experimental error into consideration too. Each criterion of the set (D.3) is independent of any response measurement and therefore our approach starts with checking them once after the other. Only when all criteria (D.3) are satisfied, the test-runs of the last simplex are repeated to estimate .the experimental error. After the experimental error is estimated the following stopping criteria corresponding with (D.1) and (D.2) are checked:

$$1/n+1 \sum_{i=1}^{n+1} (\bar{J}((\mathbf{x}_i) \; - 1/n+1 \sum_{i=1}^{n+1} \bar{J}((\mathbf{x}_i) \;)^2 \leq \mathit{crit3} + 4\zeta^2$$

$$+ \; 4\zeta/n+1 \sum_{i=1}^{n+1} | \; \bar{J}((\mathbf{x}_i) \; - 1/n+1 \sum_{i=1}^{n+1} \bar{J}((\mathbf{x}_i) \; | \qquad \text{(D.1')}$$

$$1/n+1 \; \Big| \sum_{i=1}^{n+1} (\; \bar{J}((\mathbf{x}_i^{N-1}) - \bar{J}((\mathbf{x}_i^{N}) \;) \Big| \; \leq \mathit{crit4} + 2\zeta \qquad \text{(D.2')}$$

where:

$\bar{J}(\mathbf{x}_i)$: mean of the criterion values for the vertex $\mathbf{x}_i$

ζ : number corresponding to the experimental error

ζ = $\$_Y^T \boldsymbol{\sigma} / d$ (D.4)

d : denominator of (3.26)

σ : standard deviation of the experimental error computed on the basis of replications, i.e. test-run repetitions

$$\sigma^2 = \frac{\sum_{j=1}^{n+1} \sum_{k=1}^{r_j} (y_{jk} - \bar{y}_j)^2}{(\sum_{j=1}^{n+1} r_j) - (n + 1)} \qquad \text{(D.5)}$$

y_{jk} : k^{th} measurement of the response y at the j^{th} vertex of the simplex

$\bar{y}_j = 1/r_j \sum_{k=1}^{r_j} y_{jk}$: response mean for the j^{th} vertex of the simplex

r_j : number of test-run repetitions at the j^{th} vertex of the simplex.

Of course, the user of the algorithm can decide to stop before all criteria (D.3, D.1', D.2') are satisfied. They can be seen as a mathematical support, while in the practice, depending on the application, it might be meaningful to stop earlier, when improvement is esteemed to be satisfactory enough, and the expected next test runs are too costly to perform.

Appendix E More About Line Optimization

The direction of steepest ascent is the direction of the gradient of the performance criterion and the search along this direction is called line optimization. This procedure searches for the step size α_{OPT} that maximizes the performance criterion along the path of steepest ascent.

In general, the line optimization procedure is based on one of two types of methods, which are:

1) function comparison methods (like Golden Section, Fibonacci and Bisection search);
2) polynomial interpolation methods (like quadratic or cubic interpolation) [SCA85].

The first group of methods evaluates the performance criterion at points within the interval containing a local optimum to narrow the interval around the optimum, thus enclosing the optimum ever more closely.

The second group of methods approximates the performance criterion by a quadratic or cubic polynomial in an interval known to contain the optimum. The optimum of the polynomial is then used to predict the location of maximum of the performance criterion.

For easy of discussion of a line optimization used by the FRIS-approach, the performance criterion will be temporarily written as follows:

$$J(\alpha) = J(\mathbf{x} + \alpha\mathbf{p}) \qquad \text{(E.1)}$$

where:

$\mathbf{x}$: varied recipe items; here optimization variables
α : step size
$\mathbf{p} = \nabla J(\mathbf{x})$: gradient of J; the search direction.

In the FRIS-approach the starting point for the line search is $J(\alpha_1=0)$, which is the optimal point at the validity boundary of the already estimated process model. Next, two points on the line must be evaluated, that is two test runs corresponding with $J(\alpha_2)$ and $J(\alpha_3)$ must be performed. In the beginning of the procedure one may choose

$$\alpha_3 - \alpha_2 = \alpha_2 - \alpha_1. \qquad \text{(E.2)}$$

The size of these steps can be determined depending on the process situation, i.e. it must be worked out how far from the previous boundary test runs can be run.

Three different cases may be distinguished:

i) if $J(\alpha_1) \leq J(\alpha_2)$ and $J(\alpha_2) \leq J(\alpha_3)$, then a further step $\Delta\alpha$ in the gradient direction is taken and a new test run corresponding to $\mathbf{x} + (\alpha_3 + \Delta\alpha)\mathbf{p}$ must be performed to determine a new

$$J(\alpha_3) = J(\alpha_3 + \Delta\alpha). \qquad \text{(E.3)}$$

The old $J(\alpha_3)$ and $J(\alpha_2)$ now become $J(\alpha_2)$ and $J(\alpha_1)$, respectively. The relation between the last three test runs is evaluated again. If the process situation allows, the search interval $[\alpha_1,\alpha_2]$ is shifted with a step size that preferably grows exponentially. If a hard boundary is reached, which in no case may be exceeded, $\mathbf{x} + \alpha_3\mathbf{p}$ lies on or just before the corresponding hard boundary and denotes the process conditions for the next and last test run. Then the curvature of the parabola through $J(\alpha_1)$, $J(\alpha_2)$ and $J(\alpha_3)$ is calculated. If the curvature is negative, quadratic interpolation gives α_{OPT} or else $J(\alpha_3)$ is excepted as the line maximum.

ii) if $J(\alpha_1) < J(\alpha_2)$ and $J(\alpha_2) > J(\alpha_3)$ quadratic interpolation approximates the optimum $J(\alpha_{OPT})$.
We have chosen here for quadratic and not for cubic interpolation to keep the number of the criterion evaluations, i.e. test runs, as small as possible.

iii) if neither *i)* and *ii)* is the case, a function comparison method from the group 1) can be applied either to the interval $[\alpha_1,\alpha_2]$, if $J(\alpha_1) > J(\alpha_3)$, or to the interval $[\alpha_2,\alpha_3]$, if $J(\alpha_1) < J(\alpha_3)$.

We have chosen here the Golden Section search method.
Golden Section is a geometrical proportion in which a line is divided, so that the ratio of the length of the longer line segment to the length of the entire line is equal to the ratio of the length of the shorter line segment to the length of the longer one. A Golden Section is created by the point C on line segment AB if AC/AB = CB/AC. This ratio has the numerical value ≈ 0.618, that is also the convergence ratio of this method.

> The main advantage of the Golden Section search method in comparison with Fibonacci search is that the number of iterations does not have to be determined in advance and the procedure can be terminated by the user at any moment. The Golden Section method is more rapid than Bisection; the convergence ratio of the latter is 0.5 only.

When the line optimum is found, a new experiment design task may be defined around ($\mathbf{x} + \alpha_{OPT}\ \mathbf{p}$) to estimate a new process model and the optimization procedure can be repeated. Also here, the user must decide when the improvement is satisfactory and the optimization procedure may be concluded.

Appendix F Multi-Objective Methods Based on a Measure of Optimality

F.1 The Weighting Method

This method assumes that there a set of non-negative, not all zero weights, $\omega_1, \omega_2, ..., \omega_k$ is given which represent the relative importance of the individual objectives. Then, the multi-objective optimization can be reduced to the following single-objective problem:

$$\max_{\mathbf{x} \in X} \sum_{i=1}^{k} \omega_i \, J_i(\mathbf{x}) \tag{F.1}$$

Obviously, the solution obtained in this manner depends on the choice of the weighting factors, and in many situations the decision maker may not be able specify them a priori. This is an essential drawback of the method.

F.2 The Lexicographic Method

This method assumes that the individual objectives may be ranked by their importance, so that a sequential optimization of the ordered set of single criterion is possible.

In the beginning, the decision maker has to provide a ranking of the individual objectives by giving a priority list $J_1, J_2, ..., J_k$ in decreasing importance. Next, in the first step he optimizes the most important objective:

$$\max_{\mathbf{x} \in X} J_1(\mathbf{x}) \tag{F.2}$$

Furthermore he/she solves the following problem at the sth step:

$$\max_{\mathbf{x} \in X} J_s(\mathbf{x}) \quad \text{subject to} \quad J_j(\mathbf{x}) = \zeta_j \tag{F.3}$$

where:

ζ_j : optimal value of J_j, $j = 1,...,s-1$, $s = 2,...,k$

The disadvantage of using this method is that the procedure can terminate before all, less important, objectives are optimized and, as a consequence, they can have unfavourable values. This disadvantage can be corrected by a relaxation of optimal values ζ_j and the sequential optimization (F.3) can be replaced by:

$$\max_{x \in X} J_s(\mathbf{x}) \quad \text{subject to} \quad J_j(\mathbf{x}) = \zeta_j - \rho_j \qquad j = 1,...,s-1 \tag{F.4}$$

where:

$\rho_j > 0$: given number, which is the relaxation level in the objective J_j.

The question remains how to choose these relaxation levels.

F.3 The ϵ-Constraint Method

This method replaces multi-objective optimization by optimizing only one objective with the greatest preference, say J_1. For $j \neq 1$, real numbers, ε_j, are given which are considered to be lower bounds for the objective J_j such that the decision maker does not accept any solution with a lower value than ε_j in the objective J_j. Therefore the following single optimization problem must be solved:

$$\max_{x \in X} J_1\,(\mathbf{x}) \quad \text{subject to} \; J_j(\mathbf{x}) \geq \varepsilon_j, \qquad j = 2,...,k \tag{F.5}$$

Here, if there is no feasible solution of this problem, it means that the bounds, or minimum requirements, ε_j are too high. Consequently at least one of them must be relaxed. Again, the problem is how to choose them in a proper way.

References

[ATK82]
Atkinson, A.C.,
Developments in the Design of Experiments,
International Statistical Review 50, pp 161-177, 1982.

[BAN94]
Banens, P.J.A., *et al.*,
Industriele Statistiek en Kwaliteit (in Dutch),
Kluwer Bedrijfswetenschappen, Deventer, 1994.

[BEE94]
van Beek, A.M.,
Multiplexfitting: Verder Onderzoek naar een Methode voor Experimentele Optimalisatie (in Dutch), Internal Report NR-1890, Eindhoven University of Technology, 1994.

[BOX78]
Box, G.E.P., W.G. Hunter, J.S. Hunter
Statistics for Experiments,
Wiley, New York, 1978.

[BOX87]
Box, G.E.P., N.R. Draper,
Empirical Model-Building and Response Surfaces,
Wiley, New York, 1987.

[BRU95]
Bruno, G.,
Model-Based Software Engineering,
Chapman & Hall, 1995.

[CHA83]
Changkong,V., Y.Y. Haimes,
Multiobjective Decision Making: Theory and Methodology,
North Holland, Amsterdam, 1983.

[DEM87]
Deming, S.N., S.L. Morgan,
Experimental Design: a Chemometric Approach,
Elsevier, Amsterdam, 1987.

[DRA81]
Draper, N.R., H. Smith,
Applied Regression Analysis,
Wiley, New York, 1981.

[FIS90]
Fisher, G.T.,
Batch Control Systems: Design, Application and Implementation,
ISA, North Carolina, 1990.

[HAY94]
Hays, W.L.,
Statictics,
Harcourt Brace, Forth Worth, 1994.

[HES66]
Hestenes, M.R.,
Calculus of Variations and Optimal Control Theory,
Wiley, New York, 1966.

[ING82]
Ingnizio, J.P.,
Linear Programming in Single- and Multiple Objective Systems,
Prentice-Hall, Englewood Cliffs, New York, 1982.

[ISA95]
ISA Standard ISA-S88.01-1995,
Batch Control. Models and Terminology,
ISA, 1995.

[JAL91]
Jalote, P.
An Integrated Approach to Software Engineering,
Sringer-Verlag, New York, 1991.

[JUB86]
Juba, M.R., J.W. Hamer,
Progress and Challenges in Batch Process Control,
Chemical Process Control - CPC III (Eds. M. Morari, T. McAvoy),
CACHE, 1986.

[KEA91]
Keats, J.B., D.C. Montgomery,
Statistical Process Control in Manufacturing,
Dekker, Basel, 1991.

[KEE93]
Keesman, K.J.,
Application of Flexible Recipes for Model Building, Batch Process Optimization and Control,
AIChE Journal 39(4), pp 581-588, 1993.

[KOO95]
Kooijmans, R.A.G.,
De Toepassing van FRIS in de Harsproducerende Industrie (in Dutch),
Internal Report NR-1898 (confidential), Eindhoven University of Technology, 1995.

[KOT82]
Kotarbinski, T.,
Traktrat o Dobrej Robocie (in Polish),
7th ed., Ossolineum, Wroclaw, 1982.
Translated into English as *Praxiology*, 1965.

[KRA77]
Kramer, N.J.T.A., J. de Smit,
Systems Thinking: Concepts and Notations,
Nijhoff, Leiden 1977.

[LAW94]
Lawrence C., *et al.*,
User's Guide for CFSQP Version 2.1: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisying All Inequality Constraints,
Internal Report TR-94-16r1, University of Maryland, 1994.

[LEE82]
Lee, H., K. Neville,
Handbook of Epoxy Resins,
McGraw-Hill, Inc., New York, 1982.

[LEO92]
Leonard, D., N. van Long,
Optimal Control Theory and Static Optimization in Economics,
Cambridge University Press, Cambridge, 1992.

[LIN90]
Linden, van der, R.J.P.,
Design and Application of PRIMAL: a Package for Experimental Modelling of Industrial Processes,
PhD Thesis, Eindhoven University of Technology, 1990.

[MAN94]
Manly, B.F.J.,
Multivariate Statistical Methods,
Chapman & Hall, London, 1994.

[MAR79]
De Marco, T.,
Structured Analysis and System Specification,
Prentice Hall, Englewood Cliffs, 1979

[MEI94]
Meijer, M.,
Experimentontwerp voor Tijdsafhankelike Variabelen (in Dutch),
Internal Report NR-1872, Eindhoven University of Technology, 1994.

[MON91]
Montgomery, D.C.,
Design and Analysis of Experiments,
Wiley, New York 1991.

[NEL65]
Nelder, J.A., R. Mead,
A Simplex Method for Function Optimization,
The Computer Journal 7, pp 308-313, 1965.

[OKU91]
Okumara, Y., *et al.*,
Graded-Junction Gate/n- Overlapped LDD MOSFET Structures for High Hot Carrier Reliability,
IEEE Transactions on Electron Devices 38, 1991.

[OTT93]
Otten, J.A.M. *et al.*,
Improvement of LDD MOSFET's Characteristics an Inverse Modelling Technique for Device and Process Optimization,
International Semiconductor Device Research Symposium, 1993.

[OTT95]
Otten, J.A.M.,
Measuring and Modelling of Series Resistance in Submicron MOSFETs,
PhD. Thesis, Eindhoven University of Technology, 1995.

[OVE92]
Overwater, A.J.,
Maximum Principe Pontryagin Nuttig bij Optimalisatie Batchprocessen (in Dutch), ProcesTechnologie 4, pp 31-35, 1992.

[PAR06]
Pareto, V.,
Manuale di Economia Politica,
Societa Editrice Libraria, Milano, 1906.
Translated into English by A.S. Schwier and A. Page as *Manual of Political Economy*,
Macmillan, New York, 1972.

[PAT62]
Patton, T.C.,
Alkyd Resin Technology. Formulating Techniques and Allied Calculations,
Interscience Publishers, New York, 1962.

[PLA46]
Plackett, R.L., J.P. Burman,
The Design of Optimum Multifactor Experiments,
Biometrica 33, pp 305-325, 1946.

[RAD95]
Rademaker, O.,
Computer-Based Recipe Development in the Process Industry,
Final Report STW-Project TST82/88.1620, NR-1897, Eindhoven University of Technology, 1995.

[REN95]
Renes, W.,
FRIS-Prototype User Manual,
TNO/TPD Report, 1995.

[RIJ91]
Rijnsdorp, J.E.,
Integrated Process Control and Automation, Ch. 11-12,
Elsevier, Amsterdam, 1991.

[RIJ92]
Rijnsdorp, J.E.,
Control of Batch Processes in a Dynamic Environment,
Journal A 33(1), pp 102-107, 1992.

[ROS87]
Rosenof, H.P., A. Ghosh,
Batch Process Automation,
van Nodstrand Reinhold,Wokingham, 1987.

[SCA85]
Scales, L.E.,
Introduction to Non-linear Optimization,
Macmillan Publisher Ltd, London, 1985.

[SEL90]
Selow, E.R.,
Flexibele Receptuuraanpassing voor de Batchgewijze Productie van Benzylalcohol. Modelontwikkeling en Procesoptimalisatie,
MSc. Thesis, University of Twente, 1990.

[SME95]
Smeets, J.F.C,
Toepassing van Flexibele Recepten bij de Ontwikkeling van Poederharsen (in Dutch), MSc. Thesis NR-1911, Eindhoven University of Technology, 1995.

[SPE62]
Spendly, W., *et al.*,
Sequential Application of Simplex Design in Optimization and Evolutionary Operation, Technometrics 4(4), pp 441-450, 1962.

[STE92]
Steeghs, L.,
Receptcorrectie van Batchprocessen (in Dutch),
MSc. Thesis NR-1799, Eindhoven University of Technology, 1992.

[TAG87]
Taguchi, G.,
System of Experimental Design,
ASI, Dearborn, Michigan, 1987.

[THI94]
Thielens, G.
Receptcorrrectie van een Batchproces op Basis van Witte Modellen (in Dutch),
Internal Report NR-1851, Eindhoven University of Technology, 1994.

[UPP93]
Upperman, P.M.,
Designs with a Small Number of Runs for Factorial Experiments,
PhD. Thesis, Eindhoven University of Technology, 1993.

[VAN93]
Van Impe, J.F.,
Modelling and Optimal Adaptive Control of Biotechnological Processes,
PhD. Thesis, Katolieke Universiteit Leuven, 1993.

[VERD95]
Verdellen, G.,
Implementatie Receptinitialisatie als PRIMACS-Module (in Dutch),
Final Thesis report NR-1895, Eindhoven University of Technology, 1995.

[VERH95]
Verhofdstad, F.,
PRIMACS Application Development Manual,
TNO/TPD Report, 1995.

[VER93]
Verwater-Lukszo, Z., O.Rademaker,
Improving Batch Production with Flexible Recipes,
IFAC Workshop on Production Control in Process Industry, Düsseldorf (G), March 29-31, 1993.

[VER94a]
Verwater - Lukszo, Z., *et. al.,*
Recipe Initialization for a Batch Process,
IFAC Conference on Integrated Systems Engineers, Baden-Baden (G), September 27-29, 1994a.

[VER94b]
Verwater-Lukszo, Z., G. Otten,
A Novel Model-Based Approach to Optimization and Control Design of Batch Processes,
Journal of Process Control 4(4), pp 291-295, 1994b.

[VER94c]
Verwater, Z., G. Otten,
Creation of a Flexible Recipe for the Epoxy Resin Production (in Dutch),
Internal report NR-1878, Eindhoven University of Technology, 1994c.

[VER95]
Verwater-Lukszo, Z., K.J. Keesman,
Computer-Aided Development of Flexible Batch Production Recipes,
Production Planning & Control 6(4), pp 320-330, 1995.

[VIS94]
Vissers, B.A.J.,
Verbetering van een Batchproces met Behulp van Flexible Recepten (in Dutch),
MSc. Thesis NR-1877, Eindhoven University of Technology, 1994.

[WOO85]
Woods, D.J.,
An Interactive Approach for Solving Multi-Objective Optimization Problems,
PhD Thesis, Rice University, 1985.

Samenvatting

In alle sectoren van de batch georiënteerde procesindustrie worden processen meestal uitgevoerd met behulp van vaste voorschriften: recepten. Deze voorschriften blijven soms jarenlang gelijk, terwijl de proces- en marktomstandigheden behoorlijk kunnen veranderen. Daardoor wordt lang niet altijd efficiënt omgegaan met grondstoffen, energie, produktietijd, etc. Het gevolg is dat er verspilling optreedt, wat in lagere winst, onvoldoende produkt-kwaliteit en/of onnodige milieuvervuiling uitmondt.

De sleutel om de hierboven geschetste problemen op te lossen ligt in de processen zelf. Er moet beter gebruik gemaakt worden van de gegevens die door het proces geproduceerd worden en er moet tevens voor gezorgd worden dat het proces meer nuttige informatie voortbrengt.

Het onderzoek beschreven in dit proefschrift heeft zich gericht op de ontwikkeling van methoden en strategieën voor verbetering van ladingsgewijze produktieprocessen.

De benodigde informatie voor het genereren van optimale recepten, de zogenaamde receptadaptatie-verzameling, omvat de volgende hoofdcomponenten:

1) instelbare of ten minste meetbare receptparameters, b.v. hoeveelheden gedoseerde grondstoffen, procestijd, kwaliteit van de grondstoffen;
2) procesuitgangen, b.v. produktkwaliteit of kwantiteit, hoeveelheid gebruikte energie;
3) relevante plant-, proces- en economische beperkingen;
4) ten minste één doelfunctie;
5) waarde van variabelen die gebruikt worden in de doelfunctie. In de meeste gevallen zijn dat prijzen van de grondstoffen en van de produkten;
6) een nominaal procesmodel en zijn geldigheidsgebied;
7) hulpmodellen voor naburige procescondities;
8) informatie die verzameld is t.b.v. batchcorrectie, met name informatie over de keuze van monster- en correctietijdstippen, mogelijke procescondities geanalyseerd in het genomen monster, mogelijke correctievariabelen (b.v. dosering van extra ingrediënten of verandering van procestijd) en correctiemodellen;
9) specificatie van de te gebruiken proces-units.

Doelfunctie en procesmodellen zijn typerend voor de werkwijze beschreven in dit proefschrift.

Een doelfunctie kan geformuleerd worden b.v. als winst, kosten, of gemiddelde toegevoegde waarde van het proces.

Procesmodellen kunnen verkregen worden door systematische variatie van de relevante receptparameters. De meest geschikte methoden hiervoor zijn factoriële en optimale ontwerpen. In een zogenaamd 2^n-factorieël ontwerp wordt aan elk van n belangrijk veronderstelde factoren, d.w.z. geselecteerde receptparameters, twee verschillende waarden gegeven, zodat 2^n combinaties, de zogenaamde test-runs, ontstaan. Op basis van zulke test-runs kan een tweede-orde model met interactietermen worden bepaald. Door gebruik te maken van meer uitgebreide factoriële methoden, b.v. ster-ontwerpen, is het mogelijk om hogere-orde modellen te schatten.

In de in dit proefschrift beschreven aanpak, genoemd FRIS-aanpak (Flexible Recipe-Improvement System), kunnen twee activiteitsdomeinen onderscheiden worden:

1) Ontwikkeling van een receptadaptatie-verzameling met als doel het ontwikkelen of het verbeteren van een "master-recept" dat voor de gegeven situatie het meest efficiënt is wat bijvoorbeeld gebruik van grondstoffen, beschikbare produktietijd en energie betreft, rekening houdend met het milieu. Tijdens receptontwikkeling en verbetering kan de model-gebaseerde of experimentele werkwijze gevolgd worden.
2) Toepassing van een receptadaptatie-verzameling voor het monitoren of voor het verbeteren van een lopende batch door het genereren van een optimaal "control-recept". Bij het laatst genoemde toepassingsgebied spreekt men over batchinitialisatie, d.w.z. het corrigeren voor afwijkingen van de voorgeschreven begincondities bij het starten van de batch, of over batchcorrectie, d.w.z. het corrigeren voor afwijkingen die tijdens de batch geconstateerd worden.

Ter ondersteuning van de uitgewerkte FRIS-aanpak voor het definiëren en uitvoeren van goed geplande experimenten en vervolgens voor receptontwikkeling, verbetering en aanpassing is het FRIS-pakket ontwikkeld.

De FRIS-aanpak is toegepast op twee gesimuleerde en vijf reële processen met als doel economische procesoptimalisatie en/of kwaliteitsverbetering. In het proefschrift zijn twee case-studies behandeld.

De ene toepassing heeft plaatsgevonden bij Akzo-Nobel Resins in Bergen op Zoom. Het doel was daar driedelig:

1) optimalisering receptuur voor procesduur
2) receptverbetering bij gebruik van nieuwe grondstoffen
3) procescorrectie bij afwijkingen van het gewenste zuurgraad-viscositeit traject.

De andere toepassing betrof het optimaliseren van de toegevoegde waarde van het produktieproces en de eindeigenschappen van epoxyhars Epikote 1001. Hiervoor is een optimaal "master-recept" ontwikkeld, dat vervolgens tijdens batchinitialisatie en correctie aangepast werd aan de waargenomen afwijkingen in het procesgedrag.

De resultaten tonen aan, dat de FRIS-aanpak succesvol toegepast kan worden bij de receptontwikkeling en bij de optimalisatie van winst, procestijd, produktkwaliteit, kwantiteit etc.

Curriculum Vitae

Zofia Verwater - Lukszo was born in Olsztyn, Poland, on February 12, 1959. She attended the comprehensive school Lyceum in Lodz, Poland, from 1973 to 1977. Subsequently, she studied at the Faculty of Applied Mathematics and Technical Physics at Lodz University of Technology, Poland. She obtained her Master's Degree in applied mathematics in 1982.

Parallel with her technical study she studied philosophy at University of Lodz. Upon the completion of this second study in 1983 she worked as a research assistant in the field of semantic logic at the Institute for Logic and Methodology of Science at University of Lodz until 1985.

She has lived in the Netherlands since November 1985.

From April '86 to March '95 she was employed in the System and Control Engineering Group at the Faculty of Technical Physics at the Eindhoven University of Technology, the Netherlands. During the first four years she worked on dynamic optimization of thermal energy systems and parameter identification of distributed systems.

The results of the next five-year STW-project "Computer-Aided Recipe Improvement in the Process Industry" are the subject of this thesis. The intermediate results of the project were presented at various international conferences in the Netherlands, Belgium, Austria, Germany and the United Kingdom.

Since November 1, 1995 Zofia Verwater - Lukszo has worked as an assistant professor in the Environmental Management Section at the School of Systems Engineering, Policy Analysis and Management of the Delft University of Technology.

Zofia Verwater - Lukszo is married and has two sons.

Stellingen

behorende bij het proefschrift:

A Practical Approach to Recipe Improvement and Optimization in the Batch Processing Industry

Zofia Verwater - Lukszo, 1996

1. De industrie heeft niet zozeer behoefte aan de een of andere nieuwe regeltechnische methode, als aan een doeltreffende methodiek met bijbehorende training en ondersteuning door een goed gestructureerde programmatuur.
   (dit proefschrift)

2. Bij het ontwikkelen van recepten dient men, om een zo goed mogelijk resultaat te verkrijgen, grondig "voor te denken" over zowel het kiezen van een kwantificeerbaar doel als van een geschikt experimentontwerp.
   (dit proefschrift)

3. De batch-terminologie die door de ISA tot standaard is verheven, introduceert de onpraktische term "site recipe" en verzuimt een onderscheid te maken tussen verschillende typen van "master" en "control recipes".
   (ISA Standard, *Batch Control. Models and Terminology*, 1995 en dit proefschrift)

4. Uit de literatuur blijkt, dat de meeste onderzoekers op het gebied van multicriteria-optimalisatie zich in de werkelijkheid bezig houden met het zoeken naar een optimum van één kwantitatief criterium. De in dit proefschrift beschreven "triplet-choice"-methode helpt de gebruiker effectief om te gaan met meerdere, menigmaal tegenstrijdige wensen en voert hem op gemakkelijke wijze tot een optimale beslissing.
   (dit proefschrift)

5. Bij het reorganiseren van het wiskundeonderwijs in het VWO is het nodig om naast wiskunde in thematische kaders zoals "de wiskunde in de werkelijke wereld", ook puur-wiskundige vakken, zoals klassieke stereometrie of logica, die het axiomatische karakter van de wiskunde en het wiskundig redeneren duidelijk moeten maken, te onderwijzen.
   (Nieuwland, *Nieuw Archief voor Wiskunde*, maart 1995)

6. Een van de verschillen tussen Nederland en Polen is dat Nederland met slechts circa 2% vrouwelijke hoogleraren op de voorlaatste plaats staat op de lijst van 17 onderzochte landen in diverse werelddelen, en dat Polen met circa 17% op de tweede plaats staat (zie referentie). Hieruit kan geconcludeerd worden, dat het aantal vrouwen in hogere functies bepaald wordt door de cultuur van de samenleving, en niet door de kenmerken van vrouwen.
(Lie en Malik, *The Gender Gap in Higher Education. World Yearbook of Education*, 1994)

7. Gezien de situatie in grote delen van de wereld, wordt in de westerse cultuur het toekennen van waarde aan elk individu ten onrechte niet ervaren als het belangrijkste resultaat van de ontwikkeling van ethische ideeën.

8. In het Europa van de twintigste eeuw heeft in vredestijd nog nooit zo'n plotselinge verslechtering van de gezondheid en zo'n grote toename van sterfte plaatsgevonden als in de meeste Oost-Europese landen na de val van het communisme in 1989 (*Unicef-rapport,* 1995). Dat is hoofdzakelijk toe te schrijven aan maatschappelijke en economische veranderingen en niet aan degradatie van het milieu.

9. De technische universiteiten in Nederland, zich te veel concentrerend op het overbrengen van kennis, onderkennen niet voldoende het belang van het vroegtijdig ontwikkelen van vaardigheden voor presenteren, rapporteren, discussiëren en onderhandelen.

10. In tegenstelling tot Duplo-Primo, dat onlangs tot speelgoed van het jaar werd gekozen in de categorie "baby en peuter", zijn de nu in ontwikkeling zijnde lego-stenen met een micro-chip niet het antwoord op de speelbehoefte van het kind.

11. Mijn stelling: "Ludwig Wittgenstein is de grootste filosoof van deze eeuw" is volgens Wittgenstein zelf zinloos.
(L. Wittgenstein, *Tractatus Logico-Philosophicus)*

12. De doodstraf is in alle opzichten zinloos.

Many batch processes are operated according to *fixed* recipes, in spite of the fact that production would yield more profit, or a better product, if it were efficiently adapted to changes that occur in feedstock qualities and costs, product requirements and prices, process and scheduling conditions, and so on.

The key to improving the situation is to make better use of the data generated by the process and to make the process generate more useful data.
One obviously promising approach is by way of *mathematical modelling*, another is by using the process itself as a model, like in the *Simplex*-approach.

This thesis outlines the *flexible* recipe-approach to efficient development and improvement of *master (control) recipes* by means of systematic experimentation and data handling, and to quick production adaptation by means of *batch initialization* at the start of a batch and by *batch correction* during processing.

ISBN : 90-386-0257-X