LIST OF SYMBOLS

- a: Zero mean white noise signal
- $C(\cdot)$: Controller transfer function
- e, E: Scalar error signal and a $k \times m$ matrix containing error signal values
- $f(\cdot)$: Arbitrary function
- F: Linear mapping matrix of dimension $n \times m$
- $G(\cdot)$: Transfer function (for a process, closed loop system etc.)
- i,j: Indices for vector and matrix elements
- $J(\cdot)$: Cost function
- k: Number of data samples, i.e., local iteration steps
- K: Number of global iteration steps
- K_c: Critical gain of the controller
- K_{OL}: Open loop gain of the process
- K_P: Proportional gain of the PID controller
- L: Time lag, delay
- $L(\cdot)$: Filter transfer function
- m: Number of quality measures; dimension of output space
- M: Number of latent basis vectors in output oriented subspace
- n: Number of the parameters; dimension of input space
- N: Number of latent basis vectors in input oriented subspace
- q, Q: Quality measure vector and matrix, dimensions $m \times 1$ and $k \times m$, respectively; shift operator
- r: Reference signal; setpoint
- R^n: n-dimensional linear space
- s: Laplace variable
- t: Continuous or discrete time index
- T_c: Period of the critical oscillation
- T_I: Integration time
- T_D: Derivation time
- T_R: Rise time
- T_S: Settling time
- X: Random variable
- u: Input signal, control signal
- v: Disturbance signal
- w: Weight vector of compatible size
- y: Output signal
- z, Z: Latent variable vector and matrix, dimensions $N \times 1$ and $k \times N$, respectively
- α: Arbitrary scalar constant; size of a statistical test
- β: Arbitrary scalar constant
- ϕ, φ: Eigenvectors; input and output oriented subspace basis vectors, respectively
- Φ: Set of basis vectors
- γ: Scalar step size
- κ: Data sample index

- λ: Eigenvalue; Lagrange multiplier
- θ, Θ: Parameter vector and matrix, dimensions $n\times1$ and $k\times n$, respectively
- σ: Standard deviation
- σ^2: Variance
- τ: Time constant of a process
- ψ_i: Impulse weight coefficient on lag i
- ζ: Damping coefficient of the process

NOTATIONS

- M^*: Optimal or objective value of M
- $\overline{\overline{M}}$: Nominal (prevailing) value of M
- $\hat{M}$: Estimate of M
- $\tilde{M}$: Error of M (e.g. estimation error)
- $\overline{M}$: Mean value of M
- M^T: Transpose of M
- M°: Unit vector parallel to vector M
- M_i: The ith column of the matrix M or ith element of vector M
- M_{ij}: The element on the ith row and jth column of matrix M
- $|\cdot|$: Absolute value (for scalars); Euclidean vector norm (for vectors)
- cov{·}: Covariance (matrix)
- det{·}: Determinant
- dim{·}: Dimension of vector
- E{·}: Expectation
- var{·}: Variance
- Δ: Difference of two variables

ABBREVIATIONS

- CCA/CCR: Canonical Correlation Analysis/Regression
- CPA: Control Performance Assessment
- CR: Continuum Regression
- FCOR: Filtering and Correlation (algorithm)
- IAE: Integral of Absolute value of Error
- IFT: Iterative Feedback Tuning
- ILC: Iterative Learning Control
- IMC: Internal Model Control
- IRT: Iterative Regression Tuning
- ITSE: Integral of Time-weighted Squared Error
- LQG: Linear Quadratic Gaussian
- LS: Least Squares (method)

- MCMC: Markov Chain Monte Carlo (simulation)
- MIMO: Multiple inputs, multiple outputs (model, system)
- MLR: Multilinear Regression
- MRAC: Model-Reference Adaptive Control
- MV: Minimum Variance (controller, performance index)
- MVR: Multivariate Regression
- PCA/PCR: Principal Component Analysis/Regression
- PLS: Partial Least Squares
- PPA: Process Performance Assessment
- PRBS: Pseudo Random Binary Sequence
- SISO: Single input, single output (model, system)
- SPM: Statistical Process Monitoring
- STR: Self-tuning Regulator

CONTENTS

1 INTRODUCTION

This report introduces a novel method, called Iterative Regression Tuning, for simultaneous tuning of multiple controllers and the results obtained from its first industrial scale application. The report is based on the master's thesis of Halmevaara /6/, in which new approaches initially presented by Hyötyniemi /21,22,23/ are studied and applied.

The goal of the thesis was to test the Iterative Regression Tuning (IRT) method in practice. The method has already been experimented in toy examples (see /22/). In /6/ it was tested whether the approach could be scaled up to real life industrial processes. A dynamical simulator representing a realistic power plant process was applied in the test case. This report summarizes /6/ in order to provide a comprehensible overview of the proposed tuning technique, its application possibilities and the obtained results with the example process.

The research of the master's thesis was carried out in connection with the Testing Manager project that was a cooperative research and development project of Technical Research Center of Finland (VTT) and Helsinki University of Technology (HUT) during the years 2003 - 2004. National Technology Agency (TEKES) and Finnish industry (Fortum Nuclear Services and Metso Automation) were involved in funding the project. The goal of the Testing Manager project was to provide a flexible and comprehensible environment for simulation assisted testing and tuning of industrial automation systems. This covers well-defined working practices, practical tools supporting the testing and commissioning phases, as well as open connectivity of the simulation and automation software. However, the software architectural solutions were beyond the scope of the master's thesis and thus also in this report the emphasis will be exclusively on the IRT technique.

1.1 Motivation and background

Despite all the improvements in the last decades in the field of control theory, the most common control algorithm used in process industry is still the PID algorithm. The more advanced control methods, such as, e.g., the optimal control theory, have not gained as much interest of the practicing control engineers as the less sophisticated algorithms. This is reflected as a smaller number of implementations on industrial processes. Another fact that has been recognized is that as many as about 60 % of the industrial PID controllers are behaving inefficiently or even detrimentally, i.e., far from the optimal achievable control performance /13/. This is due to the common "tuning" principle according to which some educated guesses or default values offered by the automation suppliers are used instead of any considerate way to tune the controllers. Also the amount of controllers in an industrial process is one reason that must dampen the tuning enthusiasm of the control engineers. As explained in /38/, one process engineer may be responsible for several hundred control loops.

In recent years, miscellaneous controller tuning methods and process performance assessment (PPA) techniques have drawn more and more attention. The research has mainly concentrated on single loop methods although the controllers are known to

have considerable joint effects. This usually results in rather conservative tuning of the controllers as one wants to ensure the stability and the robustness of the system. The systems that are multivariate by nature are often split into pieces and handled with tools that are developed for SISO (single input, single output) systems since the MIMO (multiple inputs, multiple outputs) systems theory is often considered too cumbersome and intangible.

Modern simulation software makes it possible to simulate the dynamic process models faster than in real time. And as the computational power of computers has increased it is possible to construct models and run simulations that are more detailed and accurate than ever. This increased simulation power introduces new possibilities in system engineering.

Modern automation and simulation systems produce huge amounts of measurement data. However, only a minority of the available data is utilized in a sensible way. Reasons for this in addition to lack of theoretical tools are the noise and the redundancy in the measurements. One inevitably faces numerical problems when noisy and collinear signals are used for modeling and analyzing the system performance. In the recent years attempts have been made to overcome these problems with different methods that can be gathered under the term *data mining*. In some cases concrete results can be reached.

1.2 General idea of the Iterative Regression Tuning

Let us examine the block diagram in Figure 1 that represents a model of a dynamic process and a system for its performance evaluation. Note that the model may well describe a rather large system, e.g., a whole power plant. Usually, the model of a system is identified in order to forecast the process responses to certain input signals. However, if the system is viewed in a somewhat wider scope and one is more interested in the *quality of the performance* rather than the actually resulting output signals, the concept of *quality measure* can be introduced /21/. Quality measures, q, are characteristic figures that measure how acceptable or desirable the performance of the system is. For instance, a quality measure could be defined as the variance of the end product properties, the efficiency in power production or the setpoint tracking ability of a controlled variable.

Here it is assumed that the systems parameters θ somehow define the current performance of the system, at least in statistical sense. Further, this unknown dependency between θ and q can be modeled if only a large enough data set of parameter values θ and the corresponding quality measures q is available. The input and output signals, u and y, can be interpreted as realizations of stochastic processes, and the performance (whether good or bad) is determined partly by the system parameters. Hence, the modeling of the originally dynamic system transforms into describing the static stochastic dependency between θ and q. After the relationship between parameters θ and quality measures q is modeled, one is able to optimize the system performance with respect to its parameters, e.g., by using iterative optimization methods such as gradient descent algorithm.

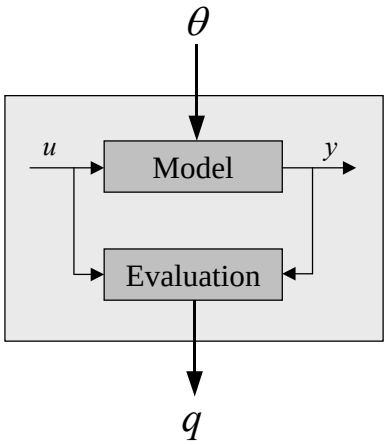

Figure 1. A dynamic model describes the response *y* of a system to an input *u*. Seen on the higher level of abstraction, the quality measures *q* that describe the system performance depend on the parameters θ /modified from 21/.

Although this report will focus on control parameter tuning, it does not imply that the IRT method is only applicable to this area. The set of parameters that is tuned does not have to be restricted to the control parameters only but it may well include any other process parameters as well, e.g., setpoint values (certain assumptions must hold however; see Chapter 4).

1.3 Introductory example

To have some intuition of how the method works an extremely simple case is studied below. Assume that a first-order process

$$G(s) = \frac{1}{\tau s + 1} \tag{1}$$

is being controlled using a P controller (see Figure 2), the proportional control parameter being *P*. The closed-loop transfer function becomes

$$G'(s) = \frac{PG(s)}{PG(s)+1} = \frac{P}{\tau s + 1 + P}. \tag{2}$$

This system can be characterized (for example) in terms of *P*, so that one can select the parameter vector as $\theta = P$. The quality of the system behavior can be measured, for example, in terms of the *steady state error*:

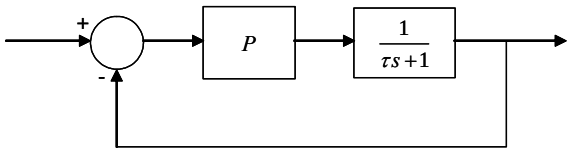

Figure 2. Simple system with one tunable parameter.

$$q = 1 - \lim_{s \to 0} G'(s) = \frac{1}{1+P}. \quad (3)$$

Now, this error can be minimized using the gradient method:

$$\theta(K+1) = \theta(K) - \frac{dq}{d\theta}(K), \quad (4)$$

or, in this case,

$$P(K+1) = P(K) + \frac{1}{\left(1+P(K)\right)^2}, \quad (5)$$

where K is the iteration index. This algorithm gives one the possibility of gradually changing the parameter to enhance the system behavior.

It needs to be recognized that the above example is extremely simple, so that analytic solutions could also be found. Iterative optimization methods are applied, since in practice the process model is not known exactly and the dependency between the parameters and the quality measures does not necessarily stay fixed over the whole parameter space as it was assumed above. Finding the process model and defining the criterion becomes more and more difficult as the complexity and the size of the system increase. Theoretical approaches often fail to scale up; the proposed methodology, on the other hand, is rather insensitive to system complexity.

Typically, the parameter – quality measure model has to be based on *data*. As will be shown, the proposed optimization idea still works for such data-based models. To master the high-dimensional data-based models corrupted by noise, new approaches have to be employed instead of the traditional control engineering methodologies, i.e., multivariate statistical methods. These new tools will also be discussed in this report.

The above example also illustrates the shortcomings of the IRT methodology. It simply implements parametric adaptation towards local minimum, assuming it exists – and in the above case, it *does not*: The value of the parameter P should be increased infinitely in the above example, resulting in problems.

It can be claimed that one of the main contributions of the proposed approach is that it makes the underlying assumptions explicit: What are the consequences of the selected criteria. In this way, the domain area expert may reach new intuition and understanding of the process. For example, when looking at the above process, one can see that the selected criterion is not good for this system; or, rather, another control structure should be selected (controller with integrative action).

1.4 Structure of the report

The structure of the report is divided into two parts. The first part (Chapters 2 – 3) consists of short literature surveys that review the current status of process performance assessment and controller tuning. In the second part (Chapters 4 - 7) the IRT method, the case study and its results are introduced along with some discussion. In Appendices A and B some mathematical tools are presented.

In Chapter 2 different approaches to process performance assessment are discussed. Several performance indices are introduced and their application as quality measures is considered. At the end of the chapter also some remarks on the multivariable extensions of performance measures are made.

Chapter 3 introduces in general terms different controller tuning techniques. In addition to the conventional tuning guidelines, auto-tuners and principles of adaptive control, also a couple of more recently proposed tuning methods are presented.

In Chapter 4 the Iterative Regression Tuning (IRT) method is explained more thoroughly. Different application practices of the tuning technique concerning different phases in the control system life cycle are also introduced. Finally, assuming that the new methodology is used for constructing a generic parameter tuning tool, some requirements and necessary functionality of the user interface and the software environment are discussed.

Chapter 5 presents the simulated process and the controllers that were tuned in the case study. Further, some comments about choosing the parameters and defining the quality measures are made.

In Chapter 6 the obtained results are presented. The validity of the assumptions concerning the IRT method is inspected and the applicability of the different multivariate regression methods in this context is discussed.

In conclusion, Chapter 7 presents the most important observations concerning the results and some interesting areas of further studies are highlighted in this section.

Appendix A shortly sums up the basic idea of statistical testing. These techniques are rather mature branch of statistics and thus an appropriate way to conduct conclusions, e.g., whether an improvement in the process performance is significant also in the statistical sense.

Different multivariate regression methods that can be used within the developed tuning technique are introduced in Appendix B. First, the ordinary Least Squares (LS) modeling is introduced, and after that, some improvements on this technique are presented.

2 PROCESS PERFORMANCE ASSESSMENT

In Introduction it was explained that the performance of a system can be evaluated in terms of quality measures q (see Figure 1). This means that suitable mathematical expressions giving numerical values for different control objectives should be defined. These quality measures naturally depend on the examined system since the performance objectives can vary considerably from one process to another. In case of controller tuning these quality measures can be associated with the concept of *control performance assessment* (CPA) and related performance indices. This chapter presents some measures reported in the literature along with traditional textbook characterization methods. In the following, the more general term *process performance assessment* (PPA) will be used instead of CPA to emphasize that it is the whole process whose behavior one is interested in optimizing.

In the last decade the PPA has drawn much attention, at least the attention of the academic people. Various techniques for evaluation of process performance have been developed and published. However, according to Harris *et al.* /13/, in 1999 only a minority of the industrial plants utilized any system for reviewing the performance of the controllers relative to their design objectives. Not until recently, i.e., during the last couple of years, the number of the implemented applications in the industry has been increasing /19,26/.

CPA is just one part of the process monitoring task that includes also diagnosis, isolation and clearing the faults. According to Stanfelj *et al.* /34/, monitoring the control loops performance is a matter of defining the best achievable (or desirable) performance, testing whether this is achieved and finally determining the steps to improve the current performance. For that reason it should be kept in mind that the concept of CPA is quite useless alone if the successive steps (tuning or changing the control strategy) are not considered.

2.1 Traditional characterizations

Traditionally, the evaluation of process performance has been based on deterministic measures, such as overshoot, rise time, settling time and decay ratio. Most of these measures, or indices, are meant to characterize the response of the system to a setpoint change or load disturbance. Typically, the indices have many slightly different definitions. In process industry fast recovering capability after a load disturbance (or insensitivity to load disturbances in the first place) is usually considered as good performance. Responses to set point changes are, on the other hand, not that much of interest and often it is sufficient to have smooth responses without excessive overshoot. However, for controllers that operate as slave controllers in cascade configurations also the setpoint tracking ability is important. The above discussion reflects the fact how the control performance objectives vary substantially from one control loop to another.

2.1.1 Deterministic measures

The performance measures mentioned above are illustrated here with an example. Figure 3 presents a step response of an underdamped 2nd order system to a setpoint change.

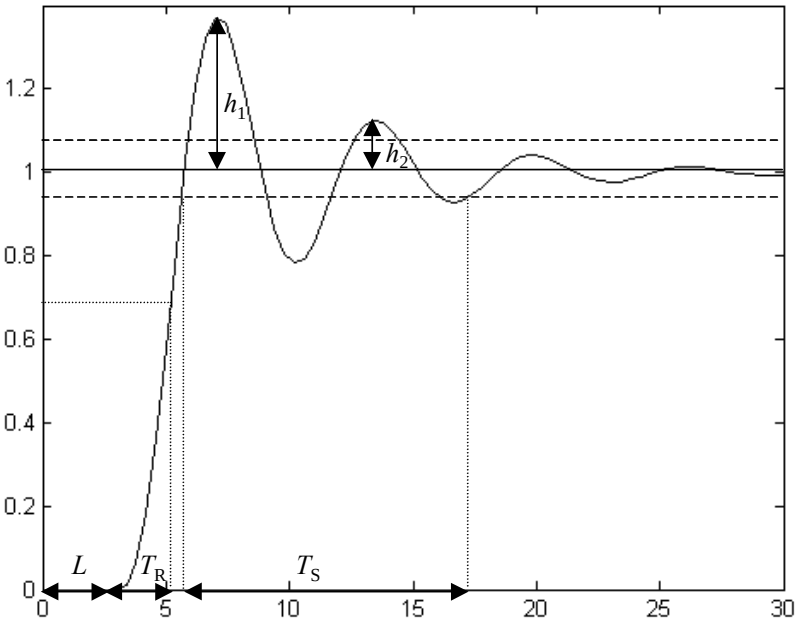

Figure 3. A step response to a setpoint change of $\Delta r = 1$. T_S is the settling time, T_R the rise time, L the dead time, and h_1 and h_2 are the heights of the successive oscillation peaks of the response.

The magnitude of the overshoot h_1 is usually expressed proportional to the size of the setpoint change Δr and its value is given in percents,

$$OS = \frac{h_1}{\Delta r} \cdot 100\% . \qquad (6)$$

The rise time T_R is determined in the literature in many ways. One typical definition is the 68% rise time. If it takes t time units after the setpoint change until the response has risen 68 percents of Δr, the rise time is

$$T_R = t - L , \qquad (7)$$

in which L is the dead time.

The settling time T_S is usually defined as the time that is needed until the values of the controlled variable settle down inside some predefined limits, e.g., inside 2% error margin (see Figure 3).

The decay ratio DR measures how fast the oscillations die out after a load disturbance and it is defined as a ratio of two successive oscillation peaks

$$DR = \frac{h_1}{h_2}. \tag{8}$$

Controlled systems are also characterized based on their frequency responses. E.g., such measures as bandwidth and resonance frequency of a closed loop system describe the frequency interval where the disturbance rejection is successful and the frequency of maximal amplification, respectively. These measures are, however, mainly used in controller design rather than in PPA. (For more details, see, e.g., /4/.)

2.1.2 Error signal integrals

One typical group of process performance characteristics is the error signal integrals of the form

$$\int_0^\infty t^\alpha |e(t)|^\beta \, dt \,, \tag{9}$$

where the error $e(t)$ is defined as the difference between the setpoint value and the process measurement. For example, the *IAE* index (Integral of absolute value of error) is obtained with $\alpha = 0$ and $\beta = 1$ whereas the *ITSE* index (Integral of time-weighted squared error) with $\alpha = 1$ and $\beta = 2$. These kinds of measures are usually considered quite problematic because they do not offer any limits for the index values, which make their interpretation difficult for a human. (However, see the discussion in Chapter 2.4). Thus, the process performance is often measured against some kind of a benchmark.

One natural way to overcome the problem of unlimited and hard-to-interpret index values is to use a reference model as a benchmark. This means that the system performance is compared to the objective model used in controller design. Such a working practice helps in decoding the obscure figures produced by the performance indices and clarifies how far from the target the prevailing performance is. E.g., the value of *IAE* index is much more comprehensible if one knows also the index value for an objective model in the same situation. Reasonable selection of the objective model is, nevertheless, far from trivial.

2.2 Examples of PPA indices

The PPA boom in the academic control theory society has resulted in a number of special purpose PPA indices. Common to all these indices is that they try to express the goodness of the process performance in a form that is somehow easier to understand than the original time series signals produced by the automation system. In a way it is a matter of highlighting some interesting phenomena and the aim is to assist the monitoring task, as the number of control loops in an industrial application may be several dozens. In this chapter a couple of control loop characterization methods are introduced.

2.2.1 Oscillation index

One typical form of bad process performance caused by improper controller tuning is continuous oscillation of the controlled variable. Oscillations cause increased energy and raw material consumption and non-uniform end product quality. In /24/ Hägglund

proposes an automatic oscillation detection procedure. The method studies the value of *IAE* index calculated between successive setpoint crossings of the controlled variable (assume that these happen at instants t_{i-1} and t_i):

$$IAE = \int_{t_{i-1}}^{t_i} \left| e(t) \right| dt \,. \tag{10}$$

Every time the observed value of *IAE* is greater than a predefined threshold IAE_{limit}, one can conclude that a load disturbance has occurred. By using exponential weighting the number of occurred load disturbances is summed together. If the occurrence frequency is high enough, the number of detected load disturbances exceeds the value of parameter n_{limit}, which indicates that the loop is oscillating. Hägglund gives guidelines for selecting the parameters IAE_{limit}, exponential weighting coefficient and n_{limit}. The detection procedure has been implemented in an industrial process controller and, according to Hägglund, it has given good results in practice. Hägglund also points out that there are many possible reasons for the oscillation in the control loop along with the poor controller tuning. The most typical reason is the friction in the control valve. Another reason might be an oscillating load disturbance that may result from another oscillating control loop. In /36/ Thornhill and Hägglund propose some methods for the characterization of oscillations.

2.2.2 Idle index

According to Hägglund /25/ most of the controllers in process industry are rather conservatively tuned to avoid instability and oscillations in varying operating points. Consequently, these controllers give sluggish responses to load disturbances, which means long-lasting deviations from the setpoint and thus even decreased product quality in the end. Thus, he introduces the Idle index for detecting the sluggish control loops. First, the time periods, when the correlation between the signal increments (derivatives) of control signal and process measurement is positive and negative, are determined:

$$\begin{cases} t_{\text{pos}} = \begin{cases} t_{\text{pos}} + h, & \text{if } \Delta u \Delta y > 0 \\ t_{\text{pos}}, & \text{if } \Delta u \Delta y \le 0 \end{cases} \\ t_{\text{neg}} = \begin{cases} t_{\text{neg}} + h, & \text{if } \Delta u \Delta y < 0 \\ t_{\text{neg}}, & \text{if } \Delta u \Delta y \ge 0 \end{cases} \end{cases} \tag{11}$$

where h is the sampling interval. Then the Idle index is defined by

$$I_{\text{I}} = \frac{t_{\text{pos}} - t_{\text{neg}}}{t_{\text{pos}} + t_{\text{neg}}} \tag{12}$$

and its values are bounded to the interval [-1,1]. Positive values close to 1 suggest that the tuning is sluggish whereas negative and small positive values ($I_{\text{I}} < 0.4$) indicate acceptable controller tuning. However, negative values close to -1 are also obtained if the loop is oscillating. In the calculation of the index it is assumed that the sign of the static process gain is known and the load disturbances should be steps or at least abrupt. The method is also sensitive to noise and therefore it is important to filter the

signals. The detection procedure of the sluggish control loops can be applied both on-line and off-line.

2.2.3 Minimum variance index

The behavior of the process is not completely explained by the deterministic models. The actual process values always differ somewhat from the assumed performance, no matter how precise the model is. This is due to the random phenomena (stochastic disturbances, noise) affecting the process, its control system and instrumentation devices. E.g., it is unlikely that the process measurement ever meets its expected value exactly in a steady state but varies randomly around it. Thus it is convenient to have some tools for evaluating the system performance also in the stochastic framework. Many basic statistical characteristics, such as variance, auto- and cross-correlation, turn out to be quite useful and they can be used for many diagnostic purposes.

For regulatory control it is reasonable to observe the variance of the controlled variable. However, also the values of the variance are unbounded and thus incommensurable, just as the error signal integrals discussed earlier. In this case a sensible benchmark is the variance achieved with the so-called minimum variance controller. Harris first proposed this practice in 1989 /11/. The minimum variance index (MV index) or the Harris index is usually expressed either as

$$\xi(L) = \frac{\sigma_y^2}{\sigma_{\mathrm{MV}}^2}, \tag{13}$$

where the index ranges $\xi(L) \geq 1$, or

$$\eta(L) = 1 - \frac{\sigma_{\mathrm{MV}}^2}{\sigma_y^2}, \tag{14}$$

where $0 \leq \eta(L) \leq 1$ /13/. In (13) and (14) L is the dead time, σ_y^2 is the observed variance of the controlled variable and σ_{MV}^2 is the minimum achievable variance. σ_{MV}^2 can be defined as

$$\sigma_{\mathrm{MV}}^2 = \left(1 + \psi_1^2 + \cdots + \psi_{L-1}^2\right)\sigma_a^2, \tag{15}$$

where σ_a^2 is the noise variance and ψ_i is the impulse weight on lag i. Equation (15) suggests that the controller cannot influence the MV part of the total variance by any means due to the dead time L. Instead, all of the impulse weights after the lag L are zero if a minimum variance controller is applied. This reflects the biggest drawback common to MV based PPA methods: A priori knowledge of the process dead time is required. If the estimate of the dead time is inaccurate, the method will inevitably provide rather poor results. Many techniques for the calculation of the MV index in practice have been proposed in the literature. E.g., Huang and Shah /17/ introduced a filtering and correlation based method, FCOR, to estimate σ_{MV}^2 and $\eta(L)$.

The idea of using minimum variance control as a benchmark does not imply that the performance of the minimum variance controller would always be desirable or even possible. Under minimum variance control the manipulated variable (or the control signal) is assumed to work very aggressively in a large range, which may not be

possible in practice. And if the process transfer function is non-invertible, it is impossible to implement a minimum variance controller. Still, the minimum achievable variance serves as a good benchmark because it offers a theoretically justified lower bound for the variance. Based on the idea of Harris, further development has been made in order to obtain good and easy-to-calculate PPA indices. These improvements are reported, e.g., in /12,14,34/.

In /34/ Stanfelj *et al.* present a hierarchical PPA method that first identifies the deviation from control objectives, and when necessary (i.e., when a noticeable deviation is found), it determines the minimum achievable variance with the current control structure and the steps needed to improve the process performance. The method is based on statistical analysis of the plant time series data using the autocorrelation and cross-correlation functions.

2.3 Multivariate PPA

Although the above presented process performance measures are developed for the SISO systems, some of them have been experimented also in assessing of MIMO systems. The goal of this attempt is comprehensible but in most of the cases designing MIMO extensions is not a straightforward task.

In /14/ an expert system is used to assess control loops in the whole plant. It collects sets of data from the plant, evaluates the current control performance of the loop and compares it to the previous results. The system archives relevant performance data and reports about the discovered problems. In this application the process performance was measured with the before mentioned Harris index.

PPA methods of truly multivariable control systems have also been developed. For example, in /12/ Harris *et al.* present an extension of the minimum variance index to multivariable case. Similarly as in the SISO case, the calculation of performance bounds for MIMO systems requires knowledge of the delay structure of the system, i.e., the interactor matrix, which may become a problem in some cases. In /18/ Huang *et al.* propose an extension of the FCOR method to the MIMO systems. Also their approach assumes the interactor matrix to be known.

2.4 General remarks on the PPA indices

One defect common to all PPA techniques is that they only provide the answer whether something should be done or not. They do not solve the actual problem of controller tuning. Of course, control performance indices support the tuning task as they point out which controllers are performing badly and in which manner. Still, the actual job of choosing the new parameters is left to the control engineer responsible for the tuning.

In /13/ Harris *et al.* remind that selecting appropriate and applicable PPA measures is not an easy task in an industrial setting: It is always a trade-off between the complexity of the method, its invasiveness and the information content it offers (see Figure 4). One has to consider whether one method gives crucial and adequate information about the control systems performance with respect to the control objectives. At the same time, if one strives for accurate and informative monitoring methods, the system should be modeled to a sufficient precision, which requires dynamic experiments on the process and thus inevitably interference in the production. One should also

consider the amount of complexity (regarding the computational effort and a priori process knowledge) that is justified.

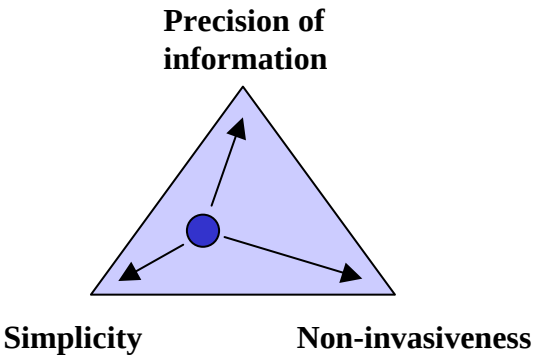

Figure 4. The trade-off concerning the selection of PPA indices in practice /modified from 13/.

The controller and its parameters are naturally not the only factors affecting the system behavior. The process dynamics (e.g., non-invertible zeros, dead time, non-linearities), disturbances, limits on the manipulated variables and the operating point also have their influence on the performance of the system. These factors differ from process to process and thus it is difficult to set any general definitions for good process performance. Also the control objectives are different in every application, for example, servo vs. regulator problems.

The PPA is a continuous task that cannot be accomplished in one go. Wear and other malfunctions in the control system, e.g., in sensors and actuators, cause a constant drift to the optimal control parameters. These characteristics should be kept in mind in the PPA task and one should monitor the changes in the process performance rather than just the present behavior.

2.4.1 PPA indices as quality measures

If one wants to use some of the preceding PPA indices as quality measures q (discussed in Chapter 1.2) they must fulfill some requirements. First of all, the quality measures have to be continuous functions of the qualifiers θ. Furthermore, these functions should behave such that they do not perform any abrupt changes, i.e., they should be “smooth”. The reason for this is obvious, if one aims to use the gradient descent algorithm in the optimization or any other method that involves differentiation.

Figure 5 presents four different quality measures. Quality measure q_1 is totally inappropriate because it is non-differentiable. Also q_2 would be a problematic choice since it is impossible to determine the gradient direction of this function in every point of the θ axis. Instead, q_3 and especially q_4 are suitable for the purpose in this context.

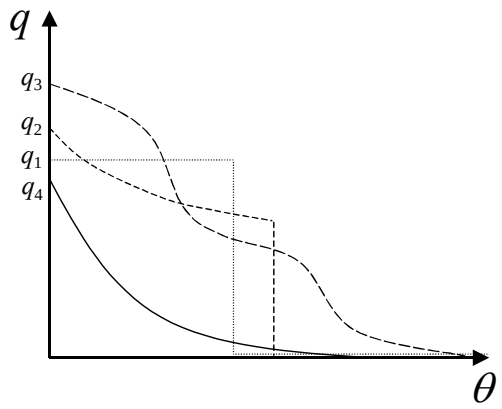

Figure 5. The quality measures q_3 and q_4 represent good examples of quality measures whereas one cannot determine the gradient direction of q_1. Also quality measures of type q_2 are problematic.

Most of the performance measures described above, e.g., the MV index, the Idle index and the Oscillation index, are defined such that their values range over some specific interval. In this manner the relevant information gained from the data can be compressed to a single (bounded) numerical value that is easier to interpret and compare by a human mind. E.g., it is quite hard to say whether 100 or 1000 are big or small values, if there is no information about the overall magnitude. Instead, if one knows that the values range from 0 to 1000, it is much easier to rate the index value 100. However, if the PPA indices are used as above-described quality measures, the scaling of their values with some kind of benchmarks is not required. Because an algorithm handles the interpretation of the values, the situation is completely different as compared to traditional approaches. Mathematical machinery is able to obtain a picture of the correlation structure of the data, no matter whether the quality measure values are bounded to some interval or not. And in fact, the less the data is crushed the better the results are.

3 CONTROLLER TUNING TECHNIQUES

As Stanfelj *et al.* argued in /34/, CPA is only one step forward if the process performance enhancement is pursued. After diagnosing which control loops are behaving badly and in which manner, the actual tuning task can begin. Traditionally, the developed controller tuning methods have mainly been concentrating on the SISO systems. Later on, the results have been adjusted to the needs of the MIMO systems as well. The concentration of the research almost exclusively on the SISO systems and to their tuning is easy to understand if the status of the PID controