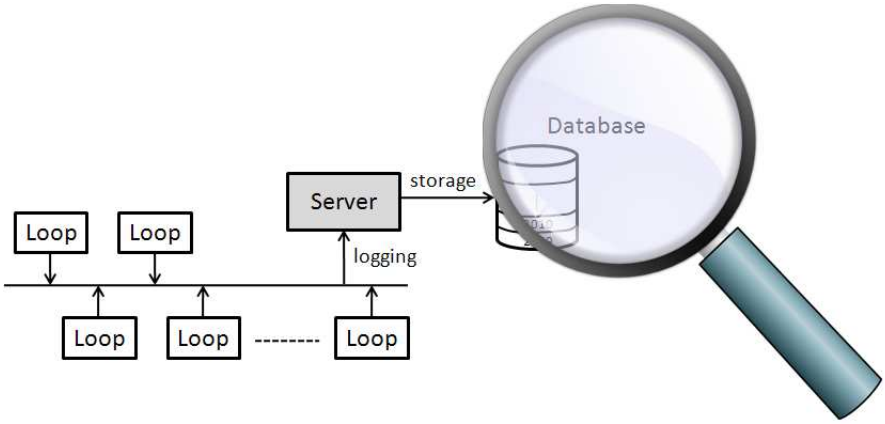
Diplomarbeit 2

Data mining for process identification

Daniel Peretzki



September 2010

UNIKASSEL VERSITÄT

Universität Kassel
FB Maschinenbau
FG Mess- und Regelungstechnik
Prof. Dr.-Ing. A. Kroll

mrt

Diplomarbeit 2

Data mining for process identification

Daniel Peretzki

Matrikelnummer:	24213103
Erstgutachter:	Prof. Dr.-Ing. Andreas Kroll
Zweitgutachter:	Dr. Hanns-Jacob Sommer
Betreuer:	Prof. Alf Isaksson
	André Carvalho Bittencourt
Tag der Abgabe:	27.09.2010

Universität Kassel
FB Maschinenbau
FG Mess- und Regelungstechnik
Prof. Dr.-Ing. A. Kroll

Versicherung

Hiermit versichere ich, daß ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe

---

Ort, Datum, Unterschrift

Sperrvermerk

Diese schriftliche Arbeit enthält vertrauliche Daten.

Sie ist nur den Erst- und Zweitgutachtern sowie befugten Mitgliedern des Prüfungsausschusses zugänglich zu machen. Veröffentlichung oder Vervielfältigung - auch auszugsweise - sind nicht gestattet.

Der Sperrvermerk gilt für eine Zeitdauer von 3 Jahren ab Ausgabe der Aufgabenstellung (bis September 2013).

_____

Ort, Datum, Unterschrift

UNIKASSEL
VERSITÄT

mrt
Mess- und Regelungstechnik
Prof. Dr.-Ing. Andreas Kroll

Diplomarbeit 2

Daniel Peretzki

Thema: Data Mining in der Prozessidentifikation

Dynamische Modelle bilden die Grundlage für modellbasierte Analyse, Simulation, Regelung, Überwachung und Optimierung technischer Systeme. Statt einer in der Regel aufwendigen differentialgleichungsbasierten Modellbildung kann ein Modell aus gemessenen Ein-/Ausgangsdaten (teil-)automatisch erstellt werden.

Häufig ist es in Produktionsanlagen unerwünscht, Experimente durchzuführen, um gezielt geeignete Daten für die Identifikation eines Modells zu gewinnen. Andererseits werden in modernen Prozessleitsystemen und Echtzeitdatenbanken Prozessgrößen erfasst und über lange Zeiträume aufgezeichnet. Diese große Datenbasis kann allerdings nur sehr selektiv für die Identifikation verwendet werden: Prozesse werden i.d.R. für längere Zeiträume stationär betrieben, so dass Ein-/Ausgangsdaten kaum für die Identifikation nutzbar sind.

Das Ziel der Diplomarbeit besteht in der Entwicklung eines Algorithmus unter der Ausnutzung von „Data Mining“ Verfahren, um in sehr großen Datenbasen effizient nach zusammenhängenden Abschnitten der aufgezeichneten Zeitreihen zu suchen, die informativ genug für die Identifikation eines dynamischen linearen Prozessmodells sind. Der Algorithmus sollte mit wenigen A-priori-Informationen über den Prozess auskommen.

Folgende Teilaufgaben sind vorgesehen:

- Einarbeitung in die Problemstellung
- Kurze Recherche zum Stand der Technik
- Entwicklung eines Datenselektionsalgorithmus für informative Abschnitte einer Zeitreihe
- Demonstration der Methoden in einer Fallstudie aus der Prozessindustrie
- Dokumentation der Ergebnisse und Kolloquiumsvortrag

Betreuer: Prof. Alf Isaksson
André Carvalho Bittencourt
Prof. Dr.-Ing. Andreas Kroll
Dr. Hanns-Jacob Sommer

Literaturhinweise:

A. Horch, “Condition monitoring of control loops,” Dissertation, KTH, Signals, Sensors and Systems Dept., 2000.

L. Ljung, *System identification: theory for the user.* 2. Auflage, Upper Saddle River: Prentice-Hall, 1999.

L. Ljung und T. Glad, *Modeling of dynamic systems*. Englewood Cliffs, NJ: PTR Prentice Hall, 2002.

Abstract

In the process industry, models are relevant for various purposes, such as optimizing production, improving control and performance monitoring. Process plants are complex systems and modeling from physical principles is often not feasible. Moreover, performing experiments for system identification is sometimes not possible due to operational reasons and also time-consuming since the time constants involved are often large and there are many interconnections. Nevertheless, the signals of each control loop are often available and can be continuously logged, forming a large database of the plant operation over time. Such database contains a lot of information about the plant but only a few intervals might actually be useful for system identification since the main operation is in steady-state.

This work focuses on the development of a scanning method that automatically searches for intervals of data which are informative enough for system identification. The data studied consists of over three years of continuous measurements from a chemical process plant, where the manufacturing process is mixed (batch and continuous). The proposed method requires a minimum knowledge of the process and is implemented in a simple and efficient recursive algorithm. The essential features of the method are the search for excitation of the input and output, combined with the estimation of a Laguerre model and statistical check of its parameter's significance by a chi-square test. The use of Laguerre models are important to handle delays and filter high frequency disturbances. The method was able to find all intervals in which identification experiments were performed as well as many other useful intervals in closed/open loop operation.

Zusammenfassung

In der Prozessindustrie werden Modelle für verschiedene Zwecke gebraucht, wie zur Produktionsoptimierung, Verbesserung von Regelungen und Überwachung der Regelgüte. Prozessanlagen sind komplexe Systeme und eine Modellierung nach physikalischen Gesetzmäßigkeiten ist oft nicht realisierbar. Weiterhin ist eine experimentelle Systemidentifikation manchmal betriebsbedingt nicht möglich, aufgrund von großen Zeitkonstanten oft zeitaufwändig und eingeschränkt durch viele Kopplungen. Allerdings sind meistens die Signale jedes Regelungskreises messbar und können kontinuierlich aufgezeichnet werden, wodurch ein großer Datenbestand der Anlagenaktivität über eine gewisse Zeit verfügbar ist. Solch ein Datenbestand beinhaltet zwar eine Menge an Informationen über die Prozessanlage, aber nur

wenige Intervalle sind für eine Systemidentifikation wirklich verwendbar, da meistens ein stationärer Zustand vorliegt.

Der Fokus dieser Arbeit liegt auf der Entwicklung einer Scanmethode, die automatisch nach Datenintervallen sucht, welche zur Systemidentifikation informativ genug sind. Die untersuchten Daten bestehen aus kontinuierlichen Messungen von einer chemischen Prozessanlage, welche zum Teil im Chargen- als auch im kontinuierlichen Betrieb war und wurden über einen Zeitraum von drei Jahren aufgenommen. Die vorgestellte Methode erfordert ein minimales Wissen über die Prozesse und wurde mit einem einfachen, effizienten und rekursiven Algorithmus implementiert. Die wesentlichen Bestandteile dieser Methode sind die Suche nach Anregung am Eingang und Ausgang in Kombination mit der Bestimmung eines Laguerre Modells und einem statistischen Signifikanztest seiner Parameter mittels des Chi-Quadrat-Tests. Die Verwendung von Laguerre Modellen ist wichtig um Totzeiten zu beherrschen und hochfrequente Störungen heraus zu filtern. Die Methode ist in der Lage alle Intervalle, in denen Experimente zur Systemidentifikation durchgeführt wurden, sowie andere brauchbare Intervalle im Steuerungs-/Regelungsbetrieb zu finden.

Acknowledgments

The present work was performed during my semester abroad in Sweden. I am grateful for the opportunity to join the automatic control group at Linköping's university. The half year was exciting and I gained a lot of new experiences related to this work as well as outside the university.

My first thanks go to Prof. Alf Isaksson who offered me this project and helped me so much with his ideas and inspiring conversations. Furthermore, my supervisor André Bittencourt is thanked for his scientific support including many comments and suggestions. I would also like to thank Prof. Lennart Ljung for advices based on his great knowledge.

The work reported here is mainly conducted within a cooperation between Linköping University, Perstorp and Asea Brown Bovery (ABB). In this context special thanks go to Krister Forsman and his control group at Perstorp who provided me with data and new ideas during the project.

Besides that, thanks go to Prof. A. Kroll at the University of Kassel who arranged the contact to Sweden and examines this work together with Dr. Hans Sommer.

Last but not least, my parents deserve a special acknowledgement for supporting me during my studies in Kassel and Linköping as well as Claudia who gave me, despite the distance, the strength to accomplish this work.

Daniel Peretzki
Hamburg
September 2010

Contents

Nomenclature

α	Real pole Laguerre model
α_c	Significance level (chi-square test)
$\varepsilon(k,\theta)$	Prediction error $y(k) - \hat{y}(k\|\theta)$
λ_R	Forgetting factor for recursively updating $\boldsymbol{R}$
λ_m	Filter parameter for calculating $m_{u,f}(k)$
λ_v	Filter parameter for calculating $v_{u,f}(k)$
$\boldsymbol{\varphi}$	Regressor vector
$\boldsymbol{\Phi}$	Regression matrix
$\boldsymbol{\Phi}_{k_{est}}$	Part of the regression matrix; $\boldsymbol{\Phi}_{k_{est}} = \boldsymbol{\Phi}(k_{est} \dots k, 1 \dots n)$
$\hat{\sigma}_e^2$	Estimated noise
σ_r	Singular values of matrix $\boldsymbol{S}$
$\hat{\sigma}_u^2$	Estimated variance
$\boldsymbol{\theta}$	Parameter vector
$\boldsymbol{\theta}_0$	Expected parameter vector
$\hat{\boldsymbol{\theta}}$	Estimated parameter vector
ν	Degrees of freedom (Chi-square test)
$\chi^2(\theta, \hat{P})$	Computed chi-squared
$\chi^2_{\alpha,\nu}$	Quantile of the standard normal distribution with ν degrees of freedom
$A(q^{-1})$	Polynomial in discrete time-domain
a_i	Polynomial coefficient
$B(q^{-1})$	Polynomial in discrete time-domain
b_i	Polynomial coefficient
$J(\theta, N)$	LS cost function
$C(q^{-1})$	Polynomial in discrete time-domain
c_i	Polynomial coefficient
$cond(\boldsymbol{R})$	Condition number of matrix $\boldsymbol{R}$
$\boldsymbol{E}$	Noise vector
$\bar{E}$	Expected value
e	Gaussian white noise disturbance ($N(0, \sigma^2)$)
F	Transfer function from $(r - y)$ to u
G	Transfer function from u to y
k	Sample point
k_{est}	Sample, where $v^*_{L_1,f}(k)$ exceeds the first time th_{est}
k_{start}	Sample, where $v^*_{u,f}(k)$ exceeds the first time th_{start}
k_0	Sample, where the first time $u(k) \neq 0$
$L_i(q, \alpha)$	Laguerre filter i
$m_{u,f}(k)$	Estimated mean of $u(k)$
n	Model order
n_a	Number of coefficients in the A plynomial
n_b	Number of coefficients in the B plynomial
n_c	Number of coefficients in the C plynomial

N	Number of measurement points (samples)
n_p	Total number of parameters
$\hat{\boldsymbol{P}}$	Estimated covariance matrix of $\hat{\boldsymbol{\theta}}$
$\boldsymbol{Q}$	Orthogonal matrix of QR factorization
q, q^{-1}	Forward and backward shift operator, variable of z-transformation
$\boldsymbol{R}$	Information matrix
$\bar{\boldsymbol{R}}$	Upper triangular matrix of QR factorization
$\boldsymbol{R}_0$	Part of matrix $\bar{\boldsymbol{R}}$ with the dimension $(n+1) \times (n+1)$
$\boldsymbol{R}_1$	Part of matrix $\boldsymbol{R}_0$ with the dimension $n \times n$
$\boldsymbol{R}_2$	Part of matrix $\boldsymbol{R}_0$ with the dimension $n \times 1$
R_3	Part of matrix $\boldsymbol{R}_0$ with the dimension 1×1
$\hat{R}_\varepsilon^N$	Scalar that indicates the model's quality (see Chapter 2.6.1)
$\boldsymbol{S}$	Diagonal matrix of SVD
s_i	Continuous-time zero (see Chapter B)
r	Setpoint
T_d	Time delay of d samples
T_s	Sample time in seconds
T_0	Number of past samples with a weight bigger than $\approx 36\%$ (see Chapter 2.3.4)
t	Continuous time in seconds
t_{step}	Step response settling time
th_c	Threshold for the maximum allowed $cond(R(k))$
th_{conf}	Threshold of confidence interval for chi-square test
th_{est}	Threshold for finding the start of the marked stretch
th_{gap}	Threshold for maximum gap between two steps
th_{step}	Threshold which determines the minimum size of a step in $u(k)$
th_{ss}	Threshold which defines the minimum duration of u(k) and y(k) together in SS before and after the step
th_{start}	Threshold for finding the start of the marked stretch
th_{v,L_1}	Threshold for minimum required variance of $L_1(k)$
$th_{v,u}$	Threshold for minimum required variance of $u(k)$
$th_{v,y}$	Threshold for minimum required variance of $y(k)$
$\boldsymbol{U}$	Orthogonal matrix of SVD
u	Process input signal
$\bar{u}_k$	Sample mean of $u(k)$
$\boldsymbol{V}$	Orthogonal matrix of SVD
$v_{L_1,f}(k)$	Estimated variance of $L_1(k)$
$v_{u,f}(k)$	Estimated variance of $u(k)$
$v_{y,f}(k)$	Estimated variance of $y(k)$
$v_{u,f}^*(k)$	Estimated variance of $u(k)$ computed by another way compared to $v_{u,f}(k)$
$v_{u,f,0}(k)$	Variance of $u(k)$ in steady-state
$\boldsymbol{Y}$	Process output vector
$\hat{\boldsymbol{Y}}$	Model predictions vector
$\hat{\boldsymbol{Y}}_{k_{est}}$	$= \hat{\boldsymbol{Y}}(k_{est} \dots k)$

y	Process output signal
$\hat{y}(k\|\theta)$	Model output for parameter vector $\boldsymbol{\theta}$ and measurement k
y_d	Disturbed process output signal
Z	Data set
z_i	Discrete-time zero
$\|\cdot\|_2$	L_2-norm (Euclidean norm)

Abbreviation

ADC	Analog-digital converter
AR	Auto-regressive
ARMA	Auto-regressive moving average
ARMAX	Auto-regressive moving average with exogenous input
ARX	Auto-regressive with exogenous input
B	Bias error
CPM	Control loop performance monitoring
FIR	Finite impulse response
LS	Least-squares
LP	Low-pass
MA	Moving average
MIMO	Multiple-input multiple-output
ML	Maximum likelihood
MPC	Model Predictive Control
MSE	Mean square error
PID	Proportional Integral Derivative (PID controller)
SISO	Single-input single-output
SNR	Signal to noise ratio
SS	Steady-state
SVD	Singular value decomposition
V	Variance error

Chapter 1

Introduction

1.1 Motivation

In the process industry, models are relevant for various purposes, such as optimizing production, improving control and performance monitoring. For control loop design it is necessary to have enough information about the process, e.g. in the form of a mathematical model. Especially the development and application of modern control methods require more or less accurate process models. Furthermore, in a typical industrial process plant, several hundreds or even thousands of basic control loops like PID or PI are installed. Those controllers have to be tuned and if the process is sensitive and dangerous, a process model has to be as accurate as possible. In addition, changes might occur in a process, caused for instance by interactions with the surrounding environment, therefore the model should be updated accordingly to keep consistency. Moreover, in practice a lot of models for control are still specified by the manufacturer of the controllers or just improperly tuned and thus inaccurate.

These factors show that a better performance of the control system is often possible if an updated model of the process is used. Hence, modeling control loops is one of the main tasks in control optimization. The requirements concerning the model complexity depend on the use of the model. For example for monitoring applications, which combine failure detection and predictive maintenance, simple models could be sufficient. In general, the model must represent the main dynamics in the relevant frequency band [13]. The following points are nowadays the main purposes for process models:

- Control design and tuning
- Process monitoring
- Control loop performance monitoring (CPM)

Every day a lot of identification methods are in practical use in companies for different applications [13]. One of the first steps a user has to perform in system

identification is to design a well planned experiment to collect data that contain information about the system dynamics/properties of interest. An accurate model might be possible in case the process can be excited with purpose of identification. In the process industry however, this can become a time-consuming task since the involved time constants are often large and there are many interconnections within the process plant. Furthermore, those experiments are often not possible for operational reasons. Alternatively, models can be derived from physical principles but process plants are mostly complex systems and therefore it is a difficult task.

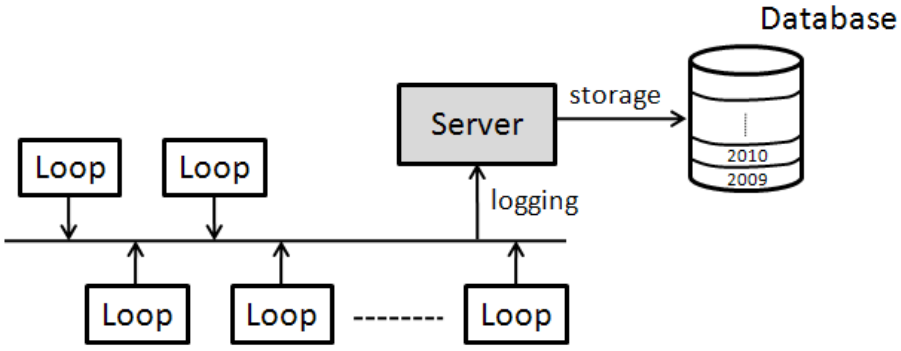


Figure 1.1. Procedure of logging process signals and forming a database.

Nevertheless, the signals of each control loop are often available and can be continuously logged, forming a large database of the plant operation over time (see Figure 1.1). Storage of data is nowadays cheap and e.g. in production units the measuring tools and data collectors have been developing in an impressive way. Huge amounts of industrial plant's entire process data, logged over multiple years with a size of GigaBytes or TeraByte, are the common case today and in near future PetaByte will be the next usual quantity to be measured [13]. This trend is boosted by several reasons, on one hand by the parallel development of increasingly powerful data processing systems, on the other hand, by more and more systems which are too complex to understand in detail, therefore this deficit of knowledge is tried to be compensated by the nowadays cheap storage of huge amounts of data which might contain a lot of information about the system [1]. The availability of a large database about the plant operation can be seen as a great opportunity to obtain useful information about the system's properties and behaviour. Such a database contains plant measurements over several months or even years and gives an alternative way to build models using such operation data.

The main problem with this data is its remarkable size and considering that only some intervals of data might be useful for system identification. The operation mode is mostly *automatic* and setpoint changes occur only occasionally whereby the main operation is in steady-state. The key to success is the use of *data mining* techniques that are also described as 'information discovery' or 'data pattern recognition' and which simply means "sorting through large databases and finding

relevant information" [13]. The idea is to combine data mining with process identification and to develop an automatic scanning procedure which searches through huge amounts of stored process data for informative intervals which are useful for process identification.

This motivates the following work which arose by a cooperation with the Swedish chemical company Perstorp. Perstorp is a world leader in several sectors of chemicals and produces mainly intermediate goods like polyols, isocyanates, acids and ester as well as some end products like additives for paints and lacquers or polyurethane. The control group at Perstorp is strongly interested in such a scanning method due to the aforementioned reasons. For this work they provided three years of data of 211 control loops, which originate from one of their chemical process plants that operates mixed (continuously and batch).

The report is organized as follows: In Section 1 a brief introduction of the previous attempts of Perstorp concerning the above mentioned problem is given as well as an overview of related work world-wide. This concludes in a clear target definition of the project which is reported here. Section 2 presents the necessary theory for the developed scanning methods. All important information about the used data is provided in Section 3. In Section 4 the main features for system identification are worked out in terms of requirements, determination of excitation in a signal and handling of disturbances. Some real examples of estimated process models are presented. The main contribution is given in Section 5 where the differently developed methods are illustrated and compared with each other. The validation of the finally chosen method is presented in section 6.

1.2 Previous work at the plant

The control group at Perstorp had already tried to find a proper method for detecting useful data intervals, but only with moderate success. Their approach was to scan the data for changes in the controller output or setpoint signal. Then, different types of process models like P1D, I, IP, IP1D[1] were automatically estimated and checked afterwards for the uncertainties/statistical significance of their parameters. The best model was then chosen. However, this method caused problems when estimating automatically the models with the *System Identification Toolbox* of MATLAB™ because the toolbox crashed in cases where the choice of the model structure was unsuitable.

During manual system identification, they executed so called 'bump tests', where the system is set in manual mode and then excited with an input step. The observed response of the system (step response) delivered all information to build a model of the system. This is still done and can be considered as a common way in the process industry.

[1] P = Proportional, 1 = first order, D = derivate, I = integral

Another applied way for manual estimation of models was looking for data intervals where the control loop was in automatic mode and the controller output went into saturation. This often delivered good models.

1.3 Related work

Current research in the area of data mining for system identification is not really active which is reflected in the low number of world-wide publications to this topic.

A lot of past work address persistent excitation for on-line system identification (e.g. [6]) but do not deal with finding intervals of useful data. Cybenko presented a method called "Just-in-Time-models" [4] that retrieves a subset of data closest to the desired operating point and performs a modeling operation on that subset of data. In [2], a data removal criterion is presented that uses the singular value decomposition (SVD) technique for discarding data which is only noise dependent and leads to a bigger mean square error (MSE) of the model estimation parameters. Furthermore, Horch introduced in [7] a method for finding the transient parts of data after a setpoint change occurred.

Some ideas can be extracted of these works but none of them delivers an approach for scanning huge amounts of data for stretches that are informative enough for system identification.

1.4 Aim of the project

Important for the goal definition is what the final user requires or better said:

"How will the client use the developed method?"

For the project we are only considering the identification necessary for PID tuning. In this context the final user wants to identify simple linear low-order models with single input single output (SISO) that can be obtained from either reference changes in automatic mode or controller output changes in manual mode. If the prospective user gets a useful data interval for system identification he can decide by himself which type of model to estimate.

Therefore, this work focuses on the development of a scanning method that automatically searches for intervals of data which are informative enough for process identification. If the algorithm finds an interval, it saves the stretch together with an indicator of its quality. Then the final user can extract the data stretch and execute process identification for achieving a model.

The following list summarizes the requirements for the scanning tool/method:

- Only available measurement signals can be used,
- Keep required knowledge about the process to a minimum,

- Fast and therefore simple algorithm for scanning the data of hundreds or more control loops,
- The processes which should be identified are restricted to be simple linear low-order models and SISO,
- Should also work if APC-Tools (Advanced process control) are used,
- The output of the method delivers start and end time and as well a quality attribute of the intervals which could be useful for process identification.

To understand, what is hidden behind the data, is essential for the development of the method and important for a good performance of the final product. Therefore the chapters 2-4 deal with the basics of system identification, plant conditions at Perstorp, data description and data features.

Chapter 2

System Identification

2.1 Why System Identification?

In many cases the necessary information to build a model from first principles of a dynamical process is lacking or the system is simply too complex. Then, empirical modeling and identification is an alternative where observed input and output data from a process is used to build a model that describes the data in a functional relation. This is an essential task in many scientific divisions. Related to control applications it is known as *System Identification*.

System Identification has been proved in many applications to be a proper method for model building [14]. The system properties are in particular modeled by estimating the values of the system through comparisons of predicted behavior with measurements. The quality of the final model depends on many factors during the system identification procedure (see Chapter 2.2). For example, the model reliability is affected by a couple of assumptions about its structure as pointed out in the following section.

2.2 Procedure overview

The aim of the identification procedure is to estimate a model with the measurements of the process input/output signals, which should be informative concerning the static and dynamical system behaviour. Therefore, we assume a correlation between the input and output signal of the process.

The fundament of system identification is always a model-like description of the physical reality by mathematical models (difference or differential equations) that describe the relationship of input and output of the system.

A model can basically be divided into model structure and model parameters. The model structure distinguishes the general behaviour and complexity of the

model (qualitative model description) while the model parameters determine especially the behaviour of a given model structure (quantitative model description). This has to be kept in mind during the whole modeling process. For example, a linear model structure of course can only describe linear process properties.

The following basic approaches depend on the previous knowledge about the modeling system:

- **White-box models**: White-box models result from a precise theoretical analysis of the system behaviour. This analysis is based on the set up of physical and geometrical equations.
- **Black-box models**: If one wants to avoid an extensive theoretical system analysis or has only insufficient knowledge about the system behaviour, the experimental analysis or identification could be an alternative. The result of an identification is a black-box model or grey-box model which are distinguished by the level of previous knowledge. In the case of hardly no prior knowledge a black-box or so called ready-made model can be estimated whose parameters have no physical meaning. This type of model reproduces only the input/output behavior.
- **Grey-box models**: They are a mixture of white- and black-box models. Grey-box models contain information from physical equations and measurements and are often based on assumptions to narrow the possible model candidates and structures. They have the advantage that often less parameters have to be estimated compared to black-box models and that the identified parameters have physical meaning.

For this project the only possible choice would be black- or grey-box models, because a detailed physical analysis of each control loop will be impossible in a automatical scanning method and has many disadvantages as already mentioned. However, black- or grey-box models entail the problem to estimate a proper model of the system by using only the relevant data of the measurement and avoiding the irrelevant data caused by noise and disturbances. In any case, the entire modeling process should receive as much information about the system as available for a successful result.

Nowadays, almost all time continuous processes are controlled by digital computers or control units. The main difference to analog control is the operation in discrete time. The measurement of the process variable $y(t)$ and the setting of the control value $u(t)$ is made by analog-digital converters (ADC) at discrete time points. The duration between two processing steps is called the *sample time* T_s. Hence, the input/output data used for system identification is only available in discrete time instants. Quantization errors can be neglected because of the common ADCs with a bit width of 12-16.

The measurements available for system identification consists then of the process

input sequence $\{u(k)\} = [u(1), u(2), \ldots, u(k)]$ and of the process output sequence $\{y(k)\} = [y(1), y(2), \ldots, y(k)]$ with $k = N \cdot T_s$ and $N :=$ number of samples. That means that only a finite number of past values is used and a discrete-time model can only deliver the relation between input and output signal at the sample time points. Furthermore, the process input signal $u(k)$ is crucial, because the quality of the estimated model depends strongly on it. An exciting enough input signal is necessary to force the process to a response that contains proper information about its dynamical behaviour so that an adequate model can be identified.

After the measurement data is given, the next step to think about is which set of candidate models (choice of the model structure) should be used for system identification. Unfortunately this question cannot be exactly answered. All information about the insights, a priori knowledge and engineering intuition leads to a model set that has adequate properties.

In the following step the parameters of the given model structure are adjusted with the aim to reduce the error between process and model as much as possible. Dependant on the model structure the parameters of the disturbance dynamics are as well estimated. This step is called *parameter estimation*. In general, the estimation is done by solving an over-determined system of input/output signals and using an error criterion (loss function).

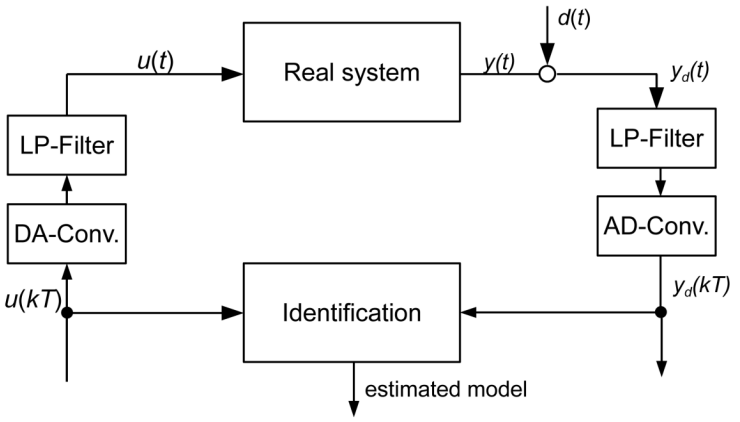


Figure 2.1. Structure of identification for a dynamic process.

Since there is on each process an impact of disturbances like measurement noise, it is only possible to use a disturbed signal for system identification, which consists of the sum of undisturbed and disturbed signal (see $y_d(t)$ in Figure 2.1). The aim is therefore to identify the system dynamics as accurately as possible from the data despite noise in the signal.

The last step is to validate the model and thereby to prove if the model can be finally accepted. Often the model is rejected because the last step fails, which can be caused by different reasons like:

1. The data for the estimation procedure was not informative enough
2. The model structure cannot explain the "true" system
3. The numerical procedure for parameter estimation failed

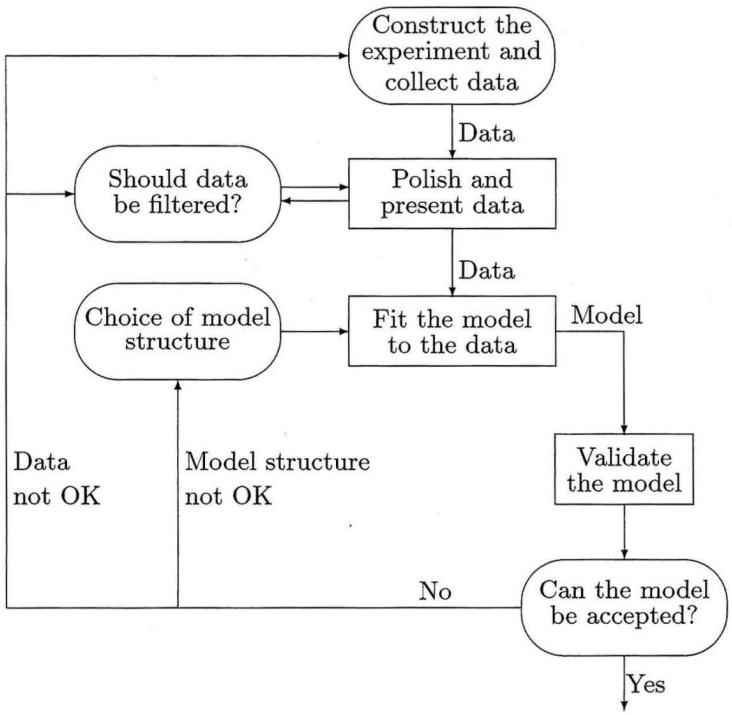


Figure 2.2. System identification procedure.

Figure 2.2 taken from [14] shows the whole procedure graphically. To summarize it, the main steps of the system identification process are:

1. In case that a experiment design is possible, collect data set that corresponds as close as possible to the data of the final model application

2. If necessary remove trends and outliers. Apply filtering to remove measurement and process noise

3. Selection of the model structure

4. Estimate parameters

5. Validation of the model, using an objective function. If the model is not satisfactory then repeat step 3

6. If a satisfactory model is still not obtained in step 5 then repeat the procedure either from step 1

According to [14] the main tasks of the user during the whole identification procedure is usually to consider the available data and think about a suitable model structure, a suitable parametrization of the model and to evaluate the estimated models .

2.3 Linear model estimation

2.3.1 Different model structures

The choice of the model structure is one of the most difficult questions. Some aspects which influence the choice are the type of the model: nonlinear or linear, static or dynamic, single-input single-output system (SISO) or multiple-input multiple-output system (MIMO).

Because of the project task, which does not offer the possibility for regarding the physical insights, only ready-made models are taken into account (see Chapter 2.2). These kind of standard models are known to be well suited for linear system and to handle a wide range of different system dynamics [14]. Furthermore we know already that the measured data is only available in discrete form and of SISO processes therefore we focus on time discrete linear SISO models. For these reasons the main characteristic of the interesting model structure is that it depends linearly on the past input and output data. Since in a dynamic system the future depends every time on the past measurements the chosen model structure will use the data set $Z = \{y(k-1), u(k-1), y(k-2), u(k-2), \dots\}$.

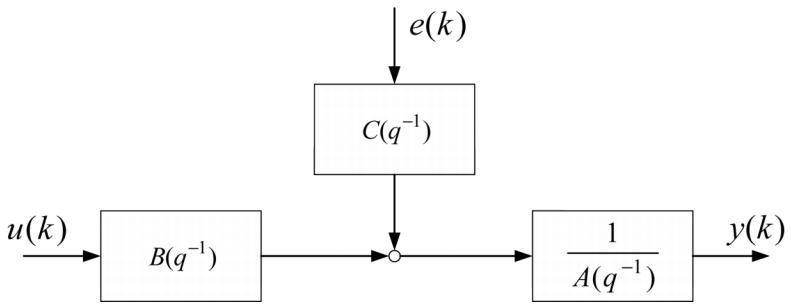


Figure 2.3. Block diagram of the ARMAX model structure.

Figure 2.3 shows the ARMAX structure with the corresponding equation [14]:

$$A(q^{-1})y(k) = B(q^{-1})u(k) + C(q^{-1})e(k) \tag{2.1}$$

where

u(k)	=	input signal
y(k)	=	output signal
e(k)	=	Gaussian white noise disturbance ($N(0, \sigma^2)$)

with the polynomials

$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_{na} q^{-n_a}$$
$$B(q^{-1}) = b_0 + b_1 q^{-1} + \ldots + b_{nb} q^{-n_b}$$
$$C(q^{-1}) = 1 + c_1 q^{-1} + \ldots + c_{nc} q^{-n_c}$$

q^{-d} is the shift operator which represents a time delay (dead time) of $T_d = d \cdot T_s$ or d samples. This operator shifts the input signals of the general model structure in the following way:

$$A(q^{-1})y(k) = q^{-d} \cdot B(q^{-1})\ u(k) + C(q^{-1})\ e(k) \tag{2.2}$$
$$= B(q^{-1})\ u(k-d) + C(q^{-1})\ e(k) \tag{2.3}$$

Moreover, we assume that the sampling interval T_s is one time unit and $e(k)$ is white noise, that means all values $e(k)$ are statistically independent with mean zero and constant variance.

The ARMAX model structure (2.1) is the basis for different types of model definitions, which result from different assumptions about the input and disturbance signal as well as about the polynomial A, B, C.

The following special models can be distinguished with the general model structure:

- **Auto-regressive (AR) model**: An AR model is achieved if $n_b = n_c = 0$ and $u(k) = 0\ \forall\ k$. In this case:

$$A(q^{-1})\ y(k) = e(k) \tag{2.4}$$

- **Moving Average (MA) model**: For the MA model is $n_a = n_b = 0$ and $u(k) = 0\ \forall\ k$ assumed. Then the following equation is valid:

$$y(k) = C(q^{-1})\ e(k) \tag{2.5}$$

- **Auto-regressive moving average (ARMA) model**: It is spoken of an ARMA model, if $n_b = 0$ and $u(k) = 0\ \forall\ k$:

$$A(q^{-1})\ y(k) = C(q^{-1})\ e(k) \tag{2.6}$$

- **Finite impulse response (FIR) model**: A FIR model is achieved in case of $n_a = n_c = 0$:

$$y(t) = B(q^{-1})\ u(k) + e(k) \tag{2.7}$$

- **Auto-regressive model with exogenous input (ARX)**: For the ARX model is assumed that $n_c = 0$:

$$A(q^{-1})\ y(k) = B(q^{-1})\ u(k) + e(k) \tag{2.8}$$

- **Auto-regressive moving average model with exogenous input (ARMAX)**: For the ARMAX model exists no special assumptions. It is given through equation (2.1).

In the previous models the dynamic of the system and of the disturbance have the same denominator $A(q^{-1})$. However, to get an more accurate model, we would have to use *output error* (OE) or *Box-Jenkins* models respectively. For OE models the system dynamics are modeled separately in contrast to the disturbance dynamics and for BJ models even the disturbances properties are additionally modeled.

Choice of the model structure

Each of the AR, MA, ARMA, ARX, and ARMAX structures offer their own advantages. To decide which model type is the best for a certain system identification problem one possibility can be to define the orders of the parameters and pick the best model afterwards [14]. However, this method demands always a critical scrutinizing of the parameter order. Therefore, a relevant question is still how to use the freedom in modeling system and disturbance dynamics that the different model structures give.

In case of identification without manipulated input signal MA-, AR- and ARMA-models should be used [15]. Autoregressive (AR) and autoregressive moving-average (ARMA) models are among the most used parametric models in time series analysis, because of their simplicity and good approximations for many processes of interest. Only numerical sensitivity (computational accuracy) limits the maximal order of an AR model. The estimation of the parameters of an AR model (see definition (2.4)) is easy [7] and only involves a quadratic least squares optimization problem (regression vector consists only of process output values; see Section 2.3.2). The ARMA model requires that a nonlinear estimation problem is solved.

FIR or MA models take no past output signals into account (not recursive) and represent a linear convolution. They use only the process input signals values $u(k)$ or $e(k)$ respectively for calculation of the output signal $y(k)$. This has the advantage that they stay always stable as long as the input signals are limited ($n_b < \infty$). The big disadvantage is that they have only finite memory. Hence, if a really slow process should be modeled, a lot of parameters are required [7].

If additionally deterministic inputs $u(k)$ should be considered then ARX- or ARMAX-models should be chosen (FIR is a special case). In case of no precise information about the disturbance signal and the estimation problem is of a linear regression type, then the ARX-model is a proper choice, even if the non measured disturbances $e(k)$ will however affect the accuracy of estimated model. The ARX-model presents a linear transfer function in the z plane and is often chosen because of its simplicity and easiness to estimate the model.

Compared to the ARX model the ARMAX model gives due to the C polynomial more freedom in dealing with the disturbance dynamics. The biggest disadvantage of the ARX and ARMAX models are that the A polynomial has an influence on the process and as well of disturbances (system and the noise dynamics have the same denominator). But this is of less consequences if the disturbances enter the process early [14] or in case of a good *signal to noise ratio* (SNR).

Choice of the model order

How good the model can approximate the "true" system depends on the structure and as well on the model order. A lot of proposals are made in the corresponding literature (see. e.g. [14]) how to choose the model order. Most of them try several orders and look for the best model performance. Clarke gives in [3] a rule of thumb for determination the order of an FIR model: The model order n must be such that $n \cdot T_s$ exceeds the plant's settling time.

Remark: The time discrete model can always be transformed into a time continuous model [14].

2.3.2 Linear Regression

The ARX equation (2.8) can be reformulated to a difference equation:

$$\begin{aligned} y(k) = &- a_1\ y(k-1) - \ldots - a_{na}\ y(k-na) \\ &+ b_0\ u(k) + b_1\ u(k-1) + \ldots + b_{nb}\ u(k-nb) \\ &+ e(k) \end{aligned} \tag{2.9}$$

If equation (2.9) is written in vector form we can derive the following formulation:

$$y(k) = \boldsymbol{\varphi}^T(k)\ \boldsymbol{\theta} + e(k) \tag{2.10}$$

where $\boldsymbol{\varphi}^T(k)$ denotes the *regressor vector*

$$\boldsymbol{\varphi}^T(k) = [-y(k-1)\ \ldots\ -y(k-na)\ u(k)\ \ldots\ u(k-nb)] \tag{2.11}$$

and $\boldsymbol{\theta}$ the *parameter vector*

$$\boldsymbol{\theta}^T = [a_1\ \ldots\ a_{na}\ b_0\ \ldots\ b_{nb}] \tag{2.12}$$

Based on equation (2.10) we can now build an equation system for N measurement points

$$\boldsymbol{Y} = \boldsymbol{\Phi}\ \boldsymbol{\theta} + \boldsymbol{E} \tag{2.13}$$

with the *regression matrix*

$$\boldsymbol{\Phi} = \begin{pmatrix} \varphi^T(1) \\ \varphi^T(2) \\ \vdots \\ \varphi^T(N) \end{pmatrix} \tag{2.14}$$

the process output vector

$$\boldsymbol{Y}^T = [y(1)\ y(2)\ \ldots\ y(N)] \tag{2.15}$$

and the noise vector

$$\boldsymbol{E}^T = [e(1)\ e(2)\ \ldots\ e(N)] \tag{2.16}$$

If a time delay of d samples should be modeled this is done with an additional time shift operator q^{-d}.

2.3.3 Parameter estimation with least squares method

The previous chapter derives an equation system (2.13) where the values of the parameter vector $\boldsymbol{\theta}$ (2.12) are unknown. After having chosen the total number of parameters n_p $(= na + nb)$ of the linear time discrete model the next step is now the parameter estimation. A so called error criterion (cost function) is used in terms of model parameter estimations which has to be reduced to a minimum.

Before starting with the parameter estimation, it is important that there are more observations than coefficients $(N > n_p)$. This means that we use more equations than parameters for averaging the measurement errors. In this case the equation system is overdetermined and there does not exist a common exact solution.

Then the maximum likelihood (ML) method is for many estimation problems the approach to the estimation procedure [24]. For the linear ARX model (2.8) and the related equation system (2.13) the ML cost function can be approximated by a quadratic one. This is the so called *least squares cost function* (Euclidean norm) and the whole method is known as the *least squares (LS) prediction error method*. This method leads to the most probable parameters of the model for random distributed deviations between model and process outputs. The remaining deviations are called *residuals* which are used for predictions concerning accuracy and reliability of the model (see Section 2.5.1).

The model output for a certain parameter vector $\boldsymbol{\theta}$ is introduced by $\hat{y}(k|\boldsymbol{\theta})$. The residual (prediction error) can then be calculated by the equation:

$$\varepsilon(k, \boldsymbol{\theta}) = y(k) - \hat{y}(k|\boldsymbol{\theta}) = y(k) - \varphi^T(k)\ \boldsymbol{\theta} \tag{2.17}$$

The LS cost function for solving the estimation problem is

$$J(\boldsymbol{\theta}, N) = \frac{\|\varepsilon(\boldsymbol{\theta})\|_2^2}{N} = \frac{1}{N}\sum_{k=1}^{N}\left(y(k) - \varphi^T(k)\ \boldsymbol{\theta}\right)^2 \tag{2.18}$$

where $\|\cdot\|_2$ denotes the L_2-norm (Euclidean norm). This norm has among others the benefit to execute no case distinction for the algebraic sign and to allow that big differences between measured and estimated signal have a higher influence on the final result.

The values of the parameters are now determined by minimizing the cost function (2.18) in the LS prediction error sense. The derivation of the final equations, whose result is the best system approximation, can be found in lots of literature (e.g. [12]) and is not derived here in detail.

We introduce the parameter vector $\hat{\boldsymbol{\theta}}$ which is estimated from the measured and noisy signal. The vector $\boldsymbol{\theta}_0$ is the optimal parameter vector that describes the process exactly when no disturbances would exist. If the regressor matrix $\boldsymbol{\Phi}$ (2.14) has full rank the optimal solution vector $\hat{\boldsymbol{\theta}}$ is unique. The solution of the LS estimate $\hat{\boldsymbol{\theta}}$ can then be found for the values of $\boldsymbol{\theta}$ that minimize the cost function (2.18):

$$\begin{aligned} \hat{\boldsymbol{\theta}} &:= arg\min_{\boldsymbol{\theta}} \{J(\theta, N)\} \\ &= arg\min_{\boldsymbol{\theta}} \left\{ \frac{1}{N} \sum_{k=1}^{N} \left[y(k) - \hat{y}(k|\boldsymbol{\theta}) \right]^2 \right\} \end{aligned} \tag{2.19}$$

The values of $\hat{\boldsymbol{\theta}}$ can be calculated with the equations (2.20) and (2.21) which are derived in [14]. The optimization problem is solved by the normal equations:

$$\hat{\boldsymbol{\theta}} = \boldsymbol{R}^{-1} \frac{1}{N} \sum_{k=1}^{N} \boldsymbol{\varphi}(k) y(k) = \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^T \, \boldsymbol{Y} \tag{2.20}$$

$$\text{where} \qquad \boldsymbol{R} = \frac{1}{N} \sum_{k=1}^{N} \boldsymbol{\varphi}(k) \; \boldsymbol{\varphi}(k)^T \tag{2.21}$$

Usually a lot of samples (total number = N) are taken but only a few coefficients should be estimated. The matrix $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$ has the dimension $n \times n$ and is often quite small compared to the design matrix $\boldsymbol{\Phi}$ with the dimension $N \times n$. Nevertheless, calculating the inverse of $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$ and solving the system of equations (2.13) requires a lot of computing time. Furthermore the process has to be excited enough to guarantee full rank of $\boldsymbol{R}$ so that the inverse $\boldsymbol{R}^{-1}$ exists and the normal equation system can be solved.

Especially if the matrix $\boldsymbol{R}$ has a high dimension it can be ill conditioned. In terms of the condition number (see Chapter 4.2), which defines the solution accuracy of a linear equation system, it is remarkable that by using $\boldsymbol{R}$ the condition number is squared [12]. Hence, the final equation system (2.20) is worse conditioned than the original overdetermined system (2.13).

Therefore it should be avoided to solve the estimation problem by using the matrix $\boldsymbol{R}$, instead it can be solved, for example, with QR factorization (see Section 2.3.5).

2.3.4 Recursive identification

Sometimes it is necessary to update the model parameters in certain time intervals. For example, an online model estimation process, whose last estimated model is updated with each new measurement/sample. In this case it would be advantageously to have a method that provides an update of the information matrix $\boldsymbol{R}$ (see equation (2.21)) with every new measurement. In addition, it is often interesting that the current data (last few measurements) have a significant higher influence on the values of $\boldsymbol{R}$ than very "old" observations.

Ljung introduces in [12] *recursive estimation methods* which are based on the *weighted least-squares criterion* (compared to equation (2.19) a weighting is added). The following equation with the so called *forgetting factor* λ delivers such a recursive method:

$$\boldsymbol{R}(k) = \lambda \cdot \boldsymbol{R}(k-1) + (1-\lambda) \cdot \boldsymbol{\varphi}^T(k)\ \boldsymbol{\varphi}(k) \tag{2.22}$$

The forgetting factor λ determines the weighting of the old data related to the current calculation of $\boldsymbol{R}$. For example a constant choice of $\lambda = 0.9$ yields a weighting of the past samples which is shown in Figure 2.4. For a lower value of λ the current measurements are more weighted and have a higher influence on the calculated values of $\boldsymbol{R}$. Ljung gives in [12] the following equation

$$T_0 = \frac{1}{1-\lambda} \tag{2.23}$$

which calculates the number of past measurements with a weight bigger than $\approx 36\%$. A typical value for λ lies in the range between 0.98 and 0.995.

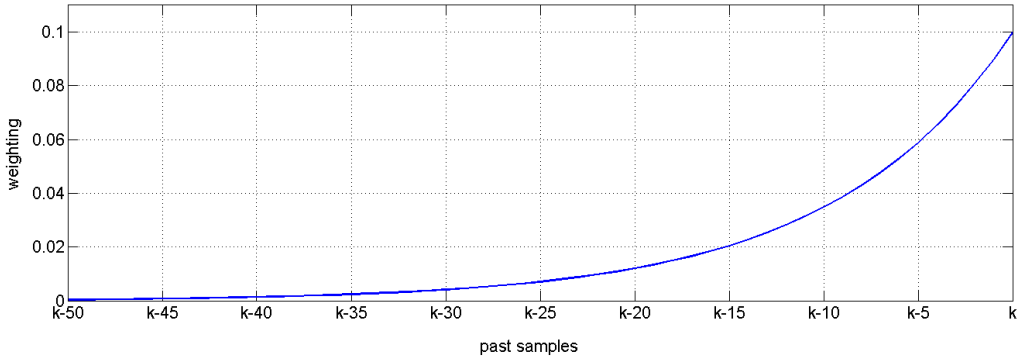


Figure 2.4. Weighting of the past samples for a forgetting factor $\lambda = 0.9$.

2.3.5 QR Factorization

An effective way to estimate $\hat{\boldsymbol{\theta}}$ numerically and faster than the method proposed in Section 2.3.3 is a technique based on QR factorization. In general, the QR factorization is a decomposition of any matrix into an orthogonal matrix $\boldsymbol{Q}$ and a upper triangular matrix $\bar{\boldsymbol{R}}$.

Especially in the least squares prediction sense this means that the regressor matrix $\boldsymbol{\Phi}$ is split into $\boldsymbol{\Phi} = \boldsymbol{Q}\ \bar{\boldsymbol{R}}$. However, to solve the least squares problem with the QR factorization, it is not necessary to compute $\boldsymbol{Q}$. Ljung shows in [12] how this method works:

If the matrix $[\boldsymbol{\Phi}\ \boldsymbol{Y}]$ is QR factorized the following matrix $\bar{\boldsymbol{R}}$ is received:

$$\bar{\boldsymbol{R}} = \begin{bmatrix} \boldsymbol{R}_0 \\ 0 \end{bmatrix} \tag{2.24}$$

where

$$\boldsymbol{R}_0 = \begin{bmatrix} \boldsymbol{R}_1 & \boldsymbol{R}_2 \\ 0 & R_3 \end{bmatrix} \tag{2.25}$$

The matrix $\boldsymbol{R}_1$ has the dimension $n \times n$ and $\boldsymbol{R}_2$ the dimension $n \times 1$. In reference to [12] it is now possible to build the equation system:

$$\boldsymbol{R}_1\ \hat{\boldsymbol{\theta}} = \boldsymbol{R}_2 \tag{2.26}$$

The latter equation system can be solved by the following steps:

$$\boldsymbol{R}_1^T\ \boldsymbol{R}_1\ \hat{\boldsymbol{\theta}} = \boldsymbol{R}_1^T\ \boldsymbol{R}_2 \tag{2.27}$$

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{R}_1^T\ \boldsymbol{R}_1)^{-1}\ \boldsymbol{R}_1^T\ \boldsymbol{R}_2 \tag{2.28}$$

Thus equation (2.28) shows the solution how to calculate the parameter values $\hat{\boldsymbol{\theta}}$ only with the submatrices of $\bar{\boldsymbol{R}}$. The same solution is computed by *Matlab* with the backslash operator:

$$\text{» } \hat{\boldsymbol{\theta}} = \boldsymbol{\Phi} \setminus \boldsymbol{Y}$$

The QR factorization has the advantage that since $\boldsymbol{R}_1$ is a triangular matrix the solution of (2.26) is very fast compared to the solution of the equation system (2.19). Thus, solving the LS estimation problem by QR factorization is much easier and consequently faster to perform.

2.3.6 Example

This section shows an example of an system identification process. The example is a simulated first order process with a time delay of 8 samples (= 2 min):

$$G(s) = \frac{0.62}{4.3\ s + 1} \cdot e^{-2\ s} \tag{2.29}$$

and a SNR of circa 15. The sample rate shall be 15 s and is for all simulated

examples in this work the same.

Figure 2.5 shows the input signal $u(k)$ and the black line in Figure 2.6 the simulated output $y(k)$. The model structure chosen was a simple ARX model ($n_a = 2, n_b = 2$) and the time delay was automatically estimated by the `Matlab` System Identification Toolbox[1] with $k = 18$:

$$y(k) = -a_1\ y(k-1) - a_2\ y(k-2) + b_0\ u(k-18) + b_1\ u(k-19) + e(k) \quad (2.30)$$

Executing the parameter estimation with the least squares method and additionally using the QR factorization yield the parameter values in Table 2.1. This table presents the estimated parameters, which are different from each other but still in the same range. Figure 2.6 demonstrates the quite good approximation of the ARX model without QR factorization (green line). For comparison of the model fit with and without QR factorization the mean square error (MSE) are calculated which are in Table 2.2 presented and are minimally different from each other.

Parameter	Values with common LS method	Values with QR factorization
a_1	- 0.331327	- 0.331835
a_2	- 0.324348	- 0.324529
b_0	0.066069	0.126357
b_1	0.110914	0.050357

Table 2.1. Estimated parameters of simulated process first order.

	Common LS method	LS method with QR factorization
MSE	32.37	32.43

Table 2.2. Mean square error of the ARX model (2,2,18) without and with QR factorization.

2.4 Open and closed loop identification

In the introduction (Chapter 1) it was already mentioned that this project deals with control loops. Hence it is a crucial question how the system identification works in open and closed loop operation.

[1] The estimate is based on a comparison of ARX models with different delays. Name of the function: `delayest`.

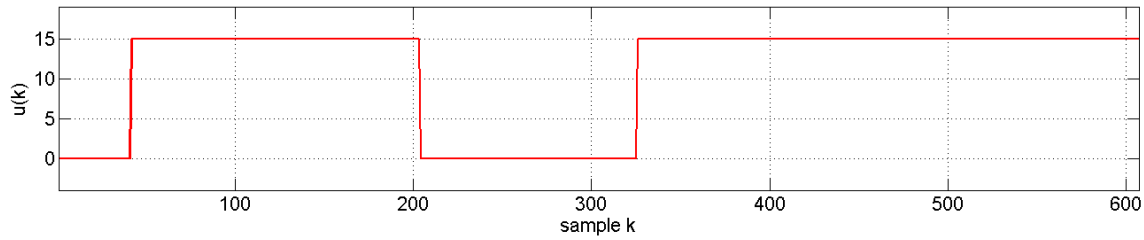


Figure 2.5. Simulated process input $u(k)$.

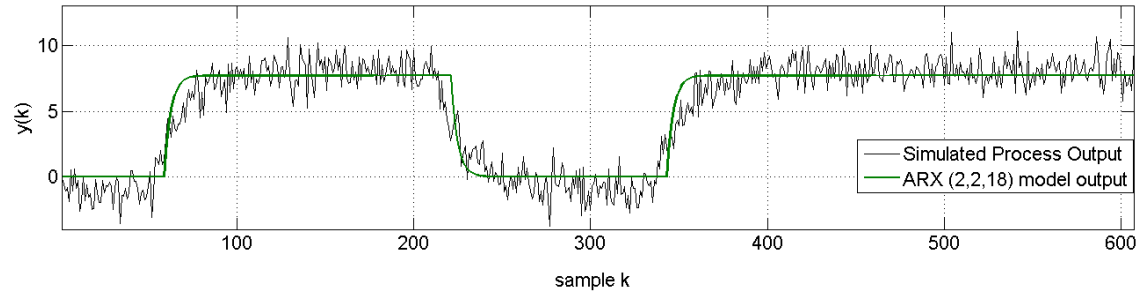


Figure 2.6. Simulated process output $y(k)$ with noise (black), predicted $\hat{y}(k|\hat{\theta})$ with ARX(2,2,18) (green).

Open loop

In open loop operation the controller output $u(k)$ depends neither on the set point signal nor the process output $y(k)$ or any disturbances respectively. Instead, $u(k)$ is set manually by an operator. As long as $u(k)$ is changing in such a way that the process is excited and some response can be observed, a system identification should be possible. The best case would be that the signal $u(k)$ includes some step changes.

Identification experiments are typically accomplished in open loop, but this is of course no option when the system is unstable or because of operational reasons.

Closed loop

In many cases it is disadvantageous to execute system identification experiments in open loop, because of technical or economical reasons. However, in closed loop mode the feedback from the process output $y(k)$ plus disturbances $e(k)$ influences the input.

Many methods for closed-loop identification are based on the identification of a closed-loop transfer function [20]. This closed loop transfer function is built between setpoint or disturbance and an internal signal of the loop (process or controller output). If the transfer function of the controller is known, the process model can then be specified. This method is known as indirect closed loop identification. Silva et al. present additionally in [20] this method using Laguerre series

expansion[2].

The theory for direct closed loop identification is well known (see [12]). If a process should be identified in closed loop, some excitation is necessary, but this excitation is almost every time given because of setpoint changes or external disturbances. In general, as Horch points out in [7], this exciting signal is only restricted to enter the loop outside the shaded area in Figure 2.7 (taken from [7]). Furthermore Ljung shows in [14] that a direct identification under feedback is mostly only successful if the reference signal (setpoint $r(k)$) is changing enough. All in all it is important that the measured output signal $y(k)$ is strongly enough correlated to the input signal $u(k)$.

Thus, in real closed loop applications, the only excitation, which enters the loop in such a way that system identification is possible, are significant enough setpoint signal changes. In this case, a parameter estimation between the input and output of the process with prediction error models like ARX or ARMAX is possible if at the same time a high SNR is met.

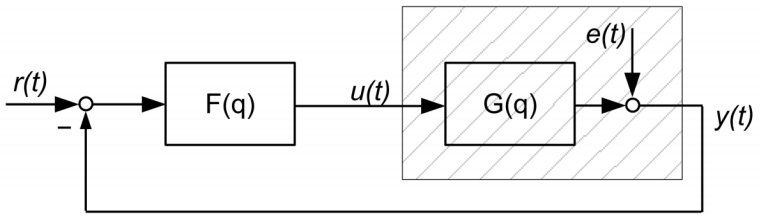


Figure 2.7. In a general closed loop the exciting signal is restricted to enter the loop outside the shaded area.

2.5 Model Properties

2.5.1 Model quality

After the user estimated a model with the LS prediction error method (see Chapter 2.3) he probably wants to know something about its quality. The goal of the model estimation process is to approximate the real system as close as possible. An exact estimate of the "true" model is impossible but the model should be able to reproduce the system behaviour.

The model quality also depends on the requirements of the user. It is not necessary for every application to have the most precise model. For modeling real processes which are often difficult to analyze/model and with large-scale solution spaces, attempts to estimate precise models are impractical, too expensive, or non-existent [1]. In [14] is stated that in terms of quality “confidence is gained

[2]For more information about Laguerre models see Chapter 2.7

in a model obtained with small variations from different measurement data under varied conditions and maybe with different identification methods".

The mean square error (MSE) can be calculated given the model predictions $\hat{\boldsymbol{Y}}$ as:

$$MSE = \bar{E}\left[(\boldsymbol{Y} - \hat{\boldsymbol{Y}})^T(\boldsymbol{Y} - \hat{\boldsymbol{Y}})\right] = B + V \tag{2.31}$$

which is composed of B the bias and V the variance error. $\bar{E}$ denotes in equation (2.31) the expected value.

The bias is practically a function of the model complexity without changing the data. If the model complexity increases, the bias decreases because the model can then better describe the system dynamics. However, it will never reach $B = 0$ unless the model structure can describe all dynamics of the process. Because of the noise influence on measurements and system the variance error occurs. For real collected data, whose length is always limited, the variance V is different with every measurement, even if the input signal and the sequence length is the same. The variance error can be decreased with a longer measurement time and increases with a more flexible representation of model, because of the model enhanced ability to approximate the noise. The latter phenomenon is known as *overfitting* and leads to less prediction accuracy for new data. Hence, both errors should be low for a model with good generalisation properties.

Residual analysis is one possibility to investigate the model quality. When the residuals are determined with equation (2.17) these are in the ideal case independent of the input $u(k)$. If not, it gives the advice that the model does not describe all system dynamics. How to perform the residual analysis is well described in [14].

Another test is to check the model's stability: if the same model structure is taken to estimate the model parameters for different data sets and the model reproduces every time approximately the same model properties it is easier to accept this model structure.

A typical test is the model simulation of the systems output. With this test it is examined how well for a *new input signal* the simulated output suits to the measured output of the system. As measured values only the input signal and initial conditions are used to calculate the output signal $y(k)$. The taken outputs results from the previous calculations. Moreover, the model's prediction of the system output can be tested (see [14]). The question is if the model can predict the system output m-steps into the future ($y(k+m)$) with the corresponding measured data set:

$$Z = \{y(k),\ y(k-1),\ \ldots\ u(k+m),\ u(k+m-1), \ldots\}$$

It is assumed that the inputs are known up to the sample $k+m$ and the output up to sample k.

2.5.2 Convergence of the estimate

Considering the prediction error (see equation (2.17)) it is interesting to know what happens if the number of measurement data tends to infinity ($N \to \infty$). The prediction error can be rewritten as

$$\boldsymbol{Y} - \hat{\boldsymbol{Y}} = \boldsymbol{\Phi}\ \boldsymbol{\theta}_0 + \boldsymbol{E} - \boldsymbol{\Phi}\ \hat{\boldsymbol{\theta}} \tag{2.32}$$

$$= \boldsymbol{\Phi}\ (\boldsymbol{\theta}_0 - \hat{\boldsymbol{\theta}}) + \boldsymbol{E} \tag{2.33}$$

As Ljung derives in [12], the estimation is unbiased under the assumption that $\boldsymbol{E}$ is white noise, with variance λ. With increasing number of data the parameter estimation vector $\hat{\boldsymbol{\theta}}$ converges to the optimal vector $\boldsymbol{\theta}_0$ provided that the data is informative enough (definition see [12]). From this follows for (2.32):

$$\lim_{N\to\infty} \boldsymbol{Y} - \hat{\boldsymbol{Y}} = \boldsymbol{\Phi}(\underbrace{\boldsymbol{\theta}_0 - \hat{\boldsymbol{\theta}}}_{\to 0}) + \boldsymbol{E} \tag{2.34}$$

$$= \boldsymbol{E} \tag{2.35}$$

and shows that for very long data sequences the prediction error tends to the noise $\boldsymbol{E}$. How fast the estimated parameter vector $\hat{\boldsymbol{\theta}}$ converges to the optimal parameter vector $\boldsymbol{\theta}_0$ depends for example on the quality of the data set.

2.5.3 Covariance matrix of the model parameters

The estimate of $\hat{\theta}$'s covariance matrix is computed with the following equation [14]:

$$\hat{\boldsymbol{P}} = \hat{\sigma}_e^2\ (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1} = \hat{\sigma}_e^2\ \boldsymbol{R}^{-1} \tag{2.36}$$

where under the assumption that the bias error is zero (see Section 2.5.1) the estimated noise variance $\hat{\sigma}_e^2$ is defined as follows:

$$\hat{\sigma}_e^2 = \frac{1}{N}\sum_{k=1}^{N} \epsilon^2(k, \hat{\theta}) \tag{2.37}$$

The matrix $\hat{\boldsymbol{P}}$ includes on its main diagonal the variances of each parameter i:

$$\hat{\sigma}_i^2 = \hat{P}_{i,i}, \qquad i = 1,\ \ldots,\ n \tag{2.38}$$

and on its secondary diagonals the covariances.

The reliability of the estimated parameters can be measured with their variance. As a rule of thumb a parameter is significant if its value is bigger than two or three times the related standard deviation $\hat{\sigma}_i$. Since the covariance matrix is proportional to the estimated noise intensity $\hat{\sigma}_e^2$ (see equation (2.36)), the significance of the estimated parameters increases in the case of less noise.

2.5.4 Chi-square test

The *chi-square test* makes use of the *chi-square distribution*, which is a continuous probability distribution. The chi-square distribution gives the probability for the sum of ν squared normal independent random variables with mean 0 and variance 1, where ν is called *degrees of freedom* [10].

For a calculated chi-square χ^2 of a sample it can be determined with the help of Table A.1 the probability $(1 - \alpha_c) \in [0, 1]$ that the variance of the sample can be found in the so called *confidence interval*. $\chi^2_{\alpha_c,\nu}$ is then called the quantile of the standard normal distribution with ν degrees of freedom.

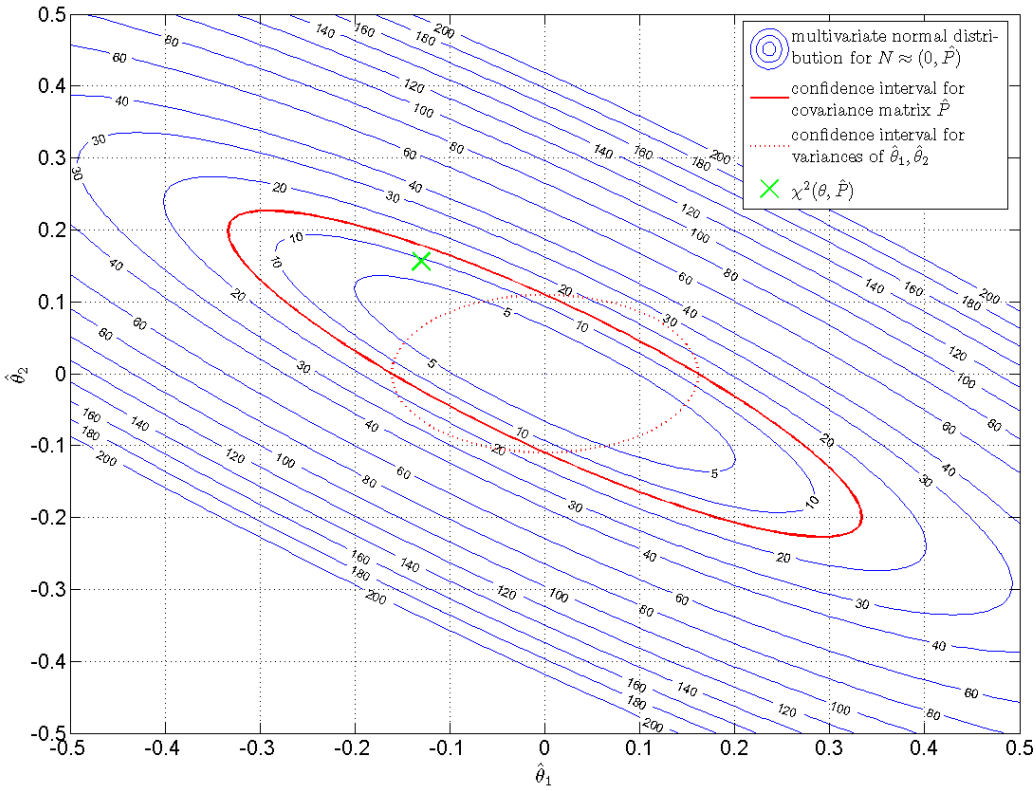


Figure 2.8. Simulated example with estimated parameters $\hat{\theta}_1, \hat{\theta}_2$ and covariance matrix $\hat{\boldsymbol{P}}$. As quantile was chosen $\chi^2_{0.001,2} = 13.82$, which generates two different confidence intervals, dependent on the considered values of $\hat{\boldsymbol{P}}$.

Chi-square test in system identification

The chi-square test is a statistical hypothesis test that verifies if a given quantity follows a chi-square distribution.

It is assumed that $\hat{\boldsymbol{\theta}}$ follows a multivariate normal distribution $\hat{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta}_0, P)$ with expected value $\boldsymbol{\theta}_0$. Concerning the model properties it might be interesting to check whether the estimated parameters in the model are significant enough. This can be expected if $N(\boldsymbol{\theta}_0, P)$ does not approximate $N(0, P)$. Therefore we

formulate the following two hypotheses:

$$H_0: \quad \hat{\boldsymbol{\theta}} \sim N(0, P) \tag{2.39}$$

$$H_1: \quad \hat{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta}_0, P) \tag{2.40}$$

For the v dimension vector $\hat{\boldsymbol{\theta}}$, the hypotheses can be checked with the quantity

$$\chi^2(\theta, \hat{P}) = \hat{\boldsymbol{\theta}}^T \hat{\boldsymbol{P}}^{-1} \hat{\boldsymbol{\theta}} \tag{2.41}$$

which under H_0 will follow a χ^2 distribution with v degrees of freedom. Since this is a binary detection problem, rejecting H_0 is the same as accepting H_1. Therefore, by performing a chi-square test under the quantity in (2.41) it is possible to check if the estimated parameters are statistically significant (see [10]). If $\chi^2(\theta, \hat{P})$ can not approximate the chi-square distribution, the null hypothesis H_0 is rejected and the estimated parameters are significant. This is the case if $\chi^2(\theta, \hat{P})$ in equation (2.41) is bigger than the quantile $\chi^2_{\alpha_c,\nu}$. In this context α_c is called the *significance level*. The division by $\hat{\boldsymbol{P}}$ in equation (2.41) is called whitening of the non unitary variance of the parameters covariance matrix $\hat{\boldsymbol{P}}$.

One advantage of the test is that it checks more precisely the significance of the estimated model parameters compared to the also common method by checking if the sum of parameter variances exceeds a certain threshold (variance check). The latter method means that χ^2 is calculated with $\hat{\boldsymbol{\theta}}$ and the diagonal values of $\hat{\boldsymbol{P}}$ without taking into account the off-diagonal covariances (the correlation between the parameters). In Example 2.1 is the difference between variance check and χ^2-test illustrated.

Example 2.1: Difference between variance check and χ^2-test

Assuming that we have $\nu = 2$ degrees of freedom ($\hat{\boldsymbol{\theta}}$ has dimension 2×1) and demand a significance level of $\alpha_c = 0.001$ then the quantile would be $\chi^2_{0.001,2} = 13.82$ by looking up the chi-square Table A.1.

Figure 2.8 explains now for a given covariance matrix $\hat{\boldsymbol{P}}$ the difference. The blue lines are level curves for a multivariate normal distribution for $N \approx (0, \hat{P})$. The green cross is the computed chi-square $\chi^2(\hat{\theta}, \hat{P})$. The red dotted circle indicates the confidence interval for the variance check and the red solid ellipsoid the confidence interval for the chi-square test. Both use the same covariance matrix $\hat{\boldsymbol{P}}$.

In this example the variance check would lead to the result that the parameters are significant since $\chi^2(\hat{\theta}, \hat{P})$ lies outside confidence interval (red dotted circle). However, the chi-square test would lead to the conclusion that $\hat{\boldsymbol{\theta}}$ is not significant enough, because $\chi^2(\hat{\theta}, \hat{P})$ still lies within the confidence interval of the chi-square distribution (red solid ellipse).

2.6 Model Validation

Model validation is an important part of the system identification procedure (see 2.2). Through the validation the whole modeling procedure is verified and if necessary one or more steps of the procedure are repeated. As long as the vadility of the model is not proved there is no confidence in the model. But what is a valid model? Practically it means that the model is sufficiently good for the intended use of the model [14].

Based on the latter description it is obvious that model validation is closely related to model quality in Chapter 2.5.1, because both consider the quality of the model.

In [11] three main categories for validation are mentioned:

- **Accuracy**: measures how well the model correlates an output signal with the dynamics of the input data. There are various measures of accuracy, but all of them are dependent on the data that is used.
- **Reliability**: assesses the way that a model performs on different data sets. A model is reliable if it generates the same type of predictions regardless of the test data that is supplied.
- **Usefulness**: includes various metrics that inform you whether the model provides useful information. A model might for example be both accurate and reliable in correlating two quantities with each other, but might not be useful, because the user cannot generalize it due to dependencies on other quantities.

2.6.1 Tests of validity

As already mentioned, validation is in a close connection to the related application of the model. Hence, this should be kept in mind when executing a validity test.

The most obvious test is the comparison of the simulated model output with the real process output for the same input signal. This test considers the difference between measured process output $y(k)$ and the model output $\hat{y}(k|\hat{\theta})$ which is called *residual*. Important is here to separate between simulation and prediction of the output (see Section 2.5.1). The better the model is, the smaller are the residuals. A threshold for the difference which separates "good" model from "bad" model depends on what the user demands. Then, for verifying the quality of the model, the squared error mean as already introduced in equation (2.31) has to be computed:

$$MSE = \frac{1}{N}\sum_{k=1}^{N}\varepsilon^2(k) \tag{2.42}$$

The lower MSE the better should be the model [12]. If a comparison of different models is made on data that has already been used for model estimation, one has

to be aware of the fact that a higher complexity of the model will always lead to a better (lower) value of cost function (2.42) because the error decreases with more parameters [2, 14]. Hence, if the data set is rich enough, it is always better to split the data into estimation and validation data.

However, one disadvantage of the last test is that the residuals are not influenced by the input used in the data set Z^N so that equation (2.42) does not represent the whole range of possible inputs.

The *Residual Analysis* is therefore a typical test that checks the covariance between residuals and past inputs:

$$\hat{R}^N_{\varepsilon u}(\tau) = \frac{1}{N}\sum_{k=1}^{N} \varepsilon(t)u(k-\tau) \qquad ,\tau \in \mathbb{N}_0 \tag{2.43}$$

Low numbers for (2.43) would be an indication for "good" quality of the model because the output error seems to be uncorrelated with the input $u(k)$. Hence for other inputs the model would still be consistent. $\hat{R}^N_{\varepsilon u}$ also gives information if there are still properties left which are not explained by the model. This is the case for too high values of $\hat{R}^N_{\varepsilon u}$.

Furthermore the covariance between the residuals themselves can be calculated:

$$\hat{R}^N_{\varepsilon}(\tau) = \frac{1}{N}\sum_{k=1}^{N} \varepsilon(k)\epsilon(k-\tau) \tag{2.44}$$

where $\tau \neq 0$.

Too large numbers of $\hat{R}^N_{\varepsilon}$ indicate covariances among the $\varepsilon(k)$ [12]. This shows that there are still model improvements possible.

Many ways exist to carry out more information about the model quality like *Whiteness Test* (for more see e.g. [12]). Finally, the most important proof of validity is when the model "survives" in real application.

2.6.2 Limitations

In connection with limitations we could speak of a limited domain of validity. This can be related to the following two points:

- model properties at a certain operating point
- accuracy of the model

The first item means that using the model far from the operation point where it was estimated will be more likely to be inconsistent.

The second point refers to the model parameter estimation. As Ljung points out in [14]:

> "A model is never true or correct. At its best it is valid and possibly credible."

The user can only try to find the best approximation by making the right choice of the model structure and taking data which is informative enough.

Another reason for bad model fit are changes of the system properties over the time. Furthermore, the measurements could be inconsistent. Last but not least, disturbances impact the model identification more or less severely and reduce the final model accuracy.

2.7 Laguerre models

Laguerre models are already used in system identification [20, 23]. A detailed overview of parametric signal modeling using Laguerre filters is presented by Wahlberg and Hannan [23]. Furthermore, extensive applications of Laguerre models to control a process can be found in the literature [17, 26, 23], for example [19] presents an application using Laguerre models for automatic tuning of controllers.

Laguerre models can be considered as generalized AR or FIR models where the delay operator is substituted by so called Laguerre filters (see next section).

The system representation by a Laguerre model has the following advantages [20]:

- Laguerre functions have a similar transient response to the usual time delay process.
- In comparison to the ready-made models introduced in Section 2.3.1 less terms are necessary for a good approximation due to its orthogonal property.
- No prior information is required such as process order or time delay which have to be known in ARMA type modeling.
- The representation quantifies the errors due to non modeled dynamics and disturbances.

2.7.1 Model structure and linear regression form

Laguerre models can be used in the FIR or the auto-regressive sense

$$y(k) = \sum_{i=1}^{n} c_i L_i(q, \alpha)\ u(k) + e(k) \tag{2.45}$$

$$y(k) = -\left(\sum_{i=1}^{n} c_i L_i(q, \alpha) \right) y(k) + e(k) \tag{2.46}$$

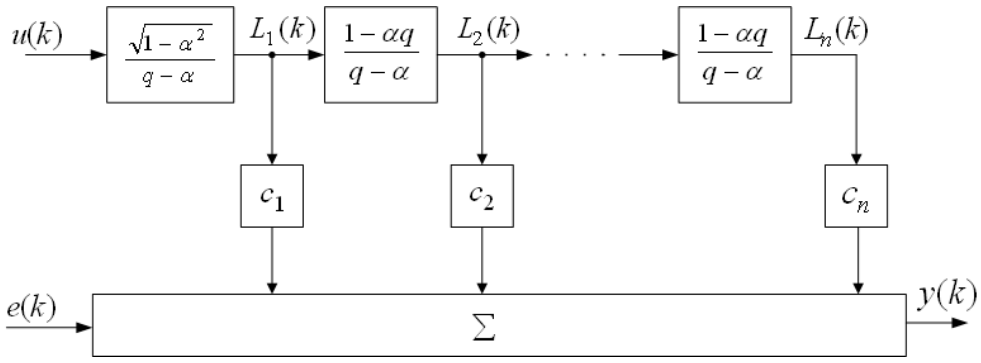


Figure 2.9. Block Diagram Laguerre Network in the FIR sense.

where the Laguerre filters are denoted in the following way:

$$L_i(q,\alpha) = \frac{\sqrt{1-\alpha^2}}{q-\alpha}\left(\frac{1-\alpha q}{q-\alpha}\right)^{i-1}, i \geq 1$$

Figure 2.9 shows the corresponding Laguerre block diagram of equation (2.45). Each Laguerre model represent a low-pass filter in cascade with an $(n-1)$th order all-pass filter. Together the Laguerre model is then of order n where θ_n is the coefficient of the (n)th order Laguerre function.

Since equation (2.45) and (2.46) are linear the parameters θ_n can be estimated by formulating the estimation problem as a linear regression in the way as shown in Chapter 2.3.2 with the restriction that α is given. Otherwise it would be a nonlinear system identification problem.

Therefore we rewrite the Laguerre model (2.45) in the linear regression form:

$$y(k) = \boldsymbol{\varphi}(k)^T \boldsymbol{\theta} + e(k), \tag{2.47}$$

$$\boldsymbol{\varphi}(k) = [x_1(k),\ \ldots,\ x_n(k)]^T, \tag{2.48}$$

$$\boldsymbol{\theta} = [\theta_1,\ \ldots,\ \theta_n]^T \in \mathbb{R}^n \tag{2.49}$$

where $x_j(k) = L_j(q,\alpha)\ u(k), \qquad j \geq 1$

Before computing the filter outputs $L_j(q,\alpha)$ to build the matrix $\boldsymbol{\Phi}$ the initial conditions have to be defined. [23] gives the following possible solutions:

- Set initial values to zero. That is optimal for systems which are in steady state at the beginning.
- Include the unknown initial conditions as parameters to be estimated.
- Set the initial values to zero, but wait until the transients are negligible before computing the least squares loss function.

As a choice of the model order it is recommended to choose a model order rather too high than too low. The model order has essential influence on the time delay, which can be modeled with a Laguerre expansion (see more in the next section). On the basis of this the Laguerre model order was determined for the developed scanning method (see Chapter 5.4).

All in all different sources (e.g. [20]) have suggested some guidelines for selecting reasonable parameters:

- **Measurement time**: $1.2 \cdot t_{step}$ - $1.5 \cdot t_{step}$, where t_{step} is the step response settling time
- **Sampling time**: $T_s \approx t_{step}/10$. An increase of the sampling time T_s leads automatically to a smaller maximal representable time-delay.
- **Number of Laguerre filters**: if the step response is quite oscillatory, a large number is recommended. Therefore ca. 5-10 filters are needed for a low-order process with a large time-delay and 10-15 filters for a high-order underdamped process with time delay.
- **Real pole** α: the goal is to choose α that maximizes $\sum_{i=1}^{n} \theta_i^2$ [25]. Large coefficients θ_i minimize more the loss function in terms of linear system approximations.

2.7.2 Properties and Characteristics

Laguerre models have some interesting properties and characteristics which are worked out in this section.

Wahlberg shows in [23] that for an truncated Laguerre approximation the estimated parameter vector $\hat{\boldsymbol{\theta}}$ tends for large data records N to the "true" (optimal) solution $\boldsymbol{\theta}_0$:

$$\hat{\boldsymbol{\theta}} \to \boldsymbol{\theta}_0, \quad N \to \infty \tag{2.50}$$

By a suitable choice of the Laguerre parameters the user can improve the numerical accuracy of the corresponding linear regression estimation problem [23].

The real pole α determines the transient response of the first Laguerre filter which is a low pass filter first order. Thus, the rise time is influenced by α and therefore the duration the first filter output $L_1(q, \alpha)$ would settle again after a step occurred. The real pole α should be equal to the dominating time constants of the related system [26]. Poles in z-plane close to the unit circle will lead to a slower convergence rate. If the real pole α is selected suitable, then the Laguerre model can efficiently approximate a large class of linear systems. But the correct choice of α is not that sensitive [7]. In general the performance of the approximation does not depend critically on the chosen real pole α [19]. Besides, Laguerre models can not explain oscillations because of its real poles. Thus, they are not suitable to

approximate undamped systems.

The number of filters has as well an influence on the accuracy of the estimated Laguerre model. After the first filter has low pass filtered the signal each further filter time shifts this signal more and more. Thereby, the Laguerre model is able to model a big time delay with only a small number of filters. Together with the choice of the real pole α the user can determine the maximum time delay which can be explained by the Laguerre model. In Appendix B it is shown how to calculate this maximum time delay. This maximum delay time is of course additionally affected by the sampling rate T_s where a higher rate leads for unchanged parameters to a lower maximum time delay. But in principle the Laguerre models are very insensitive to the choice of the sampling rate and robust to the chosen model order [23].

Additionally, the low pass feature of the first Laguerre filter can be beneficial to filter high frequent content of the signal in case these are not within the bandwidth of the system and should not have an influence on the later model estimation process.

2.7.3 Similarities to other model structures

The properties and characteristics of Laguerre models are partly similar with those of ARMAX type models. For a filter pole $\alpha = 0$ the models (2.46) and (2.7) simplify to an AR or FIR model, respectively. Hence, Laguerre models are generalized AR or FIR models.

For high sampling rates or big time delays the delay-operator for ARMAX type models (see 2.3.1) has a too short memory (one sample) and is therefore inappropriate, because an adequate approximation of the system would require a lot of parameters. This delay operator is replaced by Laguerre filters which capture now the behaviour of time delay systems. Hence, a Laguerre model needs less parameters to be estimated for the same system than e.g. an ARX model without increasing the computational complexity. Consequently, the estimation time is lower as well.

For the same simulated process first order as in Chapter 2.3.6 an Laguerre model is estimated with $n = 6$ and $\alpha = 0.8$. Table 2.3 presents the estimated parameters and Figure 2.10 shows the simulation of the process output (black line) compared to the Laguerre model output (blue line). It can be clearly seen, that the simulated time-delay of 8 samples is well explained by the Laguerre model. The MSE of the Laguerre model is 27.61 and even lower than 32.37 for the ARX model.

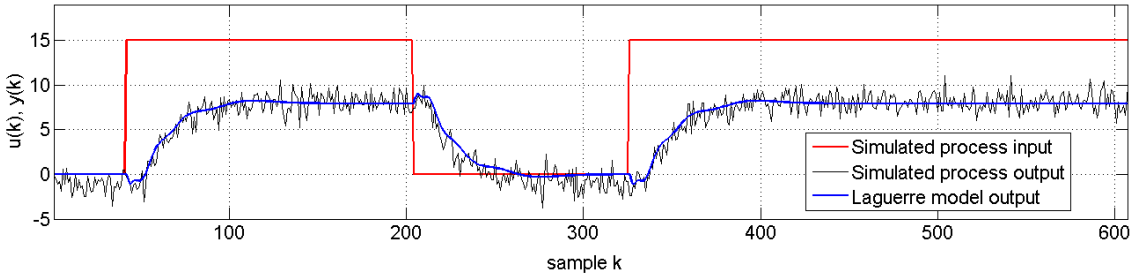


Figure 2.10. Simulation of Laguerre model output (blue line) for the same input (red line) as in Chapter 2.3.6.

Parameter	Value	Parameter	Value
θ_1	- 0.019279	θ_4	0.034019
θ_2	0.082730	θ_5	0.001099
θ_3	0.057483	θ_6	0.020939

Table 2.3. Estimated Laguerre model parameters of simulated process first order.

2.8 Summary

The identification of a system by its observed input and output data is nowadays a common method. Chapter 2 provides all necessary informations related to system identification we will make use of for the following chapters. Several steps of the identification procedure are explained with the background knowledge that the considered real systems are all single input single output (SISO) systems low order. The steps included the choice of the right model structure, building the linear regression, estimating the parameters with the least squares method, and the technique of recursive identification. As an alternative to the common least squares method was presented a technique based on the QR factorization.

Besides that, this chapter considered the possibilities and restrictions of system identification in open and closed loop. Furthermore, the general model properties of an ARX structure and some methods for proving the significance of the model parameters were figured out.

Laguerre models were especially pointed out to have some advantages compared to ARMA structured models concerning complexity (number of parameters), accuracy and sensitivity to high-frequency disturbance. An example of a simulated process identification gives an impression of the model performance.

System identification by parameter estimation of ready-made models has the advantage that no knowledge about the system insights is required although some prior knowledge can help to find a reasonable model structure and to improve the accuracy of the estimated model. Of course a computer with application software

is needed to calculate the different kinds of models. But this could save a lot of time compared to physical mathematical model building.

In summary it can be said that the three basic requirements for model estimation are:

1. An informative data set
2. A proper model set
3. A suitable identification procedure

Chapter 3

Data description

3.1 Overview and description of Perstorp's plant

In this section we will discuss the characteristics of the data set used to identify the several processes and describe the main facts of the considered plant at Perstorp. The data is generated from a chemical plant that processes fine chemicals to one of the intermediate goods of Perstorps product line-up. The data was collected every 15 s forming a data base of circa 37 months with a total size of 6.7 GigaByte. No product change took place over the mentioned period. The manufacturing process is mixed, batch and continuous. The data is filtered using the boxcar-backslope algorithm before it is stored. Moreover, the data is divided into seven different types of processes in a total containing 211 processes. The following table shows which types of processes are present and how much of it:

Process type	Total number
Density	9
Flow	65
Concentration	1
Level	58
Conductivity	2
Temperature	50
Pressure	26

Table 3.1. Overview of process types and their total number.

All control loops are PID and no Model Predictive Control (MPC) is applied. Some control loops have feedforwards and some are cascaded loops. The controllers usually have a sample rate of about 1 s what constitutes almost no restriction for identification purpose, compared to the data collection rate of 15 s. Each data set is associated with the following extra information about the process:

- Type of control loop

- Operation mode
- The range for the process input $u(k)$ and output $y(k)$ values

These informations are used by the scanning algorithm to find useful data intervals. For example, if a model has to be estimated, the model structure could be chosen by the information about the type of control loop. A level control loop should for example consist of an integrator. However, such information should be treated with care, since a generalization of the processes of one type might be critical.

The data set consists of `MATLAB`^{TM} workspace files (*.mat). Each of them includes the data for all control loops of a specific day. The nomenclature of each control loop, the encoding of the mode, statistics about the data and further information follows next.

Nomenclature convention

The name of each control loop (cl) data is defined by 4 variables. The type of controller we have is explained by the first letter of the name:

A = concentration cl	L = level cl
C = conductivity cl	P = pressure cl
D = density cl	T = temperature cl
F = flow cl	

The second letter is mostly a 'C' and can be disregarded. The following three or four digits denote the position in the plant. Close numbers implicate that the control loops are situated in the same part of the plant. The last letter constitutes the type of signal:

u = controller output	r = setpoint value
y = process output	m = operational mode

Example 3.1: An example of data nomenclature

For instance the following data name is given:

`pc2401y`

This signal is from a control loop of type pressure and denotes the process output.

Operational mode encoding

Before decoding the operational mode one has to know that the whole data comes from two different control systems. Approximately 90% of the control loops are installed in an RS3 system and the remaining are in a DeltaV system (both systems are common process control systems supplied by the company Emerson). The systems have the following encoding for their modes:

RS3 system	DeltaV system
0 = local	1 = OOS (out of service)
1 = manual	2 = iman (interlocked)
2 = auto	4 = local
3 = remote	8 = manual
	16 = auto
	32 = remote

Sample time

In general the sampling time should be faster than the system dynamics, otherwise there is information missing for estimation of the model parameters in an accurate way. There exist several rules of thumb for setting a reasonable sampling time like "placing about 5-8 sampling points over the rise time of the interesting part of the system's response" [14] or "T_s must be smaller than the smallest time constant of interest" [3]. However, the present data has a constant scan time of 15 seconds which should be no serious limitation for identification of most of the processes (according to Perstorp in ca. 90% of the cases). We will come back to this again in Section 4.5.

Statistics of the modes

Most of the time the control loops are in automatic mode. Nevertheless the mode can switch to another one as mentioned above. In Section 3.2 it is explained that the manual mode is interesting and useful for process identification. Only 8 control loops have never been in manual mode and those can not be related to a special process type. Figure 3.1 shows the average duration of each control loop type in automatic and in manual mode per day for all 211 control loops and the complete data set. Notice that the flow control loops are in average longer in manual mode than the other loops and the concentration loops stick out the other way around.

Another interesting aspect is the mean number of intervals in automatic and manual mode for each control type. On the basis of Figure 3.2 it can be seen that the different control loop types do not differ so much from each other. Considering 2.965×10^5 as the total number of intervals in both modes and 1.089×10^9 as the total number of samples, gives an impression of how much data the final developed algorithm has to scan.

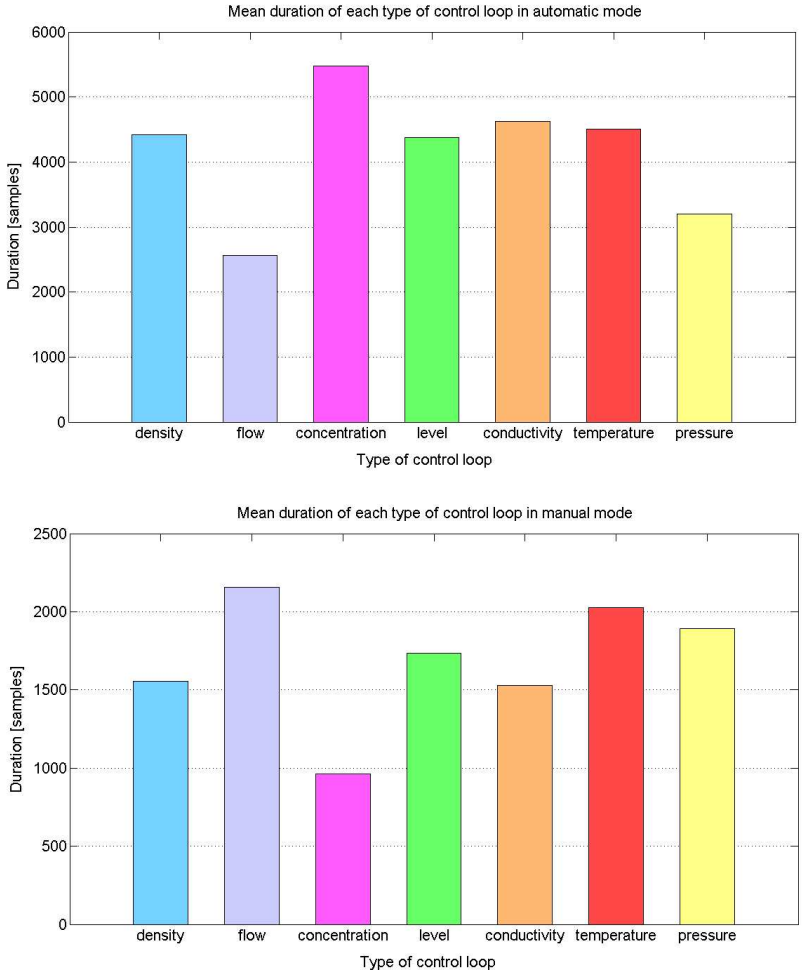


Figure 3.1. Histograms of average duration of each control type in automatic mode (top) and in manual mode (bottom) per day (one day equals 5760 samples). The histograms include all 211 control loops.

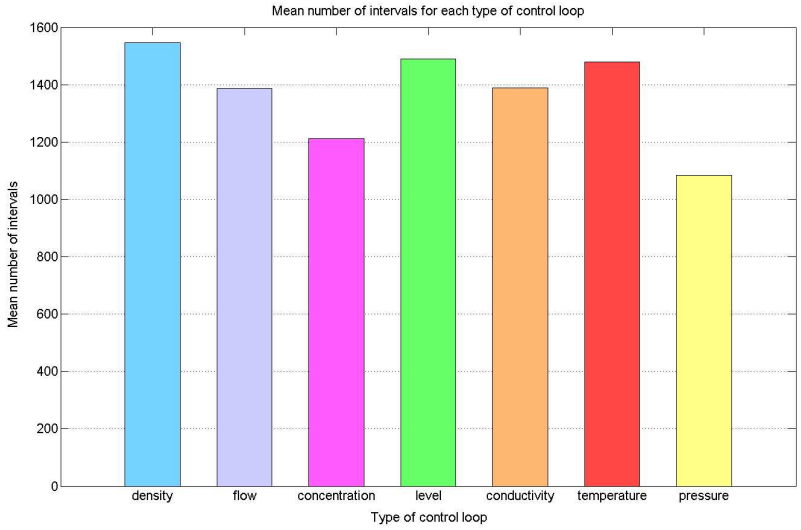


Figure 3.2. Mean number of intervals in automatic and manual mode for each control loop type.

Scaling

The original measurement values for each control loop vary in a certain fixed range which is known by the process control system. Thus, for each control loop the range is a priori given and does not change over the time. For the further processing, the signals should be scaled to a range between 0% and 100%, so that the algorithm can handle all processes. For example, $u_0(k)$ is the original measured value and the measurement range is $[a, b]$, then the scaled value $u(k)$ is calculated by the following equation:

$$u(k) = 100 \cdot \frac{u_0(k) - a}{b - a} \tag{3.1}$$

Time delays

Depending on the process type, time delays are also relevant. For example, the time delay should be known for ARX models estimation. If the time delay is wrongly included in the modeling procedure, then the estimates might be inconsistent. Additionally, the time delay can only be specified as multiple of the sampling time T_s that could deviate from the real value with a maximum error of ± 1 sample.

According to the experience of the optimization group at Perstrop the time delays of the related processes do not exceed more than 10 minutes and mainly level or temperature processes are affected. Figure 3.3 shows a real example of a level process with a time delay of approximately 10 samples ($\approx$ 150 sec = 2.5 min).

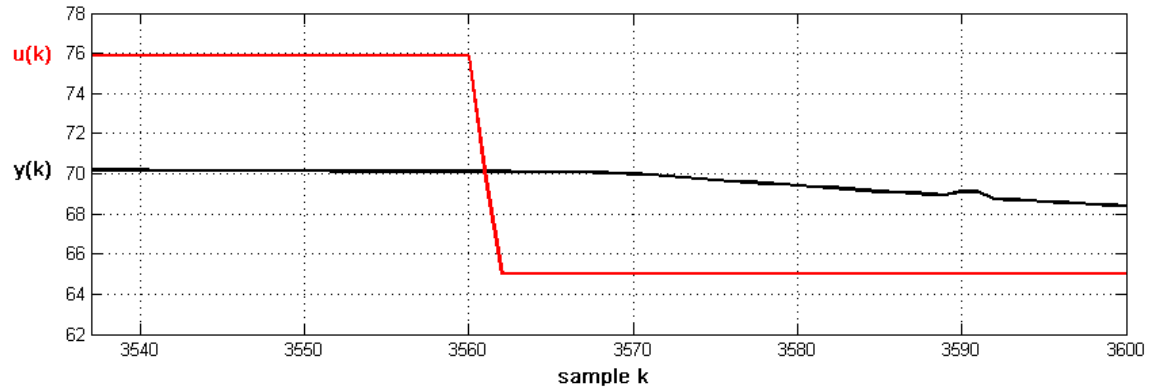


Figure 3.3. Example of a time delay in a level control loop between process input (red) and process output (black).

Different methods to estimate process time delays are proposed in literature. In [5] the time delay is determined by searching the maximal correlation between control signal and change of the process output signal. In [8] is delivered a comparison of five different methods for dead time estimation. A time delay estimation method bases on Laguerre models is proposed by Horch [7] or Isaksson [9].

3.2 Operational modes and their characteristics

The two interesting operation modes are automatic (closed loop) and manual (open loop). How to handle both modes for system identification is described in Chapter 2.4. At this point the characteristics of both modes and how they are related to the process identification task are specified.

Automatic mode

This is the main operation mode of almost every control loop and it is good to know that an identification of the process is possible with the restriction that the reference signal $r(k)$ is informative or exciting enough. However, the problem is the seldom occurrence of a setpoint change since $r(k)$ is mostly constant. Therefore, an efficient scanning algorithm has to be developed that avoids spending to much time scanning data sequences without excitation. Another problem is that the change of the reference signal is often only minimal because only small changes of the process setpoint occurs. Thus, a reliable model estimation is not given or even not possible because no significant response of the process is measurable.

Another favoured method for estimating a process model in automatic mode is to find intervals where $u(k)$ goes into saturation. In this case the loop could be considered like in manual mode and the process input is like a step which enters the process and leads to a response that is often feasible for an identification.

Manual mode

On the one hand the operation in manual mode is much more seldom than in automatic, but the manual mode has on the other hand crucial advantages. In this mode it is much more likely that the process input signal changes, because the operator sets the control signal in this mode since a manual mode operation is typically used for a direct intervention on the loop. In this case $u(k)$ is set manually to one or several values which appear like steps and differ from each other in such a way that a sufficient excitation of the process is given.

For loops where valves are installed valve stiction tests are sometimes performed in manual mode. Those tests are also constituted with input signals that could be exciting for the identification. But this method is rather not useful since the test consists of several successive small steps which forces the process to no useful response for system identification.

Remote mode

This mode is related to the inner loop of a cascaded loop. This loop is also called the "slave" of an outer loop ("master"). In this operation mode the reference signal $r(k)$ of the slave is set by the master. Therefore, in the “slave” loop $r(k)$ changes in a smoother way without big steps, as in automatic mode. Due to the latter, an exciting enough reference signal $r(k)$ (see definition in Chapter 4.1) is less likely.

Because of such peculiarities of the slave loop, it will not be considered further in this work.

3.3 Disturbances

Disturbances exist in any control loop. They enter the loop during process operation or after, when the output is measured. In the data considered in this work, there are mainly two types of disturbances. Noise, with approximately zero mean and small amplitude, conserving a high SNR, and therefore playing little effects in the identification. The second type, are deterministic disturbances due to the several couplings existing between the loops. Consider for example two tanks in series, in which the output of the first is an input to the second. Figure 3.4 shows an example of a level control loop where at around sample 2900 a big disturbance affects the process output $y(k)$ although $u(k)$ is in steady state.

Methods to detect disturbances are discussed in Chapter 4.3.

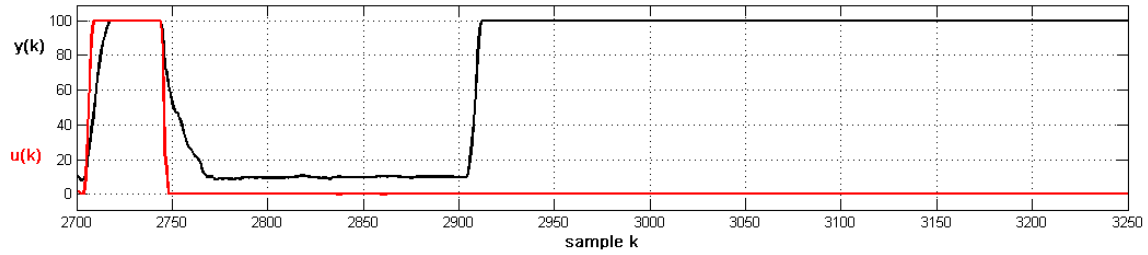


Figure 3.4. Disturbance in a real process. The process input is denoted by $u(k)$ (red) and process output by $y(k)$ (black).

3.4 Summary

The large data base consists of different types of processes, with more or less different dynamics and characteristics. For an automated scanning method, information that would typically be less relevant, might play an important role, for example the limiting levels of a process. Although, the data base is big, most of the data sequences are not useful for process identification due to a steady operation. Hence, an efficient scanning method has to be developed that considers the peculiarities of the data.

The open loop case entails some useful advantages, but since most of the operation time the control loops are in closed loop, both modes will considered for process identification. A couple of processes have a time delay included which could interfere with the scanning method and a lot of disturbances occur which certainly can be a serious problem for process identification.

Chapter 4

Data features for System Identification

4.1 Requirements for system identification

In Chapter 2, system identification was extensively discussed and it was also explained that the accuracy of the estimated model depends on different data features. If the algorithm finds intervals in the data where those features are present, an accurate process identification should be possible. These desired data features are discussed in the following.

As boundary conditions we assume that the processes are linear, SISO, time-invariant and that the closed loop systems are stable. Furthermore, in a real control system, excitation is always present whether in the form of deterministic changes of the setpoint $r(k)$ (automatic) or respectively $u(k)$ (manual) or through external disturbances which can be used for system identification in closed loop [7]. However, external disturbances are unknown (not measured) deterministic signals. We summarize the three possibilities considered for estimating models:

1. Changes of the controller output in manual mode
2. Setpoint changes in automatic mode
3. Saturation of the controller in automatic mode

Data features can now be derived from theory and empirical knowledge which are used to design an identification experiment. Referring to [14], the following points have to be taken into account when an experiment is designed:

- Sampling time
- Signals in the process that should be measured
- Properties of the signals

- Amount of data that is collected

The first item is in our case fixed since the data set already exists. Whether this is a problem for process identification depends on the time constant of the process. Sung recommends in [22] that the ratio of the time constant to the sampling time should be greater than 20 to guarantee an acceptable accuracy of calculating the Laplace transform of the process transfer function. The last three items of the latter list are as well relevant for the scanning method and are discussed below.

Signals to use

First of all a big advantage of the provided data is the fact that it is collected under the conditions for which the model is going to be used later. Which signals amongst the available ones are used by the scanning method is strongly related to the question which signals are necessary for system identification.

In general, for process identification are process input $u(k)$ and output $y(k)$ required, but as already mentioned the setpoint signal $r(k)$ should as well be considered in automatic mode. Those three signals will be used.

Properties of controller output

In the previous chapters it was already mentioned that the information provided by the observed data has to be sufficient enough for process identification which is strongly connected to changes/excitation of the process input $u(k)$. The reason why $u(k)$ is so important results from the fact that only through a change of the process input a system response can be measured at the output.

But what does an exciting enough signal look like? First of all it should change and include a lot of different frequencies. In the perfect case there are steps with a height which is big enough to excite the process. If a user designs an experiment, he would try to create an input signal that excites all interesting aspects of the system dynamic what implies that the input frequencies lie in the system's bandwidth. Moreover, the process should have some time to respond to the input signal which of course depends on its time constants. Therefore, too fast changes of the input or very short pulses are useless when the response is hardly visible.

Methods to determine excitation in data are described in Section 4.2.

Properties of the setpoint

In automatic mode the setpoint $r(k)$ should be exciting enough since this leads to a change of the controller output by disregarding possible disturbances. Therefore, the same properties as for $u(k)$ should be present for $r(k)$.

Properties of the process output

When there is enough excitation at the process input, the process should respond after a certain time depending on the time delay. How the dynamic response looks like is not important at the moment, but the measured signal should be significantly larger than the standard deviation of the noise. A reasonable threshold which was tested is circa three times the standard deviation.

Data length

As already stated in Section 2.5.2, the estimated parameters converge to the "true" values by increasing the data length. The data located closely after a change of $u(k)$ or $r(k)$ respectively has occurred, carries the most useful information due to the process response. Furthermore, an accurate model can not be estimated with a few sample points which contains too little information about the system dynamics. However, the data should only contain data where something happens. Data sequences where all signals are in steady state are useless.

4.2 Excitation in data

It is reasonably easy to describe and to imagine what kind of process input $u(k)$ is necessary to have excitation in the data, but the crucial question is how $u(k)$ can be tested if it is exciting enough or not?

If $u(k)$ is exciting (not in steady state) can be checked by the simply possibility calculating the variance of $u(k)$. An estimate of the variance of $u(k)$ would be:

$$\hat{\sigma}_u^2(k) = \frac{1}{k-1}\sum_{i=1}^{k}(u(i)-\bar{u}_k)^2 \qquad , \qquad \bar{u}_k = \frac{1}{k}\sum_{i=1}^{k}u(i) \tag{4.1}$$

where k denotes the current sample and $\bar{u}_k$ the sample mean. The index $i = 1$ refers to the first sample of the data interval.

Calculating $\hat{\sigma}_u^2(k)$ for each sample would be computationally expensive and slow. Therefore, we use a so called exponentially weighted moving average for $\bar{u}_k$ (Equation (4.3)) and a first-order filter for $\hat{\sigma}_u$ (Equation (4.2)) which calculate the variance of $u(k)$ recursively:

$$v_{u,f}(k) = \frac{2-\lambda_m}{2}\left[\lambda_v \cdot (u(k) - m_{u,f}(k))^2 + (1-\lambda_v)\cdot v_{u,f}(k-1)\right] \tag{4.2}$$

$$m_{u,f}(k) = \lambda_m \cdot u(k) + (1-\lambda_m)\cdot m_{u,f}(k-1) \tag{4.3}$$

where $0 < \lambda_v \leq 1$ and $0 < \lambda_m \leq 1$.

By the parameters λ_v and λ_m the proposed filters can be tuned. Figure 4.1 shows the calculated $v_{u,f}(k)$ for a step of $u(k)$ with $\lambda_v = 0.9$ and $\lambda_m = 0.99$. Similar to the forgetting factor for performing the recursive update of $\boldsymbol{R}(k)$ (see Chapter (2.22)), the parameters λ_m and λ_v define how fast $v_{u,f}(k)$ rises and falls after a step occurs.

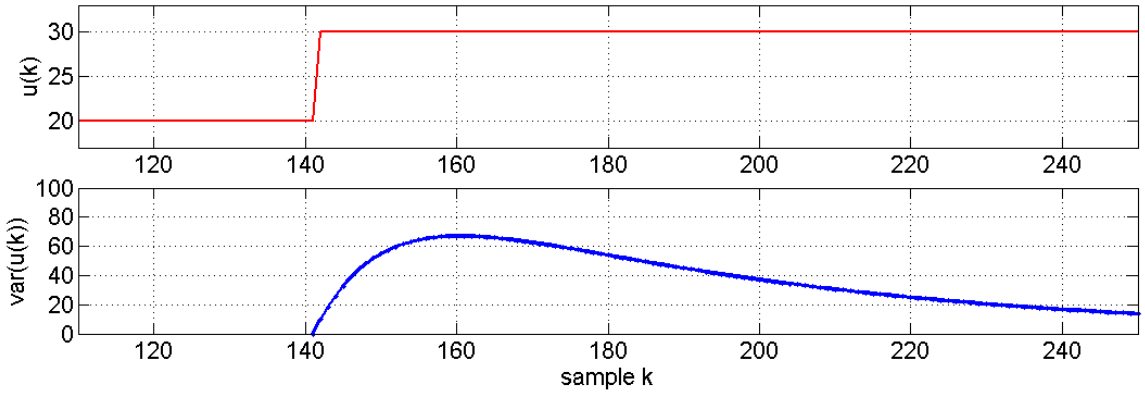


Figure 4.1. Estimated variance $v_{u,f}(k)$ for a step of the input signal $u(k)$.

The more $u(k)$ is changing, the bigger will be its variance $v_{u,f}(k)$. Therefore $v_{u,f}(k) \geq th_v$, where $th_v (> 0)$ denotes a threshold, is a necessary criterion to determine excitation but not sufficient. A reasonable threshold could for example be derived from the estimated noise since for an adequate SNR the variance of $u(k)$ should be larger than the present noise. However, a reasonable threshold related to the estimated noise was not investigated in this work.

Another way is the detection of steps/changes in the input signal $u(k)$. In this case, a threshold is also needed to determine the step size to be detected. A threshold for this could for example depend either on

- a percental value of the range or
- a estimation of the noise standard deviation of $u(k)$ (see equation 4.2).

A step would be considered significant if its size is larger than the threshold. In case of the last item it is often recommended to choose a threshold that is larger than 3-10 times the estimated noise standard deviation [7].

Another test for excitation is checking the rank of the information matrix $\boldsymbol{R}$ (see definition (2.21)). That gives a hint for the informativeness of data. The idea is that if $\boldsymbol{R}$ is rank deficient it can not be inverted for estimating the parameter values with the least squares method. This implies that the interval is not informative enough, for example, when signals are constant over a longer period of time.

Condition Number of information matrix $\boldsymbol{R}$

The condition number of the symmetric, positive definite matrix $\boldsymbol{R}$ is the ratio of its biggest to its lowest eigenvalue:

$$cond(R) = \frac{\lambda_{max}}{\lambda_{min}} \tag{4.4}$$

Applying the singular value decomposition (SVD), which factorizes the matrix $\boldsymbol{R}$, is another easy and fast method to calculate the condition number. Carrette et al. point out in [2] that the accuracy, with which the model parameters are estimated, is then determined by the singular values of the model regressor matrix. MATLAB™ uses as well the SVD when using the command:

```
» cond(R);
```

The idea behind the condition number is similar to check the rank of the information matrix as previously mentioned. The condition number of the information matrix $\boldsymbol{R}$ defines the solution accuracy of a linear equation system. If the condition number is low then it means that in the numerical sense the least-squares problem is well-conditioned. The solution of the equation system is then less sensitive to small variations of $\boldsymbol{R}$ due to the significance of the data. This indicates in turn a certain excitation of the input signal $u(k)$.

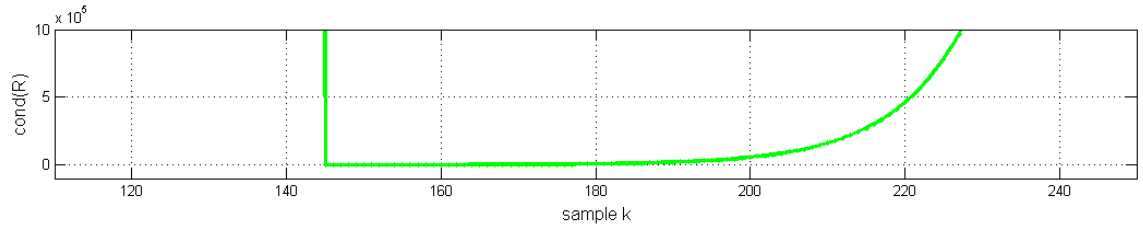


Figure 4.2. Condition number of 4th order FIR with recursive update of $\boldsymbol{R}$ with a forgetting factor $\lambda = 0.9$.

Excitation in the input signal $u(k)$ can now be directly measured by the condition number $cond(\boldsymbol{R})$. A very well conditioned matrix has a condition number close to 1. Figure 4.2 presents the calculated condition number of a FIR model of order 4 for the input signal $u(k)$ of Figure 4.1. The information matrix R was recursively updated with a forgetting factor $\lambda = 0.9$. It shows in a nice way that 4 samples (because of the 4th order) after the step occurred, the condition number goes directly from infinity to its lowest value and increases again steadily due to the recursive update of $\boldsymbol{R}$ where the influence of the step data disappears at first slowly and then faster because of the exponential weighting. Figure 4.3 shows a simulated example of a high-frequency input signal which illustrates that the condition number is in this case as well low.

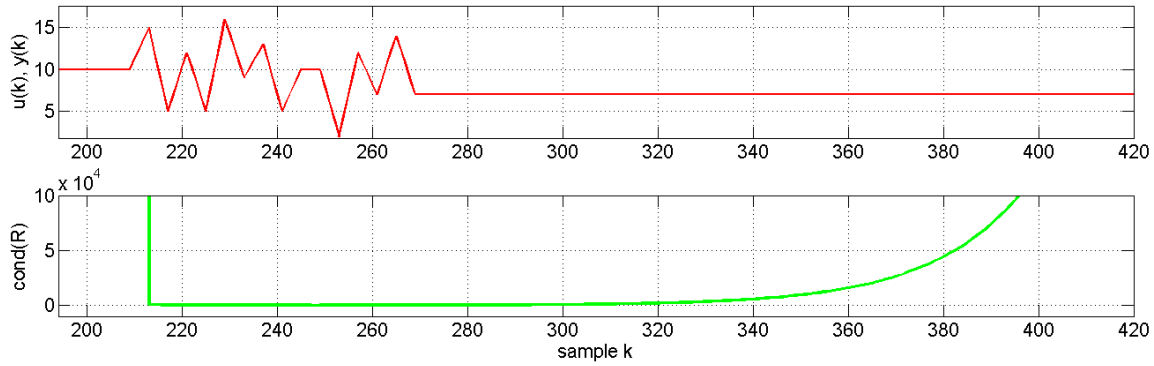


Figure 4.3. Condition number of 4th order FIR for a high-frequency input signal. The recursive update of $\boldsymbol{R}$ is performed with a forgetting factor $\lambda = 0.9$.

4.3 Avoidance of disturbances in system identification

When considering a specific control loop, it makes sense to analyse which disturbances exist. It might be possible to filter all disturbances or parts of it out of the signal. But it has to be kept in mind that a filtering of the input signal also leads to a reduction of the process excitation. However, the challenge is to handle with the final search algorithm disturbances of hundreds of processes with different kinds of disturbances whereby a deep analysis and subsequent filtering of each control loop is not feasible.

Furthermore, since the data set is so large, we use the strategy of avoiding those intervals where disturbances are present.

In terms of disturbance avoidance, several options are possible. In **manual mode** checking if $y(k)$ was in steady state before a change of $u(k)$ and additionally if $y(k)$ settles in steady state after. These would indicate that probably no big disturbances are present. Steady state could for example be determined with the method of Rhinehart and Songling [21]. In their approach they compute the noise variance with two different methods and check their ratio. In case of **automatic mode** the reference signal $r(k)$ should also be in steady state (as well as $u(k)$ and $y(k)$) before an excitation starts. This method is not possible for processes with an integrator since the process output will not go back into steady state after a step arise.

Another possibility in automatic mode is to compare the current variance of $v_{u,f}(k)$ to the variance $v_{u,f,0}$ when the process was in steady state. If the reference signal is constant for a while and $v_{u,f}(k)$ is much larger than $v_{u,f,0}$ then it is likely that disturbances are present.

4.4 Examples of manual system identification

To get an impression of the real data, which is given through the data base of Perstorp, some process models of different control loop types (concentration, flow, density, level, pressure, temperature) were estimated with the MATLAB™ *System Identification Toolbox*. The considered intervals of each process, which are all in manual mode, were found during the project as suitable examples. A good example of a conductivity control loop could not be spotted therefore this type is left out in the following considerations. Figure 4.4 shows all intervals together. The process input $u(k)$ is given by a red line, the output $y(k)$ by a black line and the simulated output of the estimated continuous-time process model by a green line. If necessary a blue dashed line marks the beginning and the end of the manual mode. In Table 4.6 the estimated continuous-time process models with the related MSE are listed. The MSE is given in percentage due to previous scaling of the data to 0% - 100%.

4.5 Typical process models

For the development of the scanning method it is interesting to know which kind of process models come into consideration. Two information sources were used to get an idea of this:

1. Experience of Perstorp's optimization group
2. Manually estimated process models (see e.g. previous chapter)

Both sources confirmed that the maximum time delay of all process models is around 10 min. Flow and pressure processes have almost no time delay. Furthermore, most of the observed processes are approximately first order plus time delay systems apart from flow processes which consists in the majority of cases only of a gain plus time delay. "Approximately" means that in cases where the process has an order higher than one further dynamics can often be neglected due to the dominant time constant.

An integrator is in almost all level processes included and as well in many cases in density, temperature, and pressure processes. This knowledge comes mainly from the experience of the control group at Perstorp.

4.6 Summary

In this chapter features in the data were defined which should be present for identification of reliable process models. Besides a sufficient data length, basic excitation in the input signals $u(k)$ or $r(k)$ (depending on the operation mode) is required and as well a response at the process output $y(k)$. This excitation can be determined by several methods. A computationally fast way of estimating the variance of the input and output signal by filters was proposed. Then, as a

criterion for excitation, a threshold for the estimated variance is set, which has to be exceeded. Furthermore, a more conservative way was suggested by detecting steps in the input signal with a sufficient size dependent on the range of the process values or on the estimated standard deviation of the noise. As an alternative method for measuring excitation in the data, the calculation of the condition number of the information matrix $\boldsymbol{R}$ was described and demonstrated with an example.

Moreover, possibilities for the detection of disturbances in the data were discussed and thereby avoiding intervals for system identification where big disturbances are present. However, the reliability of these methods were not be investigated at this point.

At last, process models for different types of control loop were estimated with the aim to get an idea which kind of process models have to be considered in this work. Additionally, the experience of the control group at Perstorp was used to confirm those estimates. In general, most of the processes can be assumed to be approximately first order plus time delay systems and some of them, like level or temperature loops, have an integrator included.

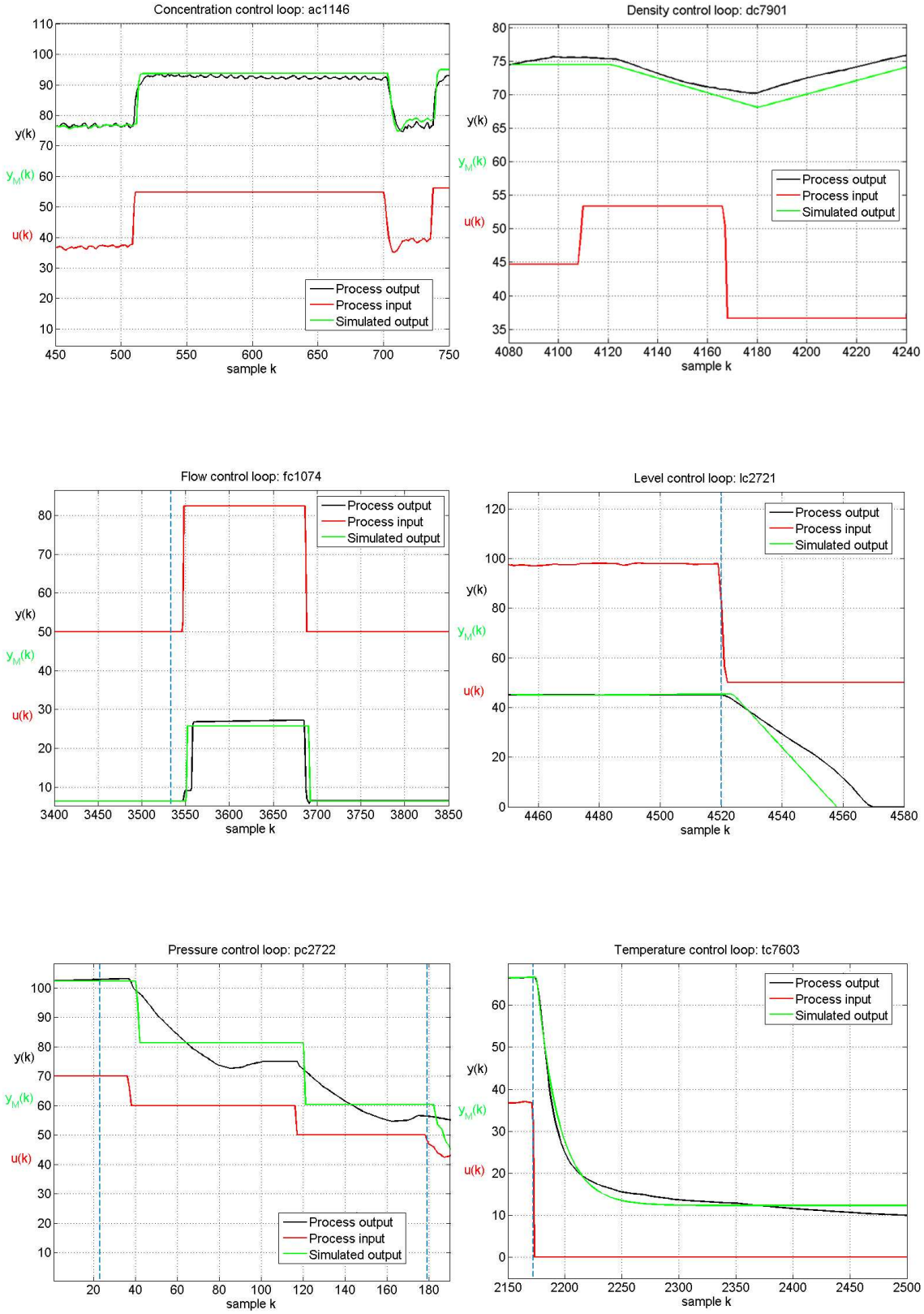


Figure 4.4. Plots of all process examples. A blue dashed line marks in some plots the beginning and/or end of the manual mode.

Type of control loop (Name and date)	Samples	Estimated transfer function $G(s)$	MSE [%]
Concentration (`ac1146`, 4th July 2007)	$509 \leq k \leq 589$	$\frac{0.961}{1 + 0.645\ s} \cdot e^{-0.25\ s}$	$1.33 \cdot 10^{-1}$
Density (`dc7901`, 16th April 2007)	$4107 \leq k \leq 4167$	$\frac{-0.0503}{s} \cdot e^{-2.84\ s}$	$2.92 \cdot 10^{-2}$
Flow (`fc1074`, 19th January 2009)	$3877 \leq k \leq 3962$	0.595	$1.14 \cdot 10^{+1}$
Level (`lc2721`, 11th October 2008)	$4520 \leq k \leq 4567$	$\frac{0.112}{s} \cdot e^{-0.496\ s}$	$7.98 \cdot 10^{-1}$
Pressure (`pc2722`, 22th January 2007)	$23 \leq k \leq 176$	2.1036	$3.12 \cdot 10^{+1}$
Temperature (`tc7603`, 24th November 2008)	$2822 \leq k \leq 2972$	$\frac{1.4756}{1 + 4.883\ s} \cdot e^{-0.324\ s}$	2.20407

Table 4.1. List of the estimated process model examples. The MSE refers to the identification data.

Chapter 5

Developed scanning methods

5.1 Overview

In the previous Chapters 1-4 basic concepts were introduced for the development of a scanning algorithm that finds in the given data set intervals which are useful for process identification. During the work several methods were developed, which are presented in this chapter. Method 1 was the first try and is mainly based on heuristics. Method 2 is more theoretically founded and its main part is the detection of excitation by checking the condition of the information matrix of 4th order FIR. Finally, Method 3 uses the statistical check of the estimated parameters of a Laguerre model.

In the following sections the three methods are presented in detail by explaining the procedure, showing examples and assessing the performance. The main focus is on Method 3, since this one is the delivered method. Advices on how to tune the parameters are given. Furthermore, all methods are compared with each other.

5.2 Method 1: Pragmatic approach - finding steps

5.2.1 Procedure

The first developed method is more pragmatic and simple. It follows the strategy a user would apply when he has to find by himself those stretches. Therefore, the developed method is in general rather "rule-based" than theoretically founded.

In the beginning of this work the focus was on manual mode, which has some advantages as described in Section 4.2 and increases the probability to find "nice" steps in the process input $u(k)$ in terms of process excitation. The following method is therefore only developed for manual operation. In Figure 5.1 it is shown the whole scan procedure of the data of one control loop for one day. The

final method would of course execute the presented method for all control loops and the days which are specified by the user.

Description of the algorithm

At first, the data of one day of one control loop is loaded from the storage file and scaled to 0% - 100% (see Equation (3.1)). The scaling is done with the fixed specified range (min/max) of the related control loop. Thus, it can be that the minimum and maximum values of the considered data interval do not reach the complete given range so that the scaled signal has values which are: $0\% < u(k) < 100\%$. The scaling procedure is illustrated by the Example 5.1.

Example 5.1: Scaling of a signal of one control loop

The given signal of one control loop has values which have a minimum of 25 and a maximum of 80. The general range of this control loop is given by the process control system as 10 - 90. After scaling the signal with equation (3.1) the scaled signal has values which lie between 15% and 87.5%.

Then, the algorithm uses the signal where the operation modes are decoded and search for manual mode intervals. If any interval in manual mode was found then the first one is taken and in a next step this interval is completely scanned for relevant steps in the process input signal $u(k)$ which have a size bigger than a predefined threshold th_{step}. A suitable step size threshold has to be chosen which guarantees a response of the process. According to the experience of Perstorp's control group a threshold of $th_{step} > 5$ % is appropriate. After checking if suitable steps are present it is possible that sequences of steps were found. One sequence consists of several steps forming one exciting input signal. In the case of a sequence the steps are merged to "one step". As a criterion to detect such a sequence it is checked if the found steps are close enough to each other. Two steps are merged if the gap between them is lower than a chosen maximum gap (th_{gap}) of 2 min, that is $th_{gap} = 8$ samples.

In the case that at least one big enough step/step-sequence was detected, the algorithm goes on to search for samples where $u(k)$ and $y(k)$ were in steady state (SS) before and after the step occurred. This is done in two ways. For SS detection of $u(k)$ we consider all samples, where $u(k)$ does not change, because $u(k)$ is in manual mode mainly constant, except when the user sets another controller value. In contrast, for SS detection of $y(k)$ the method of Songling and Rhinehart [21] is used, see Section 4.3, which can handle signals with random components (disturbances). This method finds SS through considering the ratio of the noise variance which is calculated in two different ways. The first noise variance $v_{u,f}(k)$ is calculated as in Equation (4.2). The second one can be computed by:

$$v^*_{u,f}(k) = \frac{1}{2} \cdot \left[\lambda_{v^*} \cdot (u(k) - u(k-1))^2 + (1 - \lambda_{v^*}) \cdot v^*_{u,f}(k-1)\right] \tag{5.1}$$

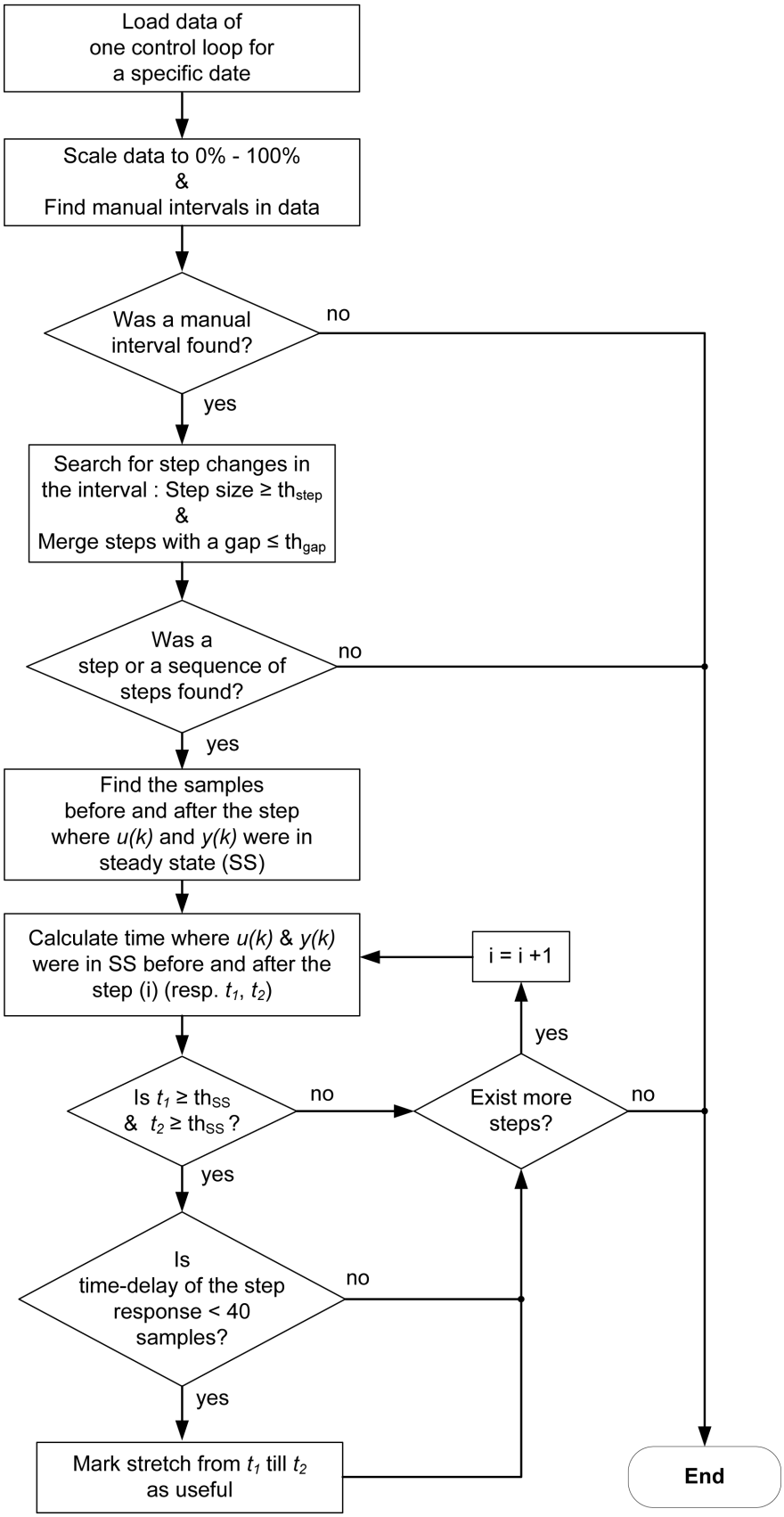


Figure 5.1. Procedure method 1.

The ratio of both variances is then defined by:

$$\zeta(k) = \frac{v_{u,f}(k)}{v^*_{u,f}(k)} \tag{5.2}$$

The data is assumed to be in steady-state, if $\zeta(k)$ is close to one. When $\zeta(k)$ is larger than a threshold th_ζ it is considered not in SS. In some examples [8, 21], where steady-states of process signals are estimated by this method, the threshold is chosen as 2 and the filter values around $\lambda_m = \lambda_v = \lambda_{v^*} = 0.1$. These parameter values were adopted and the classification result has shown to be insensitive to small changes of these values.

The reason for looking after $u(k)$ and $y(k)$ in SS before and after the step is to avoid as much as possible disturbances (see Section 4.3 for more). Therefore, we determine additionally the time durations t_1 and t_2 where $u(k)$ and $y(k)$ were both simultaneously in SS before and after the step. These times have to exceed a given threshold t_{ss} in order for the criterion to be fulfilled.

Finally, a last check is performed which checks if the time-delay of the process response to the step is not bigger than 40 samples. This maximal permitted time-delay is related to the experience of Perstorp's control group. If the time-delay is not too long, the stretch from 8 samples before the beginning of the step till the sample where $y(k)$ goes again into steady-state is marked as useful for process identification. This procedure is repeated with every found step/step-sequence and further manual interval in the data.

The several parameters and thresholds offer the possibility for tuning Method 1, but it is not easy to find the best setting. Some assessments were done to find well suitable values for this method but there is certainly room for further improvements.

5.2.2 Examples

To get an impression of the method's performance, some scans were done with the simulated signal of the process of Section 2.3.6 and with the chosen examples of the last Chapter 4.4. The simulated signal contains noise with zero mean and a variance 1. The SNR is approximately 15. The chosen thresholds for the scans are shown in Table 5.1.

Figure 5.2 shows the scanning result of the simulated manual data. All three steps were detected and as well the main parts of the process response. It is not important that the settling process is marked because the final user will see by himself which parts he takes for later process identification. In case of the real data examples (see Chapter 4.4) the algorithm found only for the flow control loop useful data (see Figure 5.3, above). However, when a scan is performed with the whole data set, the method finds some "useful" data in 144 of the 211 control loop. But as shown in the Figure 5.3 (below), where a density control loop is

presented, it finds as well data stretches which include defective information for process identification.

Value	Description
$th_{step} = 5$	Threshold which determines the minimum size of a step in $u(k)$
$th_{gap} = 8$	Two or more steps are considered as a sequence of steps, if the number of samples between is lower than this threshold
$th_{ss} = 4$	Defines the minimum duration of $u(k)$ and $y(k)$ together in SS before and after the step

Table 5.1. Chosen thresholds for Method 1.

5.2.3 Assessment of performance and results

The results of this rule-based method are not very satisfying. Almost none of the chosen real examples were found by the algorithm. It still finds useful data intervals for process identification but only a small amount. In another attempt it was additionally checked, if $u(k), y(k)$ and $r(k)$ were in steady-state before and after the manual mode to check whether there are disturbances or not. But this led to almost no results.

The main problem is that the method searches for ideal conditions, which are seldom available. Furthermore, it does not work with processes which have an integrator since the output will not go into SS after a step occurred. The detection of steady-state by the proposed method in Section 5.2.1 is not always reliable. If there is too little noise in the signal, the classification is unreliable. Moreover, it only considers the output response with no simultaneous checking of the input signal (verification of a reasonable behaviour). A lot of process models were built with the found data stretch but were often not useful. The following items list the advantages (+) and disadvantages (-):

- + low computational complexity
- + avoids intervals with large disturbances
- - not based on solid theoretical analysis
- - too conservative to require steady state
- - no check if the signals deliver a reasonable process model
- - can not handle an integrator in the process
- - difficult choice of the parameters because of no guidelines

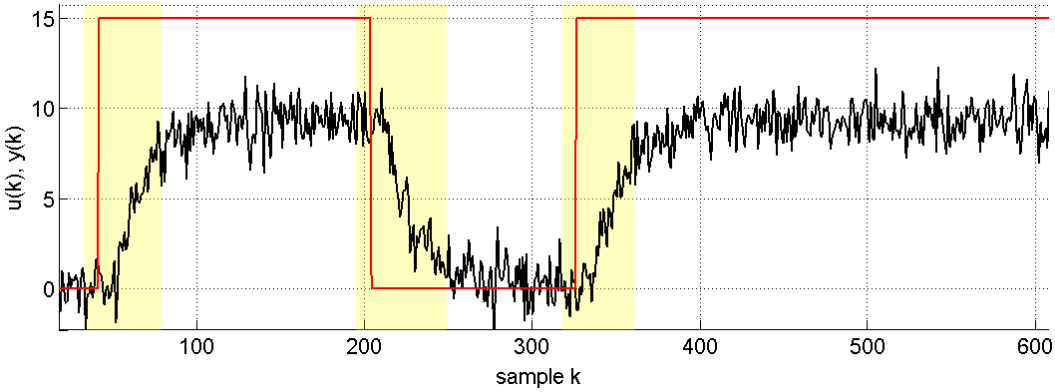


Figure 5.2. Scanning result of simulated manual data with Method 1. The yellow shaded area shows those stretches which were found by the algorithm.

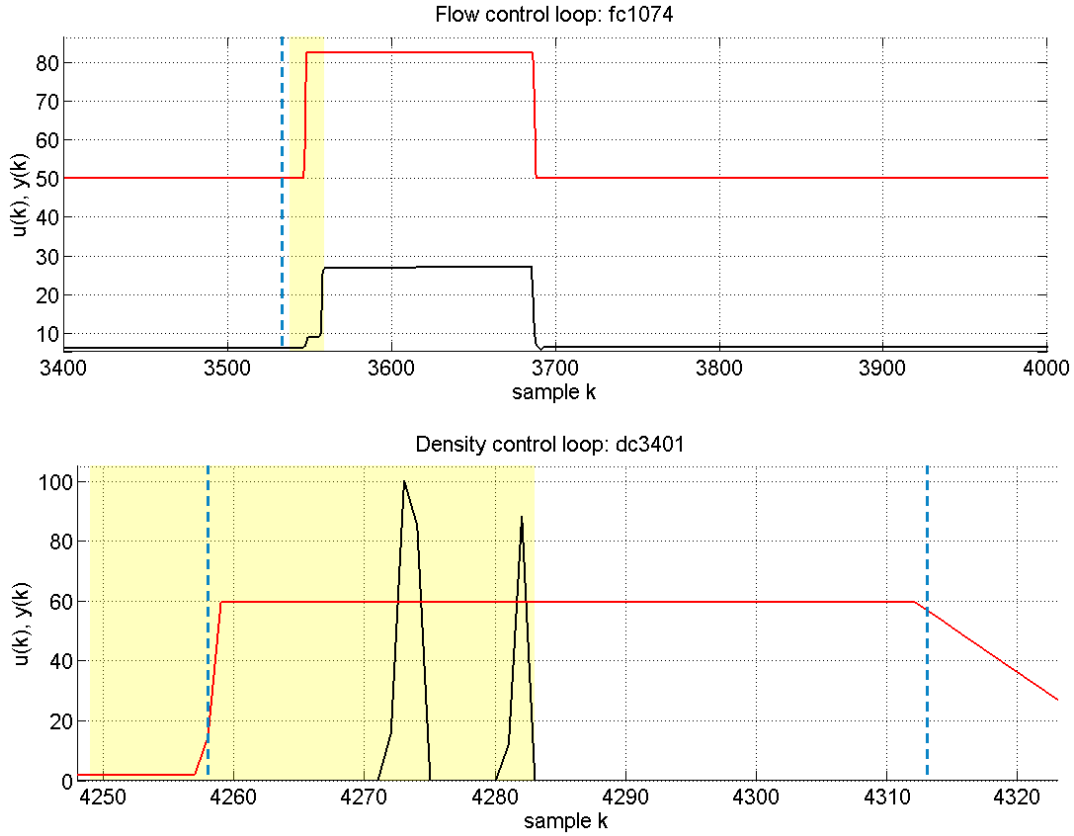


Figure 5.3. Scanning result of a flow control loop example (above) and a density control loop (below) with Method 1. The yellow shaded area shows the stretch that was found by the algorithm.

5.3 Method 2: Measure of excitation by condition number

5.3.1 Procedure

Method 2 was developed after the first method (see Chapter 5.2) led to no satisfying scanning results. One of the main problems of the previous method is that it searches for ideal signal conditions: a step change with steady-state of $u(k)$ and $y(k)$ before and after the step. The new approach is more theoretically based and follows a completely different way of finding excitation in the input signal $u(k)$.

A flow-chart of the developed method is shown in Figure 5.4 which describes again only the procedure for manual operation data of one control loop. An expansion to all control loops could be done without difficulty. The first performed steps of the algorithm are the same as with Method 1:

- *Load data of the specific data*
- *Scale data to* $0\% - 100\%$
- *Search for manual intervals in the data*

In the next step the first value of the corresponding signal is subtracted from the whole signal:

$$u(k) = u(k) - u(1) \tag{5.3}$$
$$y(k) = y(k) - y(1) \tag{5.4}$$

The reasons for the last step are on the one hand avoiding of a too big deflection of the estimated variances (see next paragraph) for the first samples since the variances $v_{u,f}(k)$ and $v_{y,f}(k)$ are initialized with zero and on the other hand avoiding of useless computations as described in the following.

Since $u(k)$ is in manual mode often constant, the algorithm searches the sample k_0 where the first time $u(k) \neq 0$. This avoids useless computations of the further procedure. Afterwards, for each sample from k_0 till the last sample k_N, the filtered variance (see equation (4.1)) of the input/output signals $u(k)$ and $y(k)$ is calculated. This yields the two vectors $v_{u,f}(k)$ and $v_{y,f}(k)$ with a dimension of $(N - k_0 + 1) \times 1$, where N denotes the length of the manual interval. Furthermore the regression matrix of a 4th order FIR model is built:

$$\mathbf{\Phi} = \begin{pmatrix} u(k_0) & u(k_0-1) & u(k_0-2) & u(k_0-3) \\ \vdots & \vdots & \vdots & \vdots \\ u(N) & u(N-1) & u(N-2) & u(N-3) \end{pmatrix} \tag{5.5}$$

The regression matrix has the dimension $(N - k_0 + 1) \times 4$ and each row includes the FIR data at one sample point k. An FIR of order 4 was chosen as being suitable

to detect excitation in the input signal $u(k)$ which was the only intended purpose. No suitable model should be estimated by this model structure.

After the previous steps are executed, the method goes back to the sample k_0 and starts to update recursively the information matrix $\boldsymbol{R}(k)$ which is introduced in Chapter 2.3.4. The recursive estimation is performed as

$$\boldsymbol{R}(k) = \lambda_R \; \boldsymbol{R}(k-1) + (1-\lambda_R) \; \boldsymbol{\Phi}(k, 1 \ldots 4) \; \boldsymbol{\Phi}(k, 1 \ldots 4)^T \tag{5.6}$$

where $\boldsymbol{\Phi}(k, 1 \ldots 4)$ denotes row k and column 1 till 4 of matrix $\boldsymbol{\Phi}$.

The matrix $\boldsymbol{R}(k)$ is initialised with $\boldsymbol{R} = 0$. The information matrix offers now the possibility to check if the data is exciting enough by calculating the conditional number $cond(\boldsymbol{R}(k))$. For the data to be marked as useful for process identification, it has to pass three criteria:

$$\begin{aligned} v_{u,f}(k) &\geq th_{v,u} \\ v_{y,f}(k) &\geq th_{v,y} \\ cond(\boldsymbol{R}(k)) &\leq th_c \end{aligned}$$

The first two criteria require a minimum variance of the input/output signals. The variance of $u(k)$ could be as well received in a filtered form from the first entry of the information matrix: $v_{u,f}(k) = \boldsymbol{R}(1,1)$. However, this variance is dependent on the chosen forgetting factor for the recursive update of $\boldsymbol{R}(k)$. Hence, the variance is calculated separately to be able to tune the algorithm for both purposes (variance and condition number) more precisely. A way to find reasonable thresholds could be for example that the variance of $u(k)$ and $y(k)$ are estimated in steady state and then used as a factor for the threshold. For the last criterion the condition number must fall below the threshold th_c which indicates sufficiently informative data. Currently, all three thresholds are chosen by the experience of the user.

If all three criteria are fulfilled, the algorithm searches the previous sample k_{start} where the variance $v_{u,f}(k)$ exceeded the low threshold th_{start} for the first time which is an indication of the start of the excitation. Then the data stretch from k_{start} till the current sample k is marked as useful for process identification.

The algorithm goes then to the next sample $k+1$, updates the information matrix $\boldsymbol{R}(k)$ and requests the criteria again. This is repeated till the end of the manual interval is reached. The scanning result is influenced by the parameters of the variance filters, the forgetting factor for updating $\boldsymbol{R}(k)$ and by the thresholds. The parameters of the variance filters have to be tuned on the one hand according to time delays and on the other hand according to the behaviour of the condition number. If there is a big delay between $u(k)$ and $y(k)$ present, the fall time of $v_{u,f}(k)$ has to be slow enough so that the filtered variances of $u(k)$ and $y(k)$ still exceeded together the thresholds. Furthermore, in case excitation is present, a too

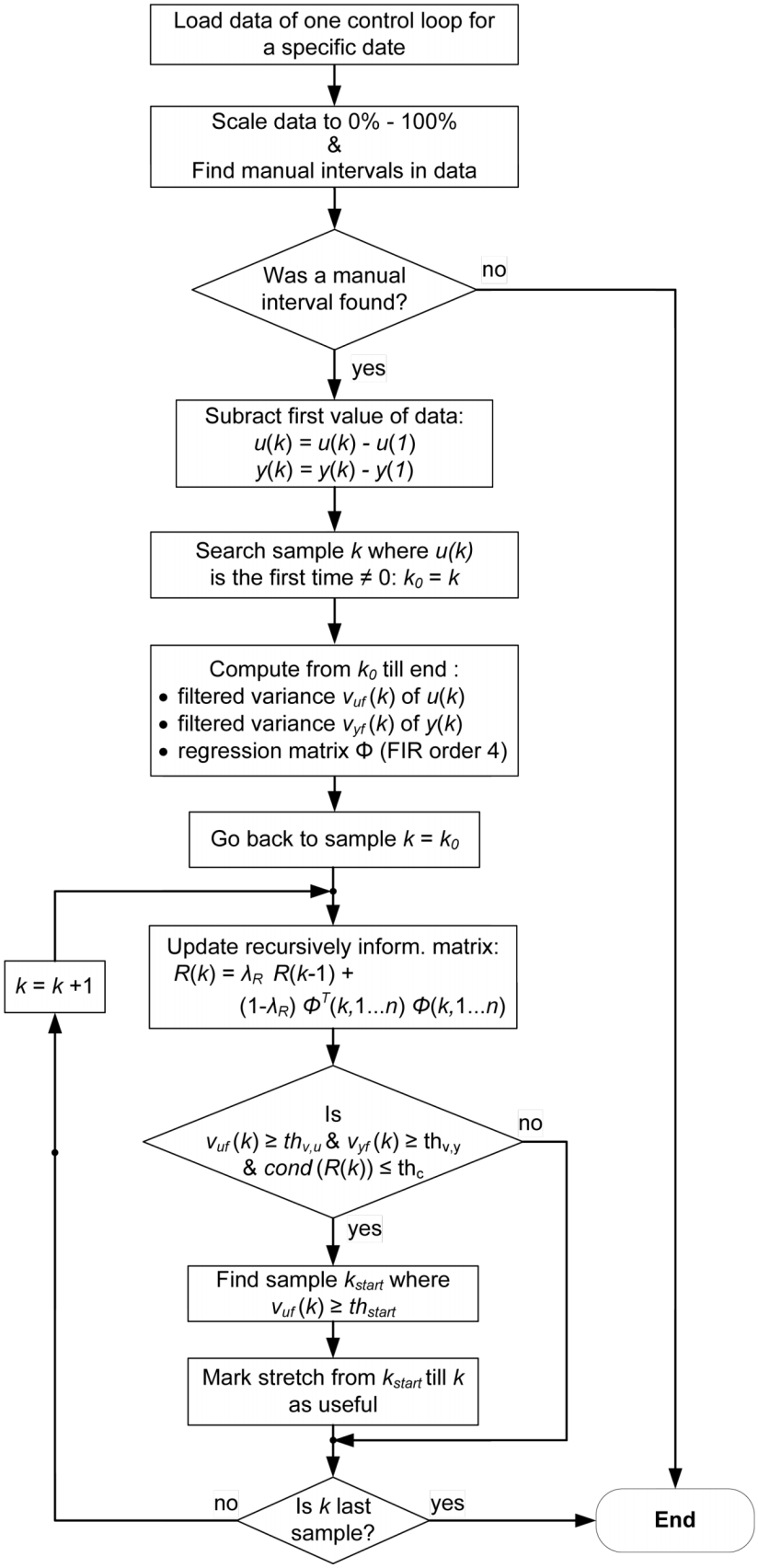


Figure 5.4. Procedure Method 2.

fast/slow increasing and decreasing variance can lead to problems if the condition number is not below the threshold at the same time as the estimated variances are larger than their thresholds. This should be kept in mind, even though the scanning result is not so sensitive to the parameter settings. Compared to this, the choice of the thresholds is much more important.

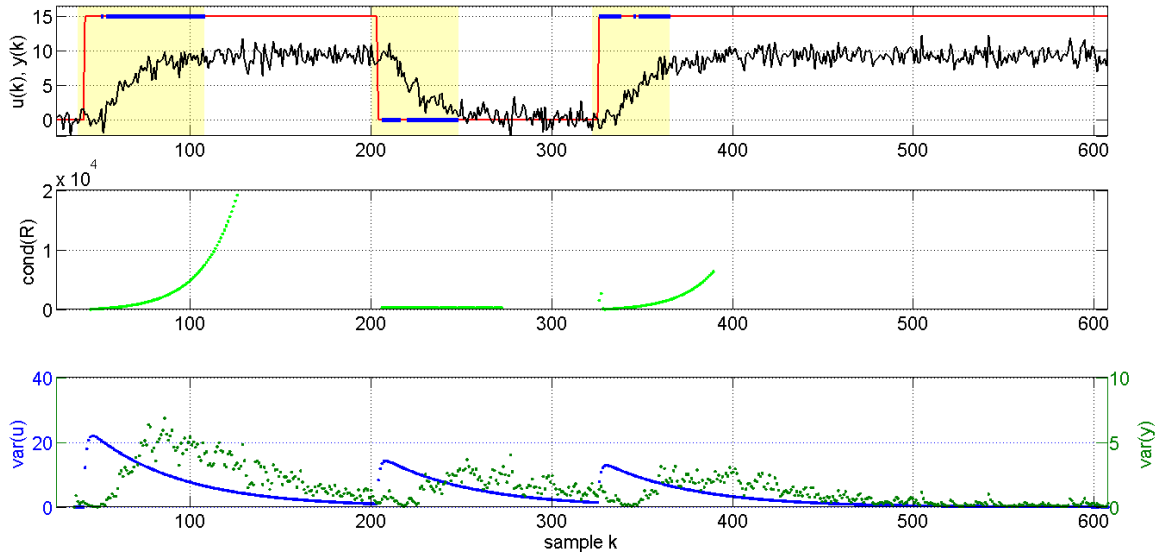


Figure 5.5. Scanning result of simulated data with Method 2. The yellow shaded area shows the stretches which were found by the algorithm. The blue lines mark those samples, where all criteria were fulfilled.

5.3.2 Examples

Method 2 is evaluated with the same example as for Method 1 (Section 5.2.2). Figure 5.5 shows the result of the simulated process. The yellow shaded background in the first subplot denotes the stretches which were marked as useful for process identification by the algorithm. The blue lines mark those samples k, where all criteria were fulfilled. The light blue dashed lines mark the beginning and end of the manual mode. The subplot in the middle presents the calculated condition number. As it can be seen, the condition number has its minimum after the step occurs, except of the second step where it is kept constantly low because of the zero entries in the regression matrix $\mathbf{\Phi}$ for this part of the signal. In the subplot below are shown the estimated variances of $u(k)$ and $y(k)$. If the variances would decrease faster after the step occurs this would lead to smaller marked stretches. The algorithm detected all three steps plus the main parts of the process response. As already mentioned for Method 1 it is not crucial that the complete settling process is marked because the final user will see by himself which parts he takes for later process identification. The parameters used in the example are shown in Table 5.2.

The corresponding scanning plots of the process examples listed in Chapter 4.4 are shown in Figure 5.7 and 5.8. For all examples, useful stretches were found by

the method. However, this method still finds a lot of data, which are obviously not useful for process identification since there seems to be no correlation between the input and output of the process. Figure 5.6 shows such an instance.

Value	Description
$\lambda_m = 0.99$	Parameter for calculating the mean estimation
$\lambda_v = 0.90$	Parameter for calculating the variance estimation
$\lambda_R = 0.95$	Parameter for recursively update $\boldsymbol{R}(k)$
$th_{v,u} = 6.5$	Threshold for the minimum required variance $v_{u,f}$
$th_{v,y} = 0.2$	Threshold for the minimum required variance $v_{y,f}$
$th_c = 1 \times 10^4$	Threshold for the maximum allowed $cond(\boldsymbol{R}(k))$
$th_{start} = 4$	Threshold for finding the start of the marked stretch

Table 5.2. Chosen parameters and thresholds for Method 2.

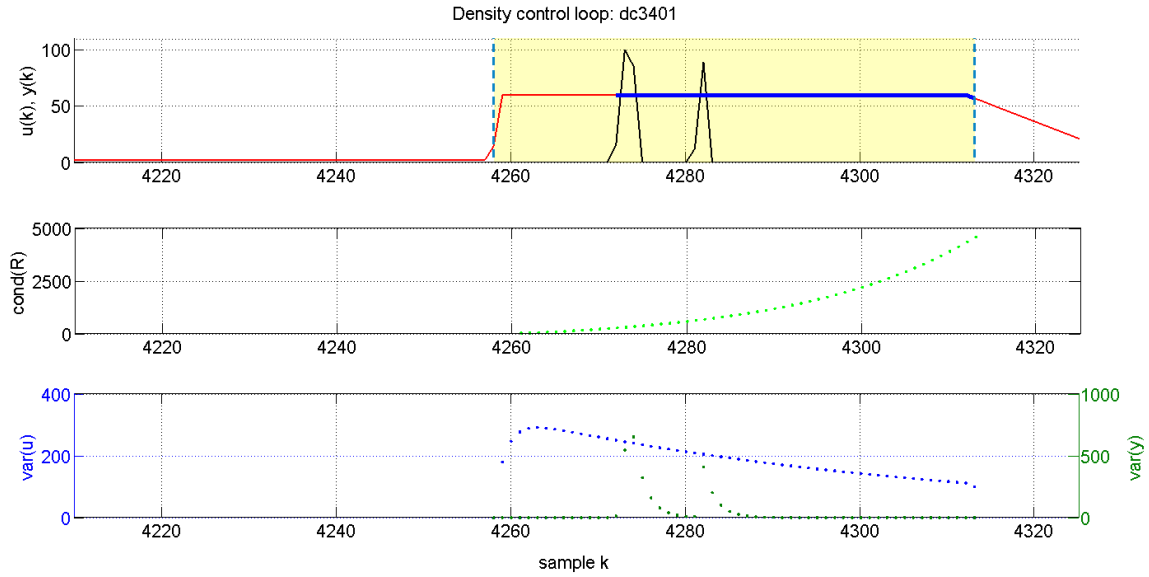


Figure 5.6. Scanning result of a real example which is not useful for system identification.

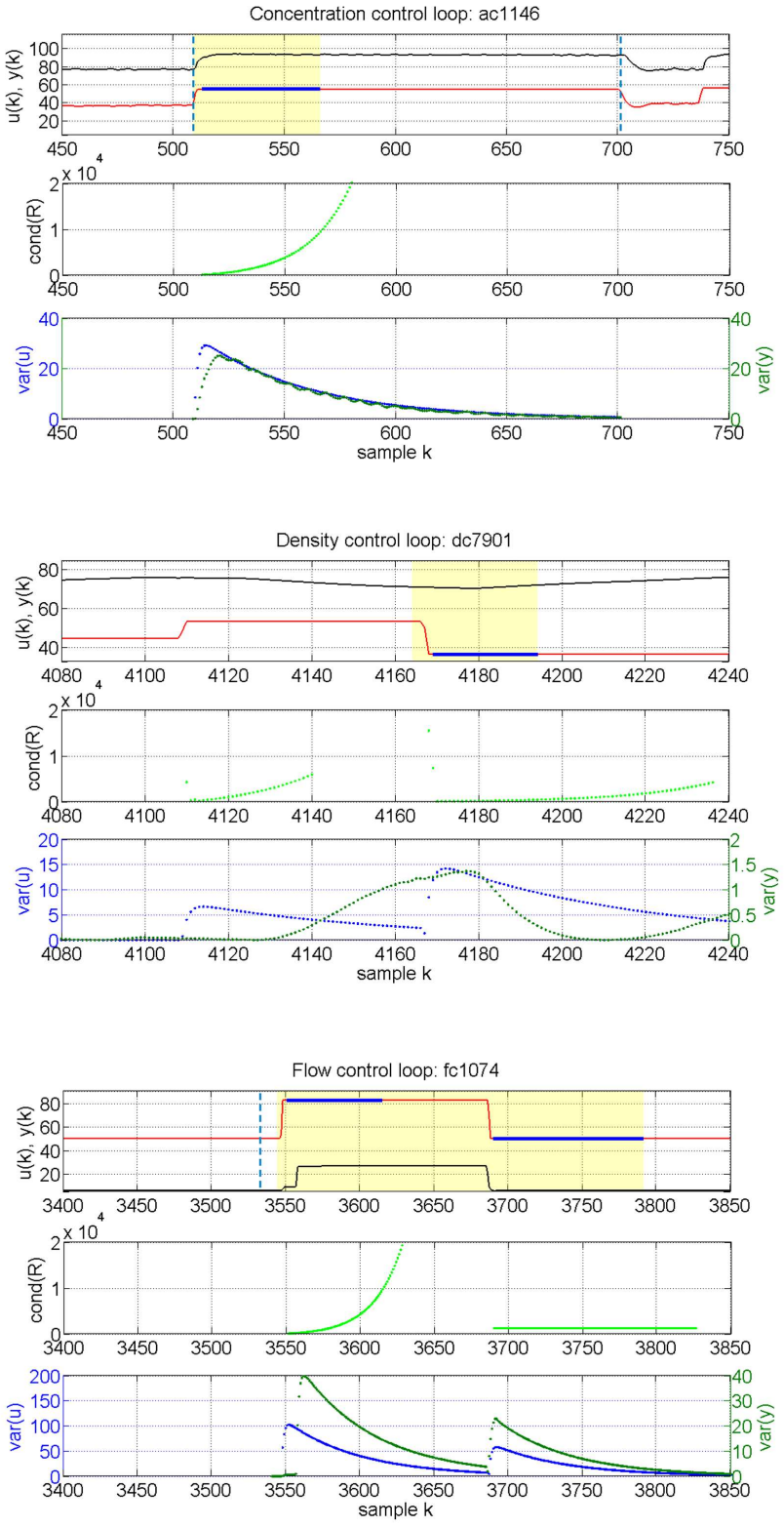


Figure 5.7. Plots of the the scanning results with Method 2 for the process examples concentration, density and flow. A light blue dashed line marks in some plots the beginning and/or end of the manual mode. A dark blue line in the first subplot of each example marks those samples, where all criteria were fulfilled.

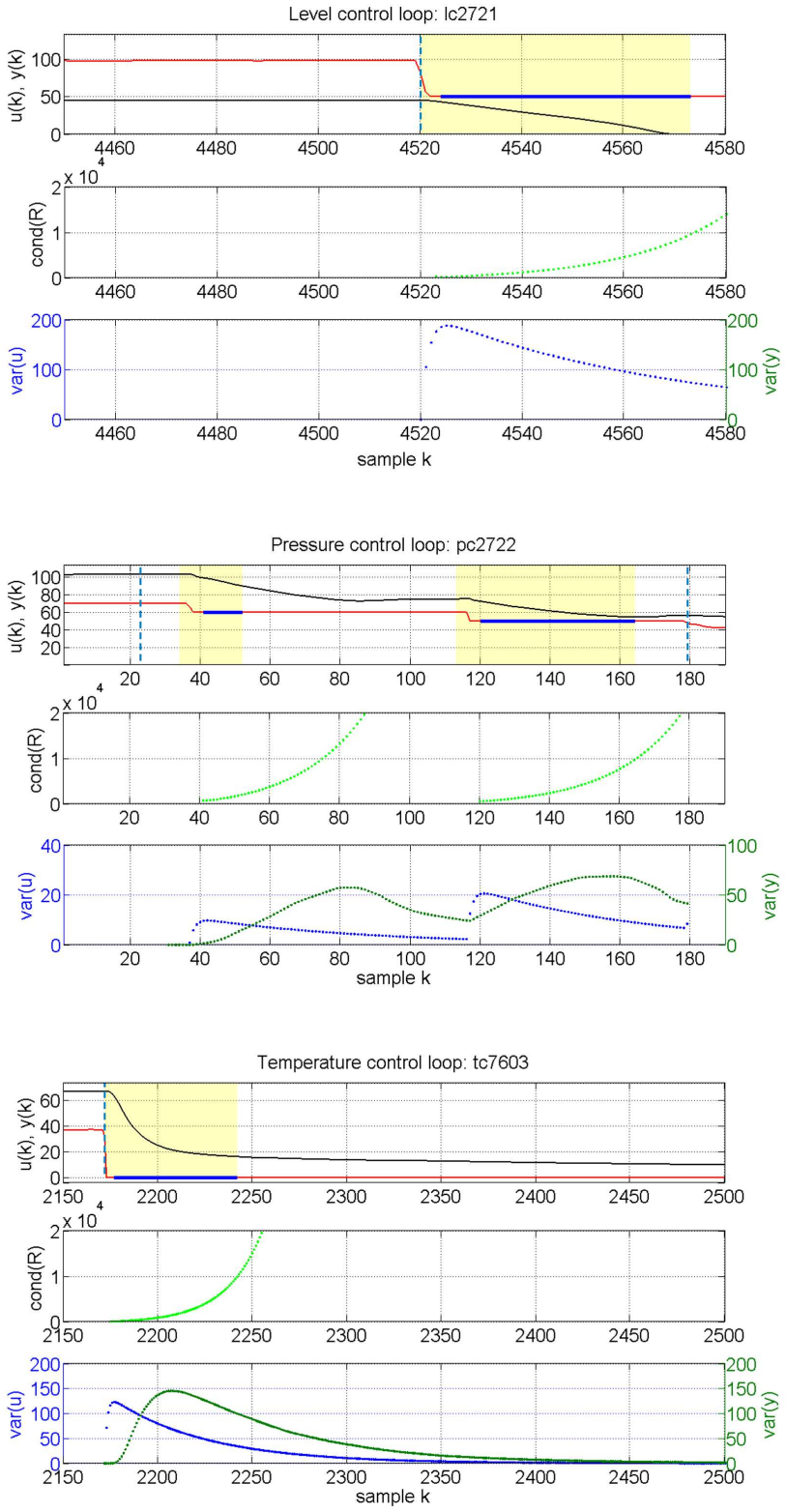


Figure 5.8. Plots of the the scanning results with Method 2 for the process examples level, pressure and temperature. A light blue dashed line marks in some plots the beginning and/or end of the manual mode. A dark blue line in the first subplot of each example marks those samples, where all criteria were fulfilled.

5.3.3 Assessment of performance and results

A hit rate of 100% for the examples points out that the proposed method is able to find something useful in the data. It shows a good performance in detection of excitation in the signal. The main reason for this is the application of the condition number which indicates sufficient excitation in the input signal $u(k)$. However, the condition number gets really low ("well conditioned data") when fast changes of $u(k)$ are present (see example in Figure 4.3) even though this is not useful for process identification since almost no response of the process is measured (see Chapter 4.1).

As for Method 1 the correlation between process input and output is not verified for which reason a lot of data is marked without meaningful usage for process identification, as shown in Figure 5.6. Considering an integrator in the process, this method has no real problems even if the criterion of checking the variance of $y(k)$ is not any more robust since the other two criteria are not influenced by the integrator.

As already mentioned in the description of the procedure the choice of the parameters was only done by experience and can therefore be put into question. It is not so easy to find reasonable values because of a lot interdependencies.

Furthermore, due to big time delays, the calculated variances might not exceed at the same time their thresholds by what not all criteria are fulfilled. Summarized, the main advantages (+) and disadvantages (-) of Method 2 are:

- + theoretical foundation
- + safe detection of excitation
- + can handle integrators
- - does not take into account the correlation between $u(k)$ and $y(k)$
- - has problems with time delays
- - may accept $u(k)$ with too fast changes (high frequencies)
- - difficult choice of the parameters because of no guidelines

5.4 Method 3: Combination of Method 1 and 2 plus chi-square test

5.4.1 Procedure

Method 3 is more complex than the previously presented methods. Besides excitation in the signals it also verifies whether there is correlation between the process input and output signals. Furthermore, this method can scan both operation modes (automatic and manual). The developed procedure is now explained step by step with the simulated example of manual mode data used for the previous methods and introduced in Chapter 2.3.6.

Description of the algorithm

The detection of excitation in the input signal $u(k)$ by computing the condition number of the information matrix of 4th order FIR works reliably as shown with Method 2. However, it classifies the input signal $u(k)$ as a suitable excitation even if the signal changes too fast for process identification. This is one reason for the application of a Laguerre model instead of FIR model since the Laguerre model consists of cascaded filters from which the first one is a low-pass filter. Another reason is that Laguerre models are advantageous for model estimation as described in detail in Section 5.4.3.

In Figure 5.9 the scanning procedure is shown for the data of an open loop operating throughout one day. The first steps are the same as in Method 2:

- *Load data of the specified data*
- *Scale data to* $0\% - 100\%$
- *Search for manual intervals in the data*

Afterwards the initial values of the signals are subtracted (see Equations (5.3) and (5.4)). This is done for the same reasons as for Method 2 (avoidance of too big deflections of the first estimated variance values and of useless computations) and in the case of Method 3 to avoid transients (estimation of Laguerre model). Then, the sample k_0 is searched where $u(k)$ is for the first time different from zero ($u(k) \neq 0$) which avoids unnecessary further computations.

As already mentioned in Chapter 2.7 Laguerre models can not explain integrators. Therefore, if the process type is known to have an integrator, the input signal is in the next step integrated. Next, the following quantities are computed for the whole interval from the sample point k_0 till the last one of the interval:

- All filter outputs $L_n(k)$ for each sample point k
- The estimated variance $v_{L_1,f}(k)$ of the $L_1(k)$
- The estimated variance $v_{y,f}(k)$ of $y(k)$

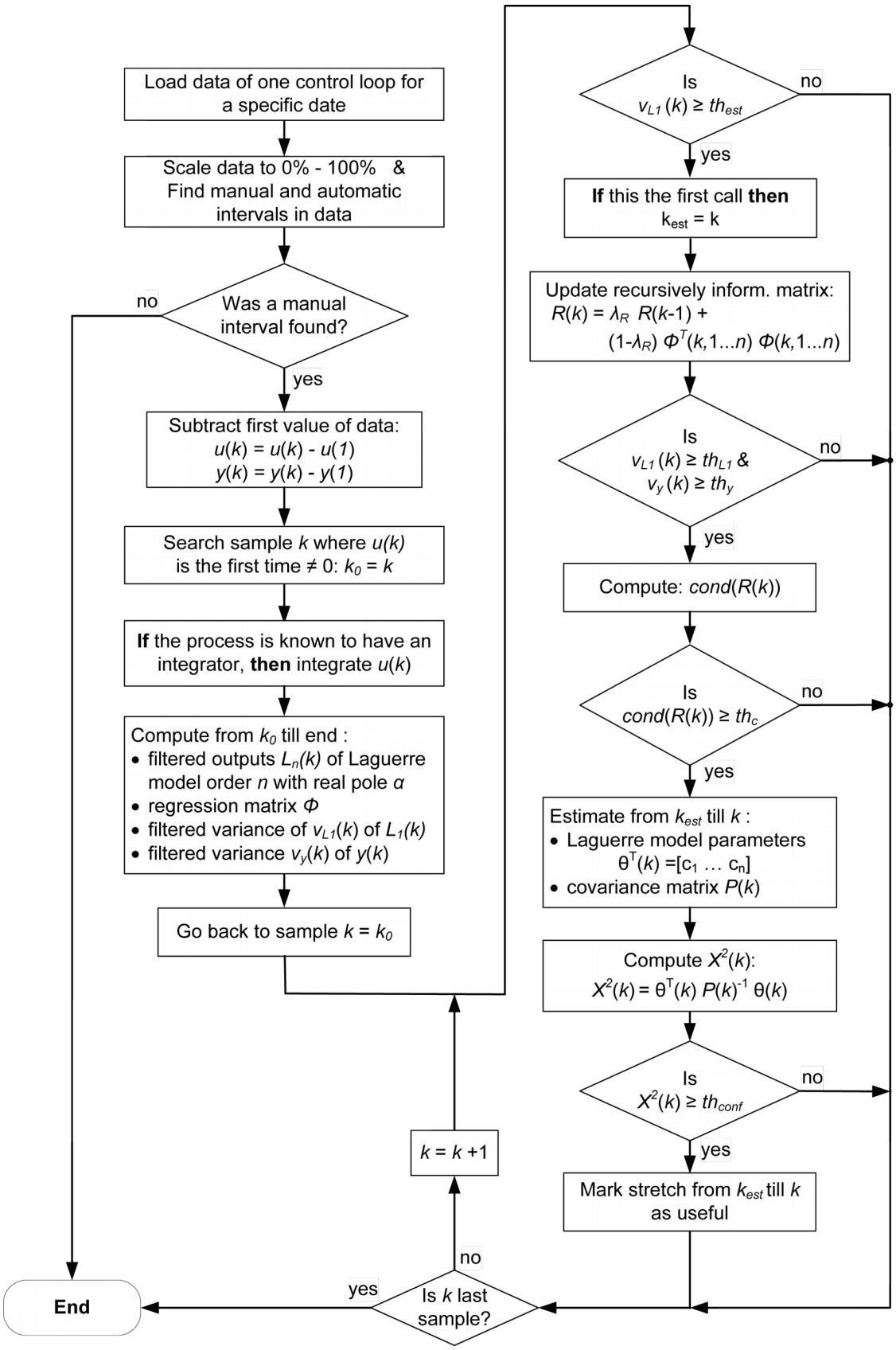


Figure 5.9. Procedure Method 3.

The real pole of all Laguerre filters is chosen as $\alpha = 0.80$ which is a default value in many applications where processes are represented by Laguerre models (see e.g [7], [8] or [16]). In this work the chosen value was successfully confirmed as suitable in a lot of tests. Furthermore, the influence of α on the Laguerre model was in Chapter 2.7 already discussed. Certainly, for an optimal adjustment of Method 3 the choice of α has to be considered more in detail, but otherwise the model quality is relatively insensitive to changes of α. However, much more important is the right determination of the model order in combination with α. The order n of Laguerre filters depends on the maximum time delay that the Laguerre model should be able to model. According to Chapter 3.1 the maximal time delay is circa 10 minutes. With the help of the equations in Appendix B a total of 6 filters is needed. The filter outputs for the example are shown in Figure 5.10, (second plot). The calculated filter outputs $L_1(k) \dots L_6(k)$ form for each sample point the regressor vector $\boldsymbol{\varphi}(k)$ (see Equation (2.48)) which is then used to build the regression matrix

$$\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\varphi}^T(k_0) \\ \vdots \\ \boldsymbol{\varphi}^T(N) \end{pmatrix} = \begin{pmatrix} L_1(k_0) & \cdots & L_6(k_0) \\ \vdots & \ddots & \vdots \\ L_1(N) & \cdots & L_6(N) \end{pmatrix} \tag{5.7}$$

where N denotes the total number of samples of the considered interval.

The regression matrix consists now of all information which is necessary to build the information matrix $\boldsymbol{R}(k)$. Then, excitation in the data can be measured by the condition number $cond(\boldsymbol{R}(k))$ (see equation (4.4)). Certainly, for detection of excitation an FIR model would still be adequate, but in terms of a Laguerre model estimation the condition number of $\boldsymbol{R}(k)$ gives an indication if the data is suitable for parameter estimation and additionally how reliable and accurate a model can be estimated. Furthermore, high frequent changes are filtered out.

The scanning algorithm has now computed the variances $v_{L_1,f}(k)$, $v_{y,f}(k)$ and the regression matrix $\boldsymbol{\Phi}$, with filter parameters which were tuned as $\lambda_m = 0.99$ and $\lambda_v = 0.90$. The algorithm then goes back to the sample point k_0 and starts to request several criteria for each sample in a loop to find useful data. Before any criterion is checked, $v_{L_1,f}(k)$ has to exceed a certain threshold th_{est}, which indicates that the excitation might start. If this is the case for the first time, the current sample is saved as $k_{est} = k$. Afterwards, the regression matrix $\boldsymbol{R}(k)$ is updated for the current sample as

$$\boldsymbol{R}(k) = \lambda_R \boldsymbol{R}(k-1) + (1 - \lambda_R)\ \boldsymbol{\Phi}(k, 1 \dots 6)\ \boldsymbol{\Phi}(k, 1 \dots 6)^T \tag{5.8}$$

where $\boldsymbol{\Phi}(k, 1 \dots 6)$ denotes row k and column 1 till 6 of matrix $\boldsymbol{\Phi}$.

$\boldsymbol{R}$ is initialized as zero matrix for the first sample. The parameter λ_R is taken as 0.95. Afterwards the variances $v_{L_1,f}(k)$, $v_{y,f}(k)$ (see Figure 5.10, fourth plot) are

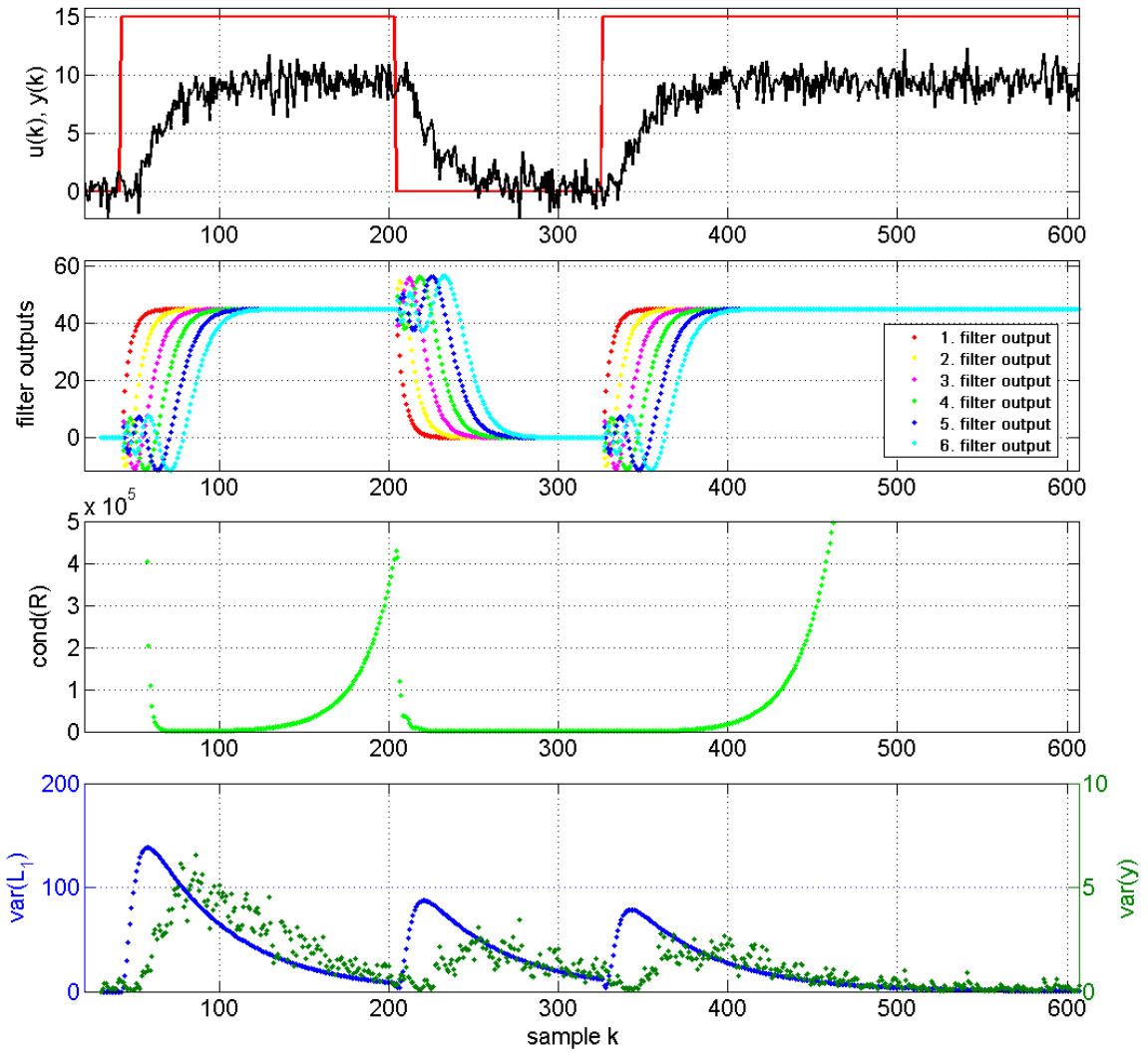


Figure 5.10. 1. Plot: simulated signals 2. Plot: Laguerre filter outputs, 3. Plot: Condition number 4. Plot: Variance of first filter output $L_1(k)$ and of process output $y(k)$.

compared to the thresholds $th_{v,L1}$ and $th_{v,y}$ to check for their magnitude

$$v_{L_1,f}(k) \geq th_{v,L1}$$
$$v_{y,f}(k) \geq th_{v,y}$$

If the last two criteria are fulfilled, the condition number $cond(\boldsymbol{R}(k))$ is computed. The calculated condition numbers of the simulated example are shown in Figure 5.10 (third plot). It is remarkable that the condition number has its minimum after each step occurrence, when the last filter output $L_6(k)$ starts to rise/fall. From the sample point where $L_6(k)$ goes back to steady-state, the condition number increases conspicuously. If the condition

$$cond(\boldsymbol{R}(k)) \leq th_c$$

is fulfilled then the stretch k_{est} till k is considered as exciting enough. However, we do not know if the measured behaviour of the process output $y(k)$ is related to

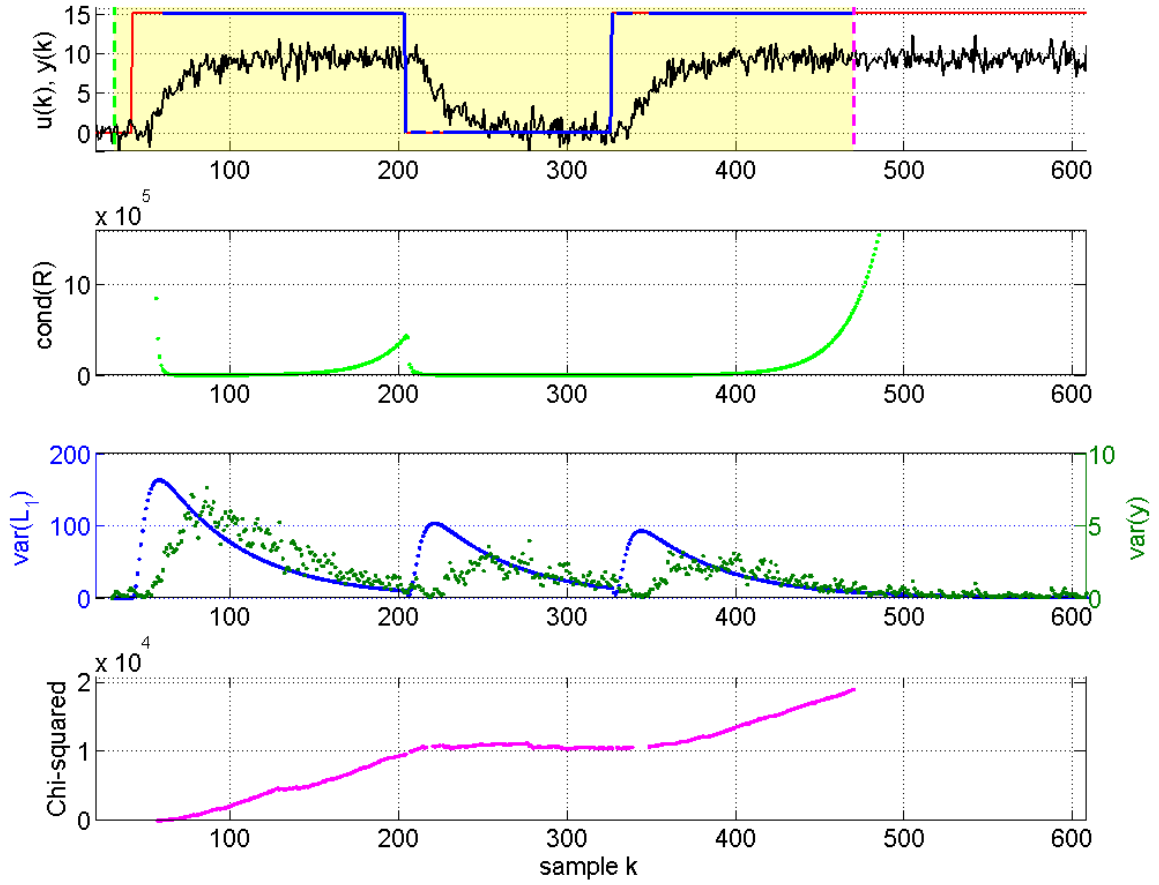


Figure 5.11. Simulated example in manual mode. 1. Plot: simulated signals, 2. Plot: Condition number 3. Plot: Variance of first filter output $L_1(k)$ and of process output $y(k)$, 4. Plot: Chi-square $\chi^2(k)$.

the excitation of $u(k)$ or if some heavy disturbances are present, which were also measured at the output.

Therefore, the correlation between $u(k)$ and $y(k)$ is verified by performing the chi-square test (see Chapter 2.5.4). But before executing the test, the parameters $\hat{\boldsymbol{\theta}}(k) = [\theta_1 \dots \theta_6]$ of the Laguerre model have to be estimated. This is done by taking the data stretch from k_{est} till the current sample k and estimate the parameters:

$$\hat{\boldsymbol{\theta}}(k) = (\boldsymbol{\Phi}_{k_{est}}^T \ \boldsymbol{\Phi}_{k_{est}})^{-1} \cdot \boldsymbol{\Phi}_{k_{est}}^T \ \hat{\boldsymbol{Y}}_{k_{est}} \tag{5.9}$$

with $\boldsymbol{\Phi}_{k_{est}} = \boldsymbol{\Phi}(k_{est} \dots k)$ and $\hat{\boldsymbol{Y}}_{k_{est}} = \hat{\boldsymbol{Y}}(k_{est} \dots k)$.

Additionally, the covariance matrix $\hat{\boldsymbol{P}}(k)$ has to be estimated which is defined as

$$\hat{\boldsymbol{P}}(k) = \hat{\sigma}_e^2 \cdot (\boldsymbol{\Phi}_{k_{est}}^T \ \boldsymbol{\Phi}_{k_{est}})^{-1} \tag{5.10}$$

where $\hat{\sigma}_e^2$ denotes the estimated noise (see Equation (2.37)) under the assumption that the bias error is zero.

Finally, the chi-square test quality is calculated

$$\chi^2(k) = \hat{\boldsymbol{\theta}}^T(k) \cdot \hat{\boldsymbol{P}}^{-1}(k) \cdot \hat{\boldsymbol{\theta}}(k) \tag{5.11}$$

$$= \frac{1}{\hat{\sigma}_e^2} \cdot \hat{\boldsymbol{\theta}}^T(k) \cdot (\boldsymbol{\Phi}_{k_{est}}^T \ \boldsymbol{\Phi}_{k_{est}}) \cdot \hat{\boldsymbol{\theta}}(k) \tag{5.12}$$

In the case that $\chi^2(k)$ lies outside the confidence interval th_{conf}, we consider the estimated parameters as statistically significant enough, that is, the Laguerre model can explain the process behaviour. Moreover, the calculated value of $\chi^2(k)$ is an indication of the estimated model reliability. Samples with a larger $\chi^2(k)$ are preferable. The computed $\chi^2(k)$ for the simulated example is shown in Figure 5.11 (last plot). Finally, the stretch $k_{est} - k$ is marked as useful and the algorithm goes to the next sample $k = k + 1$ and requests the criteria again. The more noise in the data is present the more decrease $\chi^2(k)$. If too much noise is present then $\chi^2(k)$ lies inside the confidence interval th_{conf} and the stretch $k_{est} - k$ would not be marked.

In Figure 5.11 all important calculations concerning the simulated example in manual mode are presented. In the first subplot the samples where all criteria were fulfilled are marked with blue lines. The dashed green line marks k_{est} and the dashed purple line marks the sample where $\chi^2(k)$ was at its maximum. The yellow shaded areas denote the stretches marked as useful for process identification. The other subplots present the calculated condition number, the estimated variance of the first filter output $L_1(k)$, the estimated variance of the process output $y(k)$ and the computed $\chi^2(k)$. The useful part of the example which contains the steps and process response are obviously found by the algorithm. The used parameters and thresholds in Method 3 are given in Table 5.3. Most of the thresholds were found empirically. Amongst others a successful detection of a simulated process response to steps (similar to Figure 5.11 with a SNR of 5) was required. In the next Sections 5.4.2 and 5.4.3 real examples are presented to assess the performance.

Automatic mode

In automatic mode the setpoint $r(k)$ has to be exciting enough, as described in Chapter 4.1. Therefore the only difference in determination of excitation is that the Laguerre filter outputs are calculated for the signal $r(k)$ instead of $u(k)$.

However, a crucial question is if a Laguerre model should be estimated from $r(k)$ to $y(k)$ or from $u(k)$ to $y(k)$. The closed loop system is supposed to be stable, unless the tuning is too bad. Therefore, an estimation from $r(k)$ to $y(k)$ is possible, requires no separation between processes with integrator and without and the regression matrix is already given through the excitation tests. A successful chi-square test would mean that the behaviour of the system is reasonably related

Value	Description
$\lambda_m = 0.99$	Parameter for calculating the mean estimation
$\lambda_v = 0.90$	Parameter for calculating the variance estimation
$\lambda_R = 0.95$	Parameter for recursively updating $\boldsymbol{R}(k)$
$n = 6$	Number of Laguerre filters
$\alpha = 0.8$	Real pole of Laguerre filters
$th_{v,u} = 4$	Threshold for the minimum required variance $v_{u,f}$
$th_{v,y} = 0.05$	Threshold for the minimum required variance $v_{y,f}$
$th_c = 8 \times 10^5$	Threshold for the maximum allowed $cond(\boldsymbol{R}(k))$
$th_{est} = 0.5$	Threshold for finding the start of the marked stretch
$th_{conf} = 22.46$	Confidence interval for chi-square test with $\chi^2_{0.001,6}$ as quantile

Table 5.3. Chosen parameters and thresholds for Method 3.

to $r(k)$ and consequently no big disturbances should be present. But in the end the aim is still to check if the data is useful for an identification of the process, therefore a Laguerre model is estimated between $u(k)$ and $y(k)$. Naturally, this demands again the calculation of the filter outputs.

Speeding up the scanning time

A main requirement for the method's properties is a fast algorithm for a short scanning time. Since the algorithm is currently developed in MATLAB$^{\text{TM}}$, which is specialized on matrix manipulation, it is reasonable to use this strength for speeding up the scanning time. This is done in many ways such as separating the data into manual and automatic intervals.

Furthermore, the regression matrix $\boldsymbol{\Phi}_{k_{est}}$ is often big due to the large number of samples N and therefore the inversion of $\boldsymbol{R}_{est} = \boldsymbol{\Phi}^T_{k_{est}} \cdot \boldsymbol{\Phi}_{k_{est}}$, which is needed for the estimation of the Laguerre parameters and covariance matrix, is really time-consuming. A recursive updating of $\boldsymbol{R}^{-1}_{est}$ (see [12]) would make the method more efficient. A recursive update of $\boldsymbol{R}(k)$ for calculating the condition number would then not be any more necessary, because of the following theorem:

Theorem 5.1 *The condition number of a quadratic matrix is the same as for its inverse. That is*

$$cond(\boldsymbol{R}(k)) = cond(\boldsymbol{R}^{-1}(k))$$

The proof for Theorem 5.1 is given in Appendix C. Due to time constraints this improvement could not be implemented. However, another improvement is currently implemented, which is called QR-Factorization (see Chapter 2.3.5). Calculating

the parameter vector $\hat{\boldsymbol{\theta}}$ by QR-Factorization of the matrix $[\boldsymbol{\Phi}_{k_{est}}(k)\ \boldsymbol{Y}_{k_{est}}(k)]$ was already introduced in equation (2.28). Moreover, the covariance matrix $\hat{\boldsymbol{P}}(k)$ can be reformulated according to the QR-Factorization as

$$\hat{\boldsymbol{P}}(k) = \hat{\sigma}_e^2 \cdot (\boldsymbol{R}_1^T\ \boldsymbol{R}_1)^{-1} \tag{5.13}$$

From this follows that the calculation of $\chi^2(k)$ can be simplified by the following equation:

$$\begin{aligned}
\chi^2(k) &= \hat{\boldsymbol{\theta}}^T(k)\ \hat{\boldsymbol{P}}^{-1}(k)\ \hat{\boldsymbol{\theta}}(k) \\
&\underbrace{\boldsymbol{R}_2^T\ \boldsymbol{R}_1\ (\boldsymbol{R}_1^T\ \boldsymbol{R}_1)^{-1}}_{\hat{\boldsymbol{\theta}}^T(k)}\ \underbrace{\frac{1}{\hat{\sigma}_e^2}(\boldsymbol{R}_1^T\ \boldsymbol{R}_1)}_{\hat{\boldsymbol{P}}^{-1}(k)}\ \underbrace{(\boldsymbol{R}_1^T\ \boldsymbol{R}_1)^{-1}\ \boldsymbol{R}_1^T\ \boldsymbol{R}_2}_{\hat{\boldsymbol{\theta}}(k)} && (5.14) \\
&= \frac{1}{\hat{\sigma}_e^2}\ \boldsymbol{R}_2^T\ \boldsymbol{R}_1\ (\boldsymbol{R}_1^T\ \boldsymbol{R}_1)^{-1}\ \boldsymbol{R}_1^T\ \boldsymbol{R}_2 && (5.15) \\
&= \frac{1}{\hat{\sigma}_e^2}\ \boldsymbol{R}_2^T\ \boldsymbol{R}_1\ \hat{\boldsymbol{\theta}}(k) && (5.16)
\end{aligned}$$

Since the matrices $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are of low dimension, the calculation of $\chi^2(k)$ is now faster. Additionally, according to [12], the noise estimation $\hat{\sigma}_e^2$ can be computed faster by

$$\hat{\sigma}_e^2 = \frac{1}{k - k_{est} + 1}\ |R_3|^2 \tag{5.17}$$

where $(k - k_{est} + 1)$ is the number of samples which are used for parameter estimation.

Nevertheless, the current algorithm could as well be improved by a recursive update of $\boldsymbol{R}_1$, $\boldsymbol{R}_2$ and R_3 which would decrease the scanning time even more.

Dealing with short manual intervals

Sometimes, the reference signal is changed in automatic mode, but the response speed of the process is not satisfying. For this reason, the operator switches the operation mode for a short time to manual, so that he can directly influence the process input and therefore the process output. This takes usually not longer than circa 2 samples.

But, even if the data before and after the manual mode is interesting for process identification, it would probably not appear in the scanning results due to this short manual mode. Hence, manual modes where the mode before and after is automatic and which are shorter than 3 samples, are relabeled as automatic mode and one continuous automatic stretch is formed.

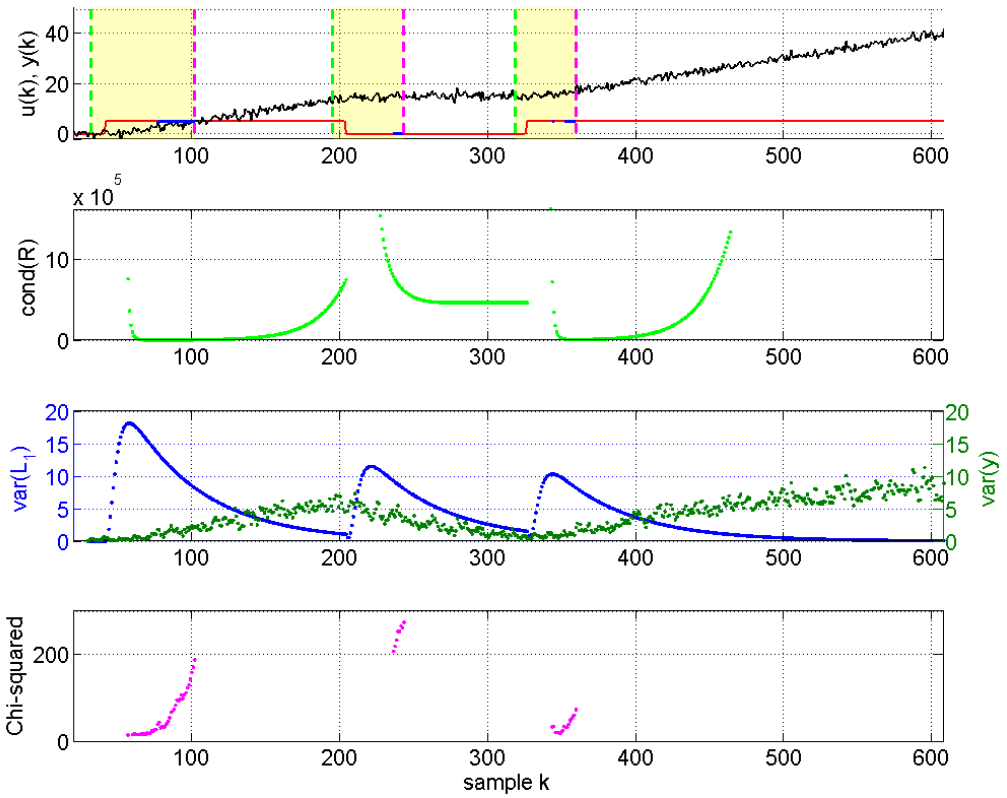


Figure 5.12. Plots of the scanning result with Method 3 for a simulated process which consists only of an integrator.

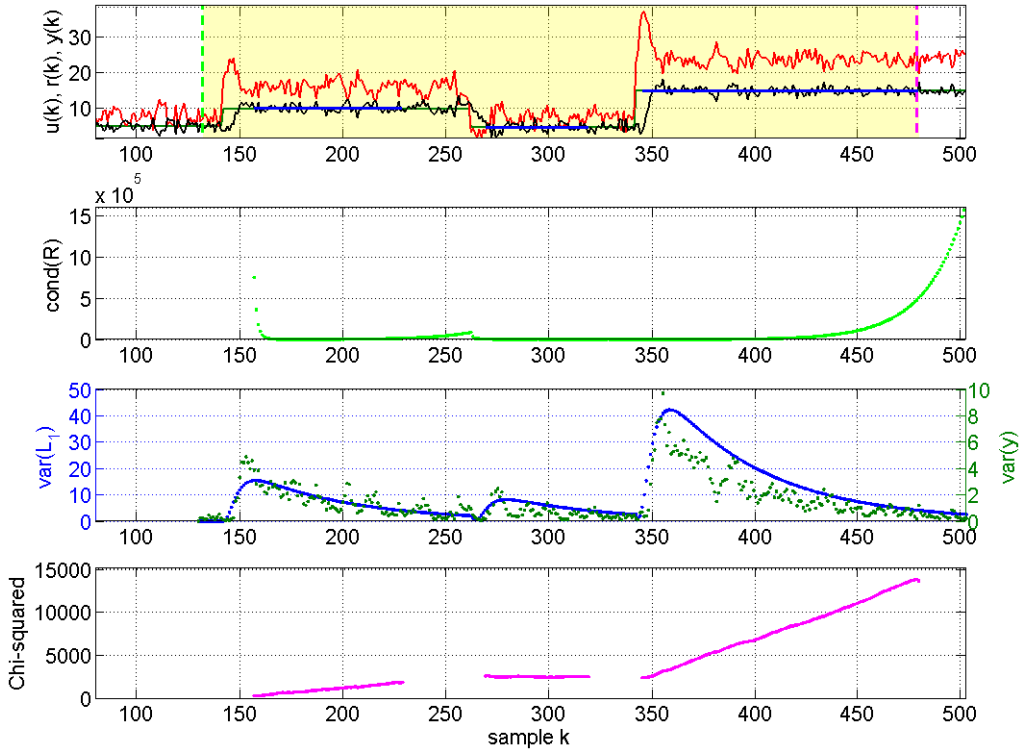


Figure 5.13. Plots of the scanning result with Method 3 for a simulated closed loop with a process first order.

5.4.2 Examples

This chapter illustrates the performance of the developed Method 3 with some examples. In the first subplot of each figure the process input signal $u(k)$ (red) and output signal $y(k)$ (black) are represented and for closed loop cases as well the reference signal $r(k)$ (green). A light blue dashed line marks in some plots the beginning and/or end of the manual mode. Dark blue lines mark those samples, where all criteria were fulfilled. A green dashed line marks the beginning of a useful stretch and a purple dashed line the sample with the maximal chi-square result. The yellow shaded area marks the whole stretch considered as useful. The second subplot shows the calculated condition numbers in the case, that the criteria of large enough variances ($v_{L1,f}(k)$ and $v_{y,f}(k)$, third plot) are fulfilled. The last plot presents the chi-square results $\chi^2(k)$ for the samples, where all previous criteria were already fulfilled.

First of all, Method 3 is tested with two further simulated process examples where the output signal $y(k)$ contains also noise with zero mean and a variance 1. Figure 5.12 shows a simulated example of a process consisting of an integrator and a time delay (8 samples = 2 min) with the equation:

$$G(s) = \frac{0.073}{s} \cdot e^{-2\ s} \tag{5.18}$$

In another example a closed loop with a first order process plus time delay (4 samples = 1 min) is simulated with the process transfer function:

$$G(s) = \frac{0.63}{1.3\ s + 1} \cdot e^{-s} \tag{5.19}$$

and the PI controller equation:

$$F(s) = 1.7\ \frac{1.3\ s + 1}{1.3\ s} \tag{5.20}$$

In both instances the step size of the input signal $u(k)$ or $r(k)$ respectively lies only between 5-10 % but the main important parts including input step and process response are found and marked as useful.

In Figure 5.14, 5.15 and 5.16 the scanning results for all real process examples in manual mode of Chapter 4.4 are demonstrated. In all cases the algorithm found useful data stretches. Interesting is that $\chi^2(k)$ decreases fast in cases where the process goes into steady-state (e.g. temperature loop) or saturation (e.g. level loop). This is favoured since the useless parts of the data should be discard. However, $\chi^2(k)$ decreases sometimes quickly after a step occurred even if there is still a process response which delivers useful information for process identification (see flow loop example). Moreover, in the concentration loop example $\chi^2(k)$ increases more and more even if $u(k)$ and $y(k)$ are already in steady-state. The reason for this is that in the absence of input excitation the parameter fit focuses on the

modeling the noise dynamics by which the parameters become statistically more significant.

The result of the simulated closed loop can be confirmed by another example like the one shown in Figure 5.17 of a density process, where two stretches were found. There, the first stretch is preferable, since $\chi^2(k)$ is much larger than for the second one.

However, Method 3 also finds useless stretches for process identification. In Figure 5.18 such a bad example is illustrated. But compared to the cases where useful stretches are found $\chi^2(k)$ is mostly higher than 1×10^4, the cases with useless stretches have often a much lower $\chi^2(k)$.

5.4.3 Assessment of performance and results

Method 3 seems to be able to detect excitation in the process signals and also frequent changes of $u(k)$ are taken less into account. The important new criterion of this method is the chi-square test. On the one hand, this test decides if the data is generally useful and on the other hand, it indicates how strong the process output is correlated to the input. Therefore, it is possible to assess the ratio of disturbance information which is included in the output signal and the accuracy of a performed process identification. At the moment the threshold th_{conf} is set by looking up a confidence interval in a chi-square table but it might be better to increase the threshold to a larger level. The chi-square test could be done as well with the estimated parameters of another model structure, but the Laguerre model has the big advantage that a time delay of the process does not have to be known previously (see properties Chapter 2.7.2). Moreover, the estimation process is stable and never caused a crash of the scanning system.

It is also able to scan data in automatic mode, even if it requires more computation due to recalculation of the Laguerre filter outputs for parameter estimation. Integrators can be handled by first integrating the input signal $u(k)$ and then estimating the model parameters.

An additional remark is that in all simulated examples the output signal $y(k)$ contains noise with a mean of zero and a variance of one. However, this has no crucial influence on the scanning result.

The possibility of process identification in automatic mode, when $u(k)$ goes into saturation, was not especially considered, hence it is necessary to know which saturation level is set for each controller. But this information is not given. However, it is problematical when $y(k)$ goes into saturation (automatic or manual mode). As shown in one example (see Figure 5.15, flow control loop) the chi-square test is low although this data is useless for process identification.

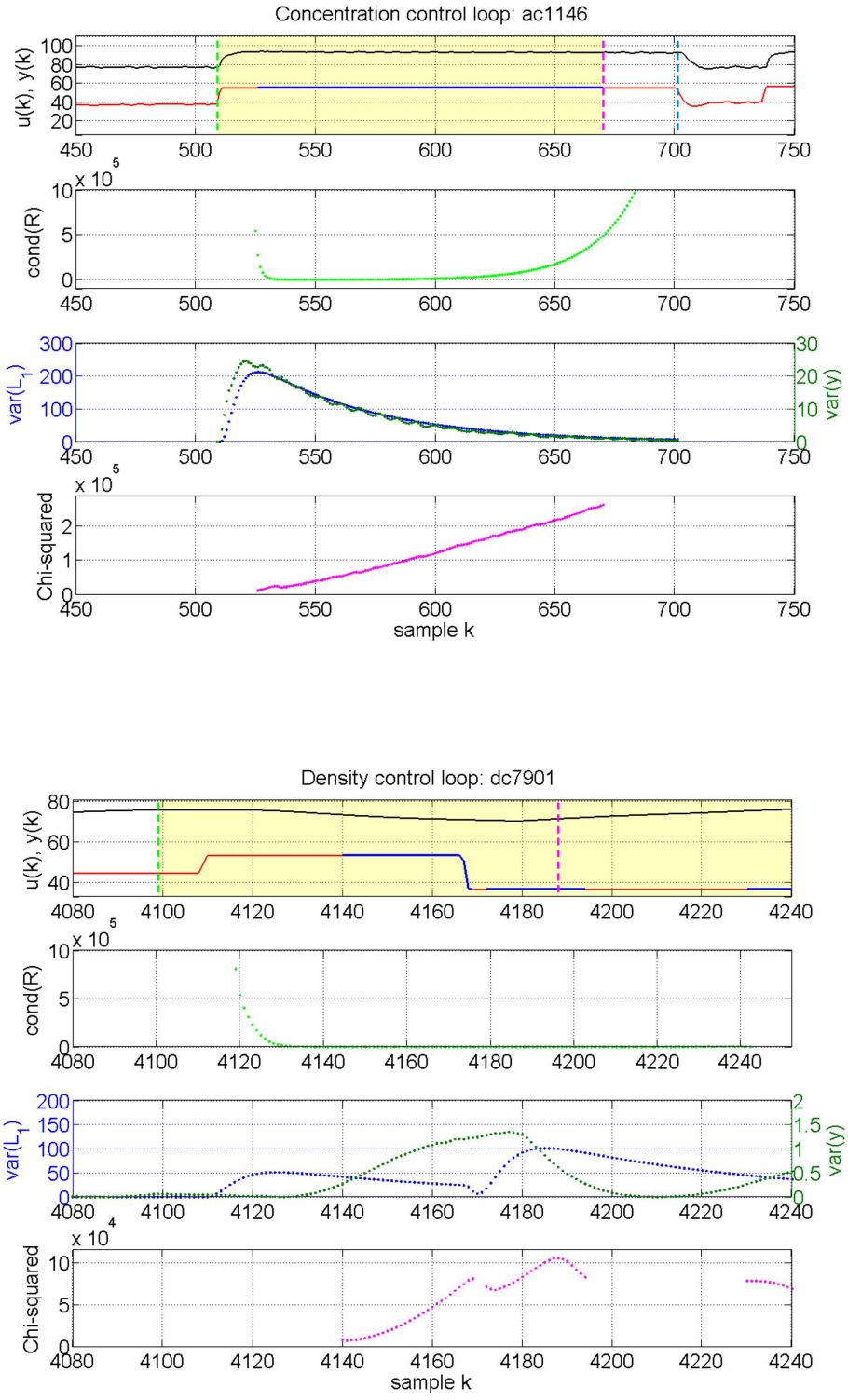


Figure 5.14. Plots of the scanning results with Method 3 for the process examples concentration and density. A light blue dashed line marks in some plots the beginning and/or end of the manual mode. A dark blue line in the first subplot of each example marks those samples, where all criteria were fulfilled.

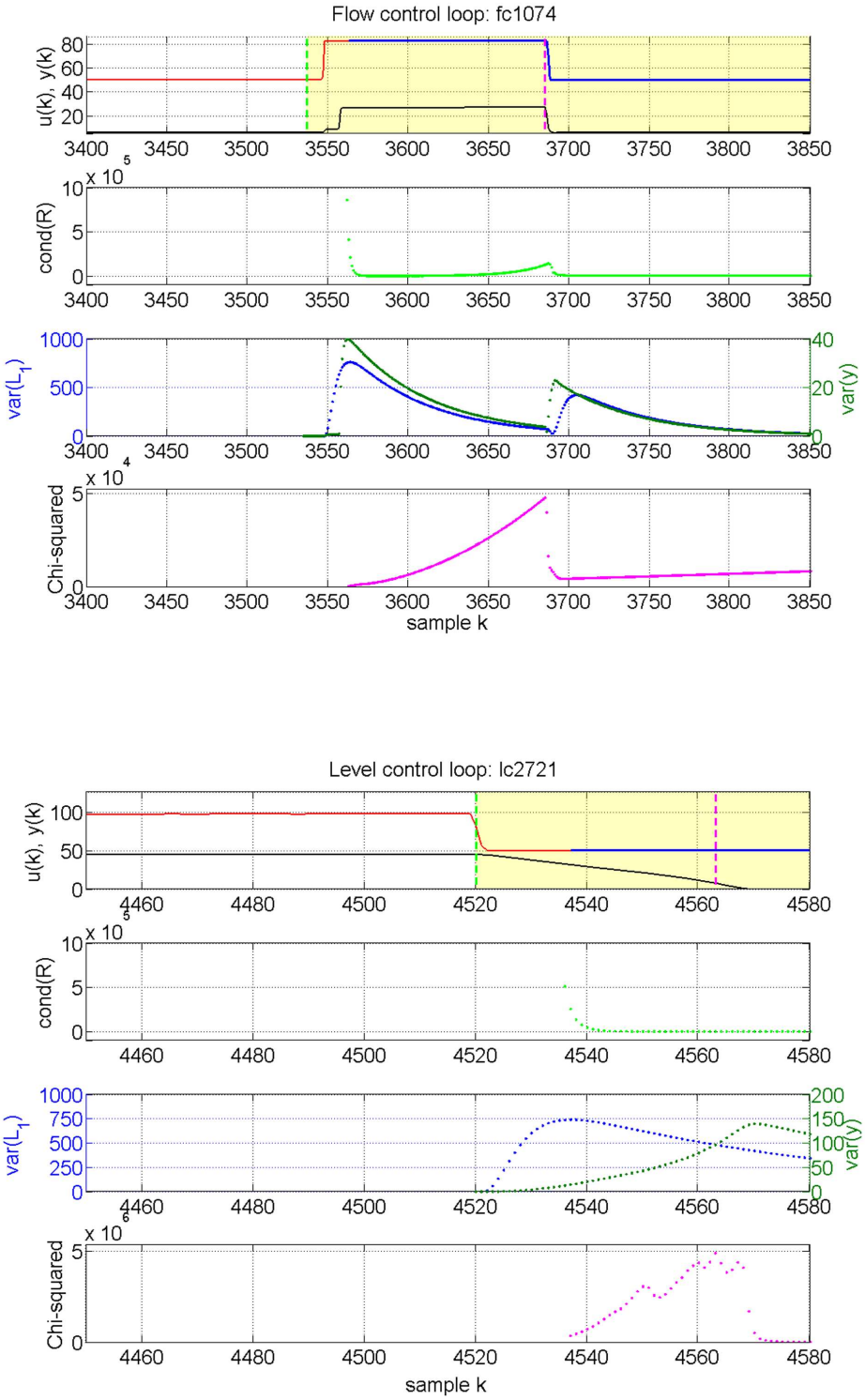


Figure 5.15. Plots of the scanning results with Method 3 for the process examples flow and level. A light blue dashed line marks in some plots the beginning and/or end of the manual mode. A dark blue line in the first subplot of each example marks those samples, where all criteria were fulfilled.

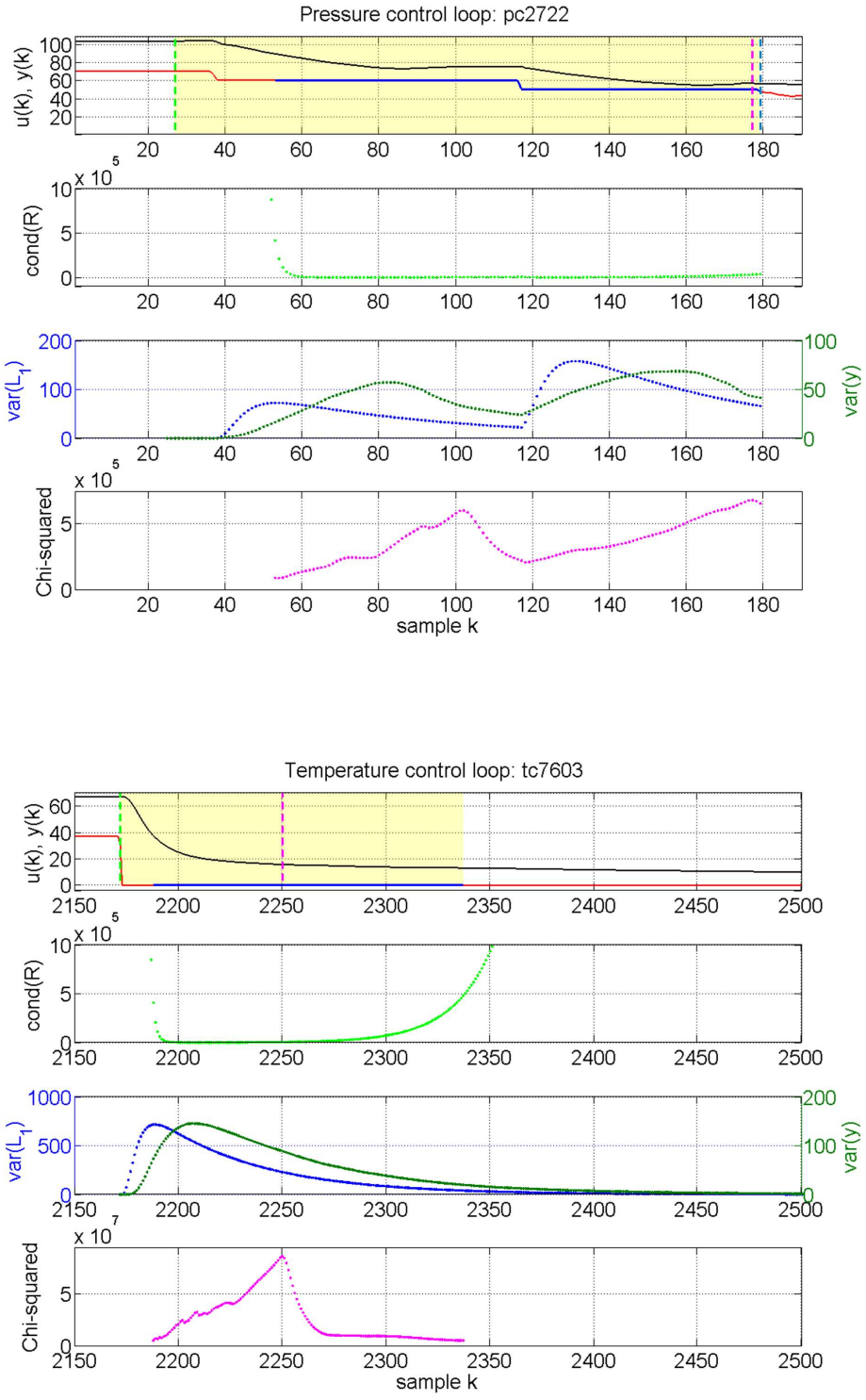


Figure 5.16. Plots of the scanning results with Method 3 for the process examples pressure and temperature. A light blue dashed line marks in some plots the beginning and/or end of the manual mode. A dark blue line in the first subplot of each example marks those samples, where all criteria were fulfilled.

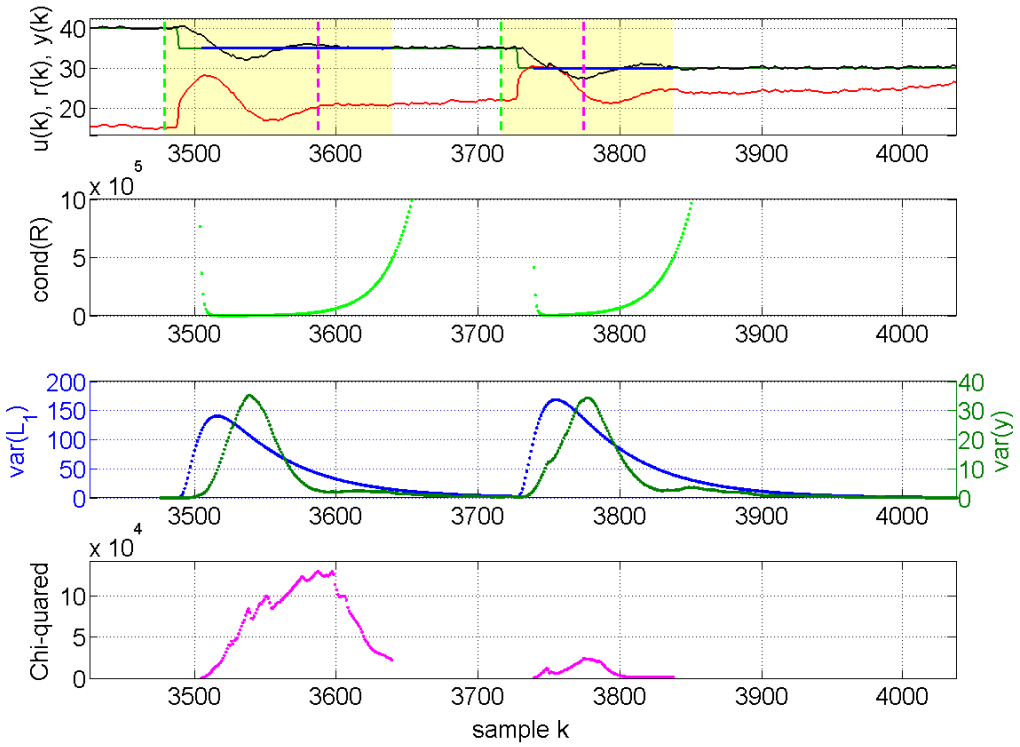


Figure 5.17. Plots of the scanning result with Method 3 for a real process (density) in automatic mode.

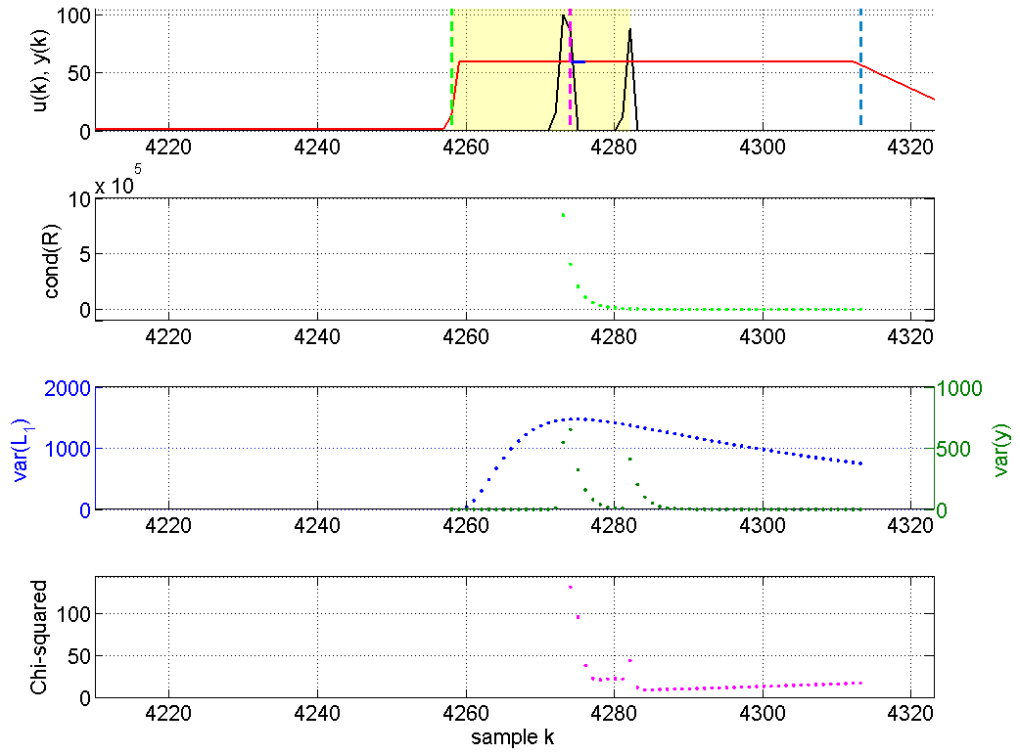


Figure 5.18. Plots of the a bad scanning result with Method 3 of a real process (density) in manual mode.

Finally, the following advantages (+) and disadvantages (-) can be mentioned:

- + theoretically based
- + safe detection of excitation
- + avoids intervals with big disturbances, because it takes $y(k)$ into account
- + can handle an integrator in the process
- + avoids $u(k)$ with too fast changes
- + less problems handling big time delays due to first Laguerre filter
- + time delay of the process does not have to be known
- + low noise sensitivity
- - high computational complexity
- - still finds useless intervals

5.5 Comparison Methods 1-3

A quantitative comparison of Methods 1-3, which method finds more useful intervals, would be difficult, since Method 3 searches additionally in automatic mode and all three methods detect also useless intervals. Hence, it is not easy to quantify this. Nevertheless, a comparison of all methods (see Table 5.4) and their presented examples show that Method 3 has qualitatively the best performance. Method 3 is based on the main ideas of Method 1 and 2, but the realization is slightly different.

Method 1 was too conservative as it demanded steady-state before and after a step and found therefore fewer useful intervals. But thereby big disturbances were avoided. The use of the condition number and the variances of the process input/output signals makes Method 2 successful in detection of excitation. However, the weak points of Method 2 are, that the behaviour of $y(k)$ is not verified in relation to the process input $u(k)$ and that big time delays are problematical.

Consequently, Method 3 tries to use the strengths of Methods 1-2 and to avoid their weaknesses. Excitation is detected as well by the analysis of the condition number and the variances, but instead of the FIR model in Method 2, a Laguerre model is used. Through the Laguerre model high frequency disturbances are filtered out and a parameter estimation is possible without knowing the time delay in advance. This parameter estimation leads to another essential part of Method 3, the chi-square test. Method 1 tries to avoid big disturbances by requiring steady-states while Method 3 does so by checking the correlation between $y(k)$ and $u(k)$ by the chi-square test. Method 3 has of course the most complex algorithm and therefore takes longer to run. Compared to Method 2 it needs circa 45% more

time but this is acceptable since much more computations are necessary like calculating the Laguerre filter outputs, estimating the Laguerre model parameters and performing the chi-square test.

The processes which should be found by Method 3 are restricted to be simple linear models and SISO. Important is that related process is approximately first order (dynamics could be neglected due to the dominant time constant). Furthermore, the developed method should also work if APC-Tools (Advanced process control) are used as controller. Moreover, Method 3 should still detect useful intervals if white noise is present in the data as long as a appropriate SNR is given.

Even if Method 3 is the most computationally expensive it delivers the best result. For this reason Method 3 is chosen to be the final method which is used for scanning the database. In Chapter 6 it is validated with a small case study.

5.6 Summary

Chapter 5 presents three developed methods for scanning the database for useful intervals for process identification. The three methods are explained in the chronological sequence they were developed during the project. This is the reason why Method 3 contains ideas and elements of the other methods and thus has the best performance of all.

Method 1 orientates more towards human scan behaviour with features as finding steps and requiring steady-state before and after the step. Therefore it is simpler and faster and avoids intervals with big disturbances but finds thereby only less useful intervals since it can not handle integrators. Moreover, there is no additional check if the signals could deliver a reasonable process model or not.

Method 2 uses theoretical knowledge for detection of excitation by calculating the condition number of the information matrix of a 4th order FIR model and checking the variance of the process input $u(k)$ and output $y(k)$. This works successfully but has the disadvantage to accept too fast changes of $u(k)$ which are not useful and furthermore does not verify how much $y(k)$ is correlated with $u(k)$.

The final Method 3 was developed with the gained experience of the previous methods. It is based on Laguerre series expansions and has two essential features which makes it more successful compared to Methods 1 and 2:

1. Scan for excitation by analyzing the variances of the first Laguerre filter output and of $y(k)$ and by additionally checking the condition number of the information matrix $\boldsymbol{R}(k)$.

2. Estimation of a Laguerre model and a statistical check of its parameters significance by a chi-square test. This gives an indication of the accuracy and reliability of an estimated process model.

This method has a higher complexity but it is implemented efficiently. Furthermore it can handle processes with integrator and both manual and automatic modes. It is noise robust and big disturbances can be indicated by a low chi-square test result. The performance was already shown in Section 5.4.2 and will be validated by a case study in Chapter 6.

Rules to determine the parameter values and thresholds could only be offered in parts. For future work this would require more attention. Moreover, it has to be mentioned that intervals which are useless for process identification are still found. In such cases however, the chi-square test is much lower in comparison to cases where good excitation is presented and therefore bad intervals can be discarded.

It is important to keep in mind that the aim is to find useful intervals and not to deliver a process model. This is the user's final task, to decide which of the found intervals are suitable for identification. For this reason some other interesting points in terms of process identification were not considered, like the handling of scattered parts of useful data. In [12] is for example explained how to use several useful data intervals for system identification.

Method 1	Method 2	Method 3
• + low computational complexity • + avoids intervals with large disturbances • - not based on solid theoretical analysis • - too conservative to require steady state • - no check if the signals deliver a reasonable process model • - can not handle an integrator in the process • - difficult choice of the parameters because of no guidelines	• + theoretical foundation • + safe detection of excitation • + can handle integrators • - does not take into account the correlation between $u(k)$ and $y(k)$ • - has problems with time delays • - may accept $u(k)$ with too fast changes (high frequencies) • - difficult choice of the parameters because of no guidelines	• + theoretically based • + safe detection of excitation • + avoids intervals with big disturbances, because it takes $y(k)$ into account • + can handle an integrator in the process • + avoids $u(k)$ with too fast changes • + less problems handling big time delays due to first Laguerre filter • + time delay of the process does not have to be known • + low noise sensitivity • - high computational complexity • - still finds useless intervals

Table 5.4: Overview of the advantages (+) and disadvantages (-) of Method 1-3

Chapter 6

Case study

In the previous chapter different types of scanning algorithm were proposed, compared and Method 3, which had the best performance, was chosen. A first assessment of its performance was already done in Chapter 5.4.3. However, only some simulated and a few real process data were considered. Before this method can be applied in a daily routine, it is important to get more information about its quality and characteristics. Therefore, a case study is presented in Section 6.1 which includes a scan of the entire database and an analysis of the results. Furthermore, Method 3 is validated with these scanning results in Section 6.2 and some advices for the final user of the developed method are given in Section 6.3. At last the presentation of Method 3 at Perstorp is shortly reported.

6.1 Results of an entire scan of the database

The entire database, presented in Chapter 3, is taken for the following case study. This database originates from a chemical plant of the Swedish company Perstorp which produces mainly intermediate goods. The database consists of measurements of only one plant that processes fine chemicals and operates mixed (continuous and batch). Moreover, the process data was continuously logged from January 2007 till January 2010 (circa 3 years) with a sample rate of 15s (total number of samples $= 1.089 \times 10^9$). During this time no product change took place. Furthermore, seven different types of processes have to be taken into account. The types are: density, flow, concentration, level, conductivity, temperature and pressure. Most of the observed processes are approximately first order or lower and could have time delays up to 10 min. All control loops are PID, the total number is 211 and no MPC is applied. The control loops were operating in manual as well as in automatic mode, however the main mode was automatic. The measurements include disturbances, which are basically noise with approximately zero mean and small amplitude (a high SNR is given) but can also be deterministic due to several couplings existing between the loops.

The above described database was scanned using Method 3 with the same set-

tings as in Chapter 5.4 (see Table 5.3). Some interesting statistics of the scan are presented in Table 6.1. A scan with a usual desktop PC takes circa 1.5 h which is acceptable in comparison to the amount of samples scanned (6.7 GigaByte). Furthermore, from 211 control loops 190 loops were found (90.1%) containing a minimum of one "useful" data stretch. However, at this point it can not be stated how much of the found stretches are really useful for process identification. Considering both modes separately, it is noticeable that in manual mode almost 20% more data stretches were found than in automatic mode.

Additionally, Table 6.1 delivers information about the average of found intervals for each control loop separated by the loop type. For the flow control loops the algorithm found roughly 5 times more intervals compared to the remaining control

Scanning results	
1.52 h	Scanning time (9.31×10^8 automatic mode samples and 1.57×10^8 manual mode samples)
143 (67.7%) 185 (87.7%) 190 (90.1%)	Total number of control loops where some useful data was found: - automatic mode - manual mode - both modes together
239 660 84 130 0 35.3 100	Mean number of intervals found: - Density - Flow - Concentration - Level - Conductivity - Temperature - Pressure
102.8 125.3 114.1	Mean duration of found intervals [*samples*]: - automatic mode - manual mode - both modes

Table 6.1. Statistical values of the scanning results of the complete database consisting of three years of data with 211 control loops.

loops. This fact is illustrated in Figure 6.1. Moreover, it is identifiable that for the other control loop types the average of intervals which were found differ from 33 (temperature control loops) till 209 (density control loops) with the exception of conductivity control loops for which nothing was found. The reason for the latter is on the one hand that only 2 control loops of this type exist and on the other hand one of them has a lot of high frequency disturbances included and the other one seemingly has hardly any excitation in the input signal. For the control group at Perstorp all densities loops, most of the temperature loops and some level and pressure loops are important. Flow loops are as well important but rarely cause a problem for process identification. Therefore, the bad hit rate for the conductivity control loops is acceptable.

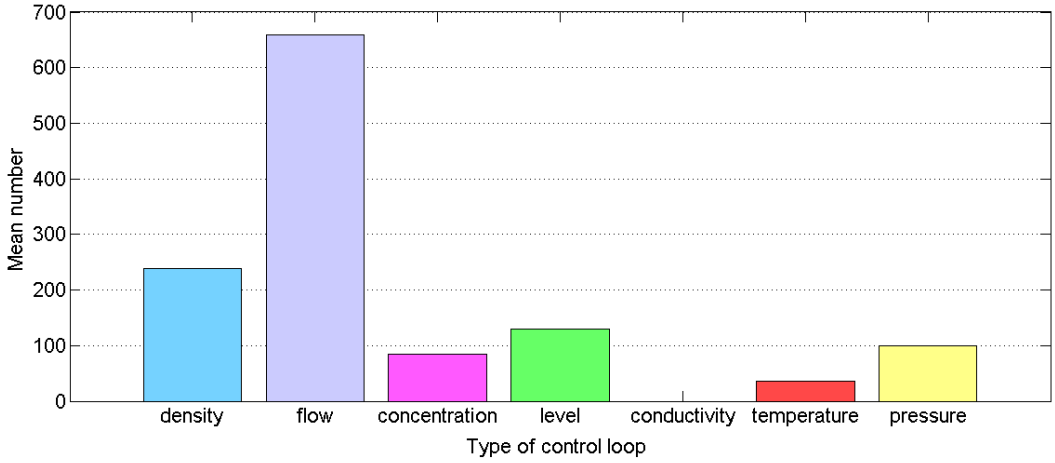


Figure 6.1. Mean number of intervals found for each control loop type.

Furthermore, related to the found intervals, the question is how the single control loops are distributed. For that purpose Figure 6.2 shows the absolute number of intervals found for each control loop. Each type of control loop has a different distribution. The reason for this is probably that some types of control loops change more often than others, due to operational reasons. Thus, there exist some control loops for which the method only found a few or even no useful intervals. Less than 10 intervals were found for 73 control loops (34.6% of all).

The mean time of the intervals found is also given by Table 6.1. All in all the intervals found are in average circa 120 samples ($\approx$ 30 min) long. That is, in general, sufficient for process identification. Since the found intervals should only include the informative part of the data (input excitation and process response), a reflection of this is expected in the mean duration of the different control loop types related to their typical time constants. However, this can not be confirmed by the Figure 6.3 which shows the mean time of the intervals found for the different control loop types. For example, flow control loops have usually small time constants but the scanning results show a mean duration longer than most of the other control loop types, especially in automatic mode. Despite this, the mean duration of most of the control loops is shorter in automatic than in manual mode.

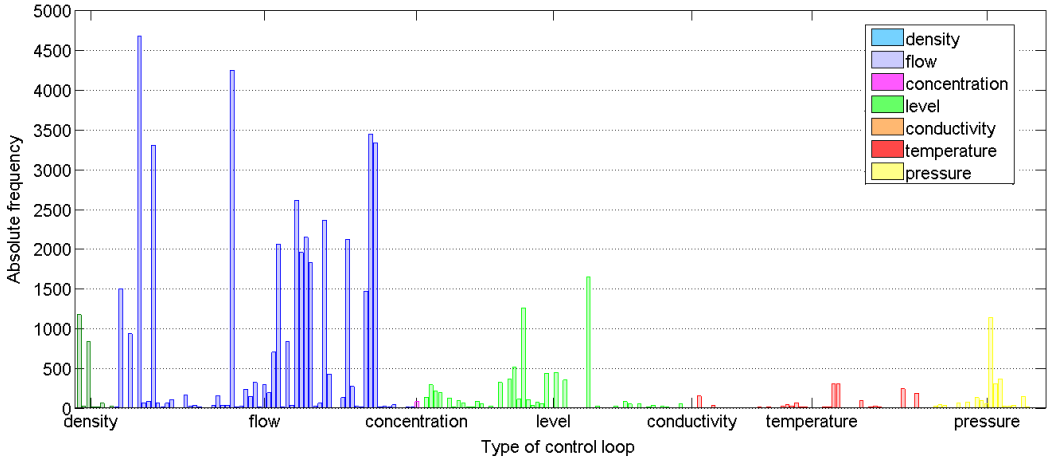


Figure 6.2. Absolute number of found intervals for each control loop.

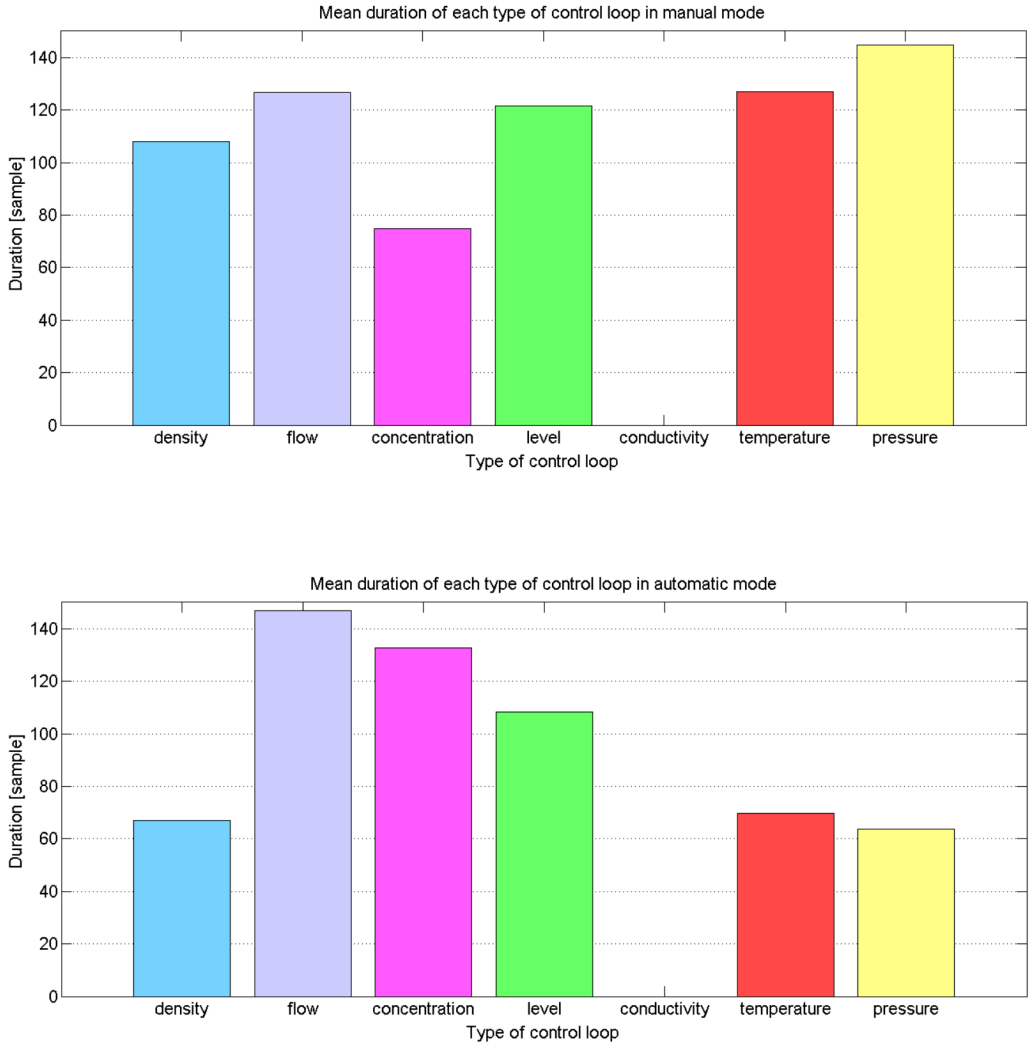


Figure 6.3. Mean duration of found intervals of each control loop type in manual (top) and automatic (bottom) mode.

6.2 Validation of Method 3

This chapter deals with the validation of Method 3. The aim is to illustrate its reliability and accuracy for finding useful data in real measured process data. It might be difficult to find a method to assess the performance. One possibility would be to estimate process models using intervals that were found and compare them with the related "true" models. However, "true" models hardly exist. An analysis by checking the variances of the estimated parameters is already done by the chi-square test. Advantageous would be a training set, where all samples are labeled as "useful" or "not useful". Then the scanning results only have to be compared with the labels. This validation would be really exact but to build such a training set costs a lot of time and currently no solution exists on how to label the data error-free.

For validation the bump tests of Perstorp's control group (see Chapter 1.2) are considered. This data is related to performed identification experiments (known to be useful for process identification and therefore suitable as a performance criterion). The scanning results are checked in terms of how many of those bump tests are detected by the algorithm or not. With the same parameters and thresholds as before all 21 bump tests are found by Method 3.

Additional to the bump tests, Method 3 finds other stretches of data which might lead to a useful identification. Estimated process models of these stretches have parameter values close to those, estimated from the bump tests.

To see how robust the classification of Method 3 is against variations of its design parameters, one after another was changed. After each variation, a complete scan of the entire database was performed and the results were finally analysed. On the one hand, it is checked the total amount of control loops, for which something useful was found, and on the other hand how many of the bump tests are still detected. The results are given in Table 6.2 and show that small deviations of the parameters have only little influence on the performance of the algorithm. The total amount of control loops which were found changes minimally and most of the bump tests are still detected, except for variations of λ_m when it differed up to 8 from the default value.

6.3 Advice for application of Method 3

When using Method 3 there a few relevant factors to consider. One should take into account that the scanning time is not negligible. However, a scan of the entire data of one day of all 211 control loops takes currently approximately 4.9 s which is in most cases fast enough, because only a scan of a few days is required or even only of the measured data of one day (according to Perstorp's control group).

Furthermore, the user should sort the scanning results for one control loop by the chi-square test results, which ranks the several stretches for process identifica-

Parameter variation (deviation)	Total amount of control loops, for which sth. useful was detected	Number of Bump tests detected (max. 21)
With default values	190	21
Real pole:		
$\alpha = 0.7$ ($-$ 0.1)	190	20
$\alpha = 0.9$ ($+$ 0.1)	190	20
Forgetting factor:		
$\lambda_R = 0.92$ ($-$ 0.03)	190	21
$\lambda_R = 0.98$ ($+$ 0.03)	190	21
Mean estimation		
$\lambda_m = 0.90$ ($-$ 0.09)	187	13
$\lambda_m = 0.95$ ($-$ 0.04)	186	19
Variance estimation		
$\lambda_v = 0.85$ ($-$ 0.05)	190	21
$\lambda_v = 0.95$ ($+$ 0.05)	189	18

Table 6.2. Robustness analysis of Method 3 against variations of its design parameter. After each change a scan of the entire database was performed.

tion. Therefore, it is advisable that one should start to estimate a process model with the stretch which has the highest χ^2. Moreover, it is recommended that several models with different intervals found are estimated and compared with each other to find a reliable model (see Chapter 2.6 for information about model validation).

Last but not least, the user must keep in mind that the current parameter and threshold settings are adapted to the given dataset and have to be rechecked for other databases.

6.4 Presentation of Method 3 at Perstorp

The developed scanning algorithm was presented at Perstorp in the middle of May 2010. After discussing the several steps that Method 3 performs, a lot of time was spent going through a large number of randomly chosen examples of intervals and control loops. Krister Forsman, Corporate Specialist Process Control at Perstorp, stated: “We were very impressed by how good the method works! To be honest I think it’s very useful to us already as it is, let alone how good it will be after some

more tweaking with the parameters. ... I'm sure we will use these methods in the future, on other plants as well."

For the control group of Perstorp the project was therefore successful and the developed scanning tool will make it now much more easier for them to identify suitable process models of their chemical plant.

6.5 Summary

In this chapter a case study was presented and furthermore Method 3 is validated with the results of this case study. The characteristic features of the scanning result can be summarized with the following items:

- A scanning time of 1.5 h was necessary for the entire database (this equals to circa 4.9 s for one day).
- At least one data stretch was found for 190 of 211 control loops.
- The number of data stretches found for one control loop varies a lot. On the one hand it depends on the loop type and on the other hand on its general activity (input signal changes).
- The mean duration of the intervals found is circa 30 min. A differentiated consideration of the control loop types shows no obvious correlation between the typical process time constant and the mean duration.

For validation the amount of bump tests found was checked. All bump tests were found by Method 3. Additionally, Method 3 finds other useful intervals in closed/open loop operation. Moreover, the robustness of the algorithm against small deviations of its parameters was analysed. It can be asserted that those small deviations have no big influence. However, the current settings are mainly fitted to the given database and do not constitute a best general choice. Perstorp, the final user of Method 3, is satisfied with the first results and considers this method a useful tool to be applied on other plants as well.

Chapter 7

Summary and conclusions

Models in the process industry are important for several reasons, such as process development, optimization tasks, control design and performance monitoring. Traditional methods to obtain a model are modeling from physical principles and performing identification experiments. Process plants are however complex systems and experiments are often time-consuming because of large time constants or even impossible due to operational reasons. Nevertheless, the signals of each control loop nowadays are mostly logged and it is possible to form a huge database of the plant. The Swedish chemical company Perstorp provided a database taken over three years of data of 211 control loops with a sample rate of 15 s.

The aim of this work was the development of a method which scans automatically the database for intervals which are informative enough for process identification. The method should require a minimum of knowledge about the process and the algorithm must be fast and therefore simple because of the huge amount of data. Furthermore, the processes to be identified are restricted to be linear low-order models and SISO. The development was challenged by several factors like: different types of processes, interconnections, disturbances, time delays and the fact that the loops are mainly in steady-state. No suitable method is known from the literature.

Three different methods were developed from which the last one has the most complex algorithm but also the best performance of all. The so-called Method 3 requires no further knowledge of the process, except of its type (density, flow, concentration, level, conductivity, temperature and pressure). It has two essential features which make this method successful:

1. Scanning for excitation by analyzing the variance and condition number of the first Laguerre filter and of the process output.

2. In case of excitation, a Laguerre model is estimated and a statistical check of its parameters significance by a chi-square test is performed.

Finally, a case study with data from a mixed (continuous and batch) operated chemical plant of Perstorp is presented and the proposed Method 3 is validated. In the case study a scan of the entire database (data originate from one single plant) is executed which delivered a fast average scanning time of 4.9 s for one day. From 211 control loops 190 loops are found (90.1%) containing a minimum of one "useful" data stretch or more. Furthermore, for most of the control loops, several intervals are found with a sufficient length for process identification. It also finds all intervals in the data in which identification experiments were performed and thus are known to be useful. Moreover, Method 3 finds additionally other useful intervals in closed/open loop operation. The choice of its parameters and threshold values has certainly a big influence on the performance but the scanning results are quite insensitive to small deviations of the current parameter values. However, it should be mentioned that Method 3 still finds useless intervals, for which reason a manual screening of the scanning results is still necessary before estimating process models. All in all, the proposed method gives direction to a new possibility for process identification besides the classical ones. The feasibility of such a scanning method is demonstrated in this work. In addition, Method 3 was presented to the control group of Perstorp and the algorithm was handed over. They were "very impressed by how good the method works" and evaluate it to be "very useful". Furthermore, the developed scanning algorithm shall be patented.

Future work

Although the developed method delivers useful results there is still a lot of research to be done and it is more than likely that the accuracy of the scanning method can be improved. The choice of the parameter and threshold settings could be analysed in more detail and objectively since in this work some values were chosen only by experience. Furthermore, the algorithm can be speeded up for example by recursively updating the covariance matrix $\boldsymbol{P}(k)$ or by other simplifications of the algorithm. Besides that, some other aspects were less discussed and not implemented, like the use of data when the process input $u(k)$ goes into saturation or the reconstruction of missing samples. At the moment, the entire interval is skipped during the scanning procedure if data is missing (e.g. because of measurements errors).

Another important aspect which has to be stressed is that the proposed method is only developed with the data of one industrial plant. It would be interesting to see how the scanning method works with data from other plants. Thereby, a better validation of the developed method would be possible and the parameters and thresholds could be adapted in terms of a higher generalization of the method.

Furthermore, the developed method might open different, new possibilities for other purposes. One idea is for example that a few or even no found intervals of one control loop indicate that the related controller is hardly tunable since the behaviour of the loop seems not to be explainable by the estimated Laguerre model.

Appendix A

Chi-square table

d.f.	$\chi^2_{.25}$	$\chi^2_{.10}$	$\chi^2_{.05}$	$\chi^2_{.025}$	$\chi^2_{.010}$	$\chi^2_{.005}$	$\chi^2_{.001}$
1	1.32	2.71	3.84	5.02	6.63	7.88	10.8
2	2.77	4.61	5.99	7.38	9.21	10.6	13.8
3	4.11	6.25	7.81	9.35	11.3	12.8	16.3
4	5.39	7.78	9.49	11.1	13.3	14.9	18.5
5	6.63	9.24	11.1	12.8	15.1	16.7	20.5
6	7.84	10.6	12.6	14.4	16.8	18.5	22.5
7	9.04	12.0	14.1	16.0	18.5	20.3	24.3
8	10.2	13.4	15.5	17.5	20.1	22.0	26.1
9	11.4	14.7	16.9	19.0	21.7	23.6	27.9
10	12.5	16.0	18.3	20.5	23.2	25.2	29.6
11	13.7	17.3	19.7	21.9	24.7	26.8	31.3
12	14.8	18.5	21.0	23.3	26.2	28.3	32.9
13	16.0	19.8	22.4	24.7	27.7	29.8	34.5
14	17.1	21.1	23.7	26.1	29.1	31.3	36.1
15	18.2	22.3	25.0	27.5	30.6	32.8	37.7
16	19.4	23.5	26.3	28.8	32.0	34.3	39.3
17	20.5	24.8	27.6	30.2	33.4	35.7	40.8
18	21.6	26.0	28.9	31.5	34.8	37.2	42.3
19	22.7	27.2	30.1	32.9	36.2	38.6	32.8
20	23.8	28.4	31.4	34.2	37.6	40.0	45.3
21	24.9	29.6	32.7	35.5	38.9	41.4	46.8
22	26.0	30.8	33.9	36.8	40.3	42.8	48.3
23	27.1	32.0	35.2	38.1	41.6	44.2	49.7
24	28.2	33.2	36.4	39.4	32.0	45.6	51.2
25	29.3	34.4	37.7	40.6	44.3	46.9	52.6
26	30.4	35.6	38.9	41.9	45.6	48.3	54.1
27	31.5	36.7	40.1	43.2	47.0	49.6	55.5
28	32.6	37.9	41.3	44.5	48.3	51.0	56.9
29	33.7	39.1	42.6	45.7	49.6	52.3	58.3
30	34.8	40.3	43.8	47.0	50.9	53.7	59.7
40	45.6	51.8	55.8	59.3	63.7	66.8	73.4
50	56.3	63.2	67.5	71.4	76.2	79.5	86.7
60	67.0	74.4	79.1	83.3	88.4	92.0	99.6
70	77.6	85.5	90.5	95.0	100	104	112
80	88.1	96.6	102	107	112	116	125
90	98.6	108	113	118	124	128	137
100	109	118	124	130	136	140	149

Table A.1. Chi-square table taken from [27].

Appendix B

Maximal time delay modelled by Laguerre model

In [9] is described how to estimate the dead-time T_d of a process by estimating a Laguerre model. As described in Chapter 2.7, a Laguerre model of order n consists of one low-pass filter and $(n-1)$ cascaded time delays with the discrete-time zeros z_i. After the Laguerre model is estimated, the discrete-time zeros z_i are converted to approximate continuous-time zeros

$$s_i = \frac{1}{T_s}\ log(z_i) = -\ \frac{1}{T_s}\ log(\alpha) \tag{B.1}$$

with $z_i = \frac{1}{\alpha}$ and the sampling time T_s.

In the following, the Padé approximation (see Equation (B.2), [18]) of a continuous-time dead-time T_d is used by comparing the estimated zeros s_i to it.

$$e^{-s\ T_d} \approx \left(\frac{1\ -\ s\ \frac{T_d}{2n}}{1\ +\ s\ \frac{T_d}{2n}} \right)^n \tag{B.2}$$

$$\frac{T_d}{2} = \sum_{i=1}^{n} \frac{1}{s_i}\ =\ \frac{n}{s_i} \tag{B.3}$$

Concerning the Laguerre model all s_i are equal whereby Equation (B.3) can be formulated. Then, the maximal time delay $T_{d,max}$ which can be modelled by a specified Laguerre model is

$$T_{d,max} = 2 \cdot (n-1) \cdot \frac{1}{s_i} = -2 \cdot (n-1) \cdot T_s \cdot \frac{1}{log(\alpha)} \tag{B.4}$$

since $n-1$ time delays are cascaded.

The other way around, the necessary number n of Laguerre filters can be determined if Equation (B.4) is written in terms of n:

$$n = \frac{T_{d,max} \cdot log(\alpha)}{2\ T_s} + 1 \tag{B.5}$$

Example B.1 demonstrates how the necessary number of Laguerre filters is calculated with the parameter values which are used in this work.

Example B.1: Determination of necessary number Lag. filters

Assuming that we have a real pole $\alpha = 0.8$, a sample time of $T_s = 15\ s$ and expect a maximum time delay of $T_{d,max} = 10\ min = 600\ s$, then a necessary number of $n = 6$ can be determined with Equation (B.5):

$$n = \frac{600 \cdot log(0.8)}{2 \cdot 15} + 1 = 5.5 \approx 6$$

Appendix C

Proof of Theorem 5.1

Proof The condition number of a given $(m \times n)$ matrix $\boldsymbol{A}$ can be calculated by using the *Singular Value Decomposition* (SVD). The SVD factorizes the matrix $\boldsymbol{A}$ into three special matrixes, whose product result is $\boldsymbol{A}$:

$$\boldsymbol{A} = \boldsymbol{U} \cdot \boldsymbol{S} \cdot \boldsymbol{V}^T \tag{C.1}$$

The matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal and $\boldsymbol{S}$ is a $(m \times n)$ diagonal matrix with the structure:

$$\boldsymbol{S} = \left(\begin{array}{ccc|ccc} \sigma_1 & \cdots & 0 & & \vdots & \\ \vdots & \ddots & \vdots & \cdots & 0 & \cdots \\ 0 & \ldots & \sigma_r & & \vdots & \\ \hline & \vdots & & & \vdots & \\ \cdots & 0 & \cdots & \cdots & 0 & \cdots \\ & \vdots & & & \vdots & \end{array}\right) \tag{C.2}$$

where the entries $\sigma_1 \ldots \sigma_r$ are nonnegative real number which are called the *singular values* with $\{r \geq m \wedge r \geq n\}$.

The condition number of the matrix $\boldsymbol{A}$ is then defined by the following equation

$$cond(\boldsymbol{A}) = \frac{\sigma_{max}}{\sigma_{min}} \tag{C.3}$$

where σ_{max} and σ_{min} denote the maximal and minimal singular values of $\boldsymbol{S}$, respectively.

Related to the information matrix $\boldsymbol{R}(k)$ we assume that $\boldsymbol{A}$ is a quadratic $n \times n$ matrix. The inverse $\boldsymbol{A}^{-1}$ can be formulated by using the matrixes $\boldsymbol{U}$, $\boldsymbol{S}$ and $\boldsymbol{V}$ in the way

$$\boldsymbol{A}^{-1} = (\boldsymbol{V}^T)^{-1} \cdot \boldsymbol{S}^{-1} \cdot \boldsymbol{U}^{-1} \tag{C.4}$$

Since $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal and $\boldsymbol{S}$ diagonal, the Equation (C.4) can be reformulated to

$$\boldsymbol{A}^{-1} = \boldsymbol{V} \cdot \boldsymbol{S}^{-1} \cdot \boldsymbol{U}^{T} \tag{C.5}$$

and $\boldsymbol{S}^{-1}$ is given with the entries

$$\boldsymbol{S}^{-1} = \begin{pmatrix} \sigma_1^{-1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n^{-1} \end{pmatrix} \tag{C.6}$$

Therefore, the condition number of the $\boldsymbol{A}^{-1}$ is defined by

$$cond(\boldsymbol{A}^{-1}) = \frac{\sigma_{min}^{-1}}{\sigma_{max}^{-1}} = \frac{\sigma_{max}}{\sigma_{min}} \tag{C.7}$$

which is the same as calculating the condition of $\boldsymbol{A}$ in Equation (C.3) and thus

$$cond(\boldsymbol{A}) = cond(\boldsymbol{A}^{-1}) \tag{C.8}$$

□

Bibliography

[1] Balázs Abonyi, János; Feil. *Cluster Analysis for Data Mining and System Identification*. Birkhäuser Verlag AG, 2007.

[2] P. Carrette, G. Bastin, Y.Y. Genin, and M. Gevers. Discarding data may help in system identification. *IEEE Transactions on Signal Processing*, 44(9):2300–2310, 1996.

[3] D. Clarke. Self-tuning multistep optimisation controllers. *Adaptive Control Strategies for Industrial Use*, pages 1–28.

[4] G. Cybenko. Just-in-time learning and estimation. *NATO ASI Series*, pages 423–434, 1996.

[5] A. Elnaggar, G.A. Dumont, and A.L. Elshafei. Delay estimation using variable regression. In *Proceedings of American Control Conference*.

[6] M. Green and J.B. Moore. Persistence of excitation in linear systems. *Systems & Control Letters*, 7(5):351–360, 1986.

[7] A. Horch. *Condition Monitoring of Control Loops*. PhD thesis, KTH, Signals, Sensors and Systems, 2000.

[8] A.J. Isaksson, A. Horch, and G.A. Dumont. Event-triggered deadtime estimation–comparison of methods. *Control Systems 2000*, pages 209–215, 2000.

[9] M. Isaksson. A comparison of some approaches to time-delay estimation. Master's thesis, Lund Institute of Technology, 1997.

[10] R. A. Johnson, I. Miller, and J. E. Freund. *Miller and Freund's probability and statistics for engineers*. Pearson Prentice Hall, 2005.

[11] MSDN Library. Validating data mining models (analysis services - data mining), June 2010. http://msdn.microsoft.com/en-us/library/.

[12] L. Ljung. *System identification: Theory for the user*. Prentice-Hall Englewood Cliffs, NJ, 2nd edition edition, 1998.

[13] L. Ljung. Perspectives on system identification. In *Proceedings of 17th IFAC World Congress*, pages 7172–7184, 2008.

[14] L. Ljung and T. Glad. *Modeling of dynamic systems*. PTR Prentice Hall Englewood Cliffs, NJ, 2002.

[15] H. Lutz and W. Wendt. *Taschenbuch der Regelungstechnik: Mit Matlab und Simulink*. Harri Deutsch Verlag, 2007.

[16] C.B. Lynch and G.A. Dumont. Control loop performance monitoring. *IEEE transactions on control systems technology*, 4(2):185–192, 1996.

[17] P.M. Mäkila. Approximation of stable systems by Laguerre filters. *Automatica*, 26(2):333–345, 1990.

[18] H. Padé. *Sur la représentation approchée d'une fonction par des fractions rationelles*. PhD thesis, Annales de l'Ecole Normale Sup., Paris, 1892.

[19] H.I. Park, S.W. Sung, I.B. Lee, and J. Lee. On-line process identification using the Laguerre series for automatic tuning of the proportional-integral-derivative controller. *Ind. Eng. Chem. Res*, 36(1):101–111, 1997.

[20] R. Silva, D. Sbarbaro, and B.A.L. de la Barra. Closed-loop process identification under PI control: A time domain approach. *Ind. Eng. Chem. Res*, 45(13):4671–4678, 2006.

[21] C. Songling and R.R. Rhinehart. An efficient method for on-line identification of steady state. *Journal of Process Control*, 5(6):363–374, 1995.

[22] S.W. Sung and I.B. Lee. On-line process identification and PID controller autotuning. *Korean Journal of Chemical Engineering*, 16(1):45–55, 1999.

[23] B. Wahlberg. System identification using laguerre models. *IEEE Trans. on Automatic Control*, January 1991.

[24] B. Wahlberg and E.J. Hannan. Parametric signal modelling using Laguerre filters. *The Annals of Applied Probability*, 3(2):467–496, 1993.

[25] L. Wang and WR Cluett. Optimal choice of time-scaling factor for linear system approximations using Laguerre models. *IEEE Transactions on Automatic Control*, 39(7):1463–1467, 1994.

[26] L. Wang and W.R. Cluett. Building transfer function models from noisy step response data using the Laguerre network. *Chemical Engineering Science*, 50(1):149–161, 1995.

[27] R.J. Wonnacott and T.H. Wonnacott. *Statistics: Discovering its power*. John Wiley & Sons, 1982.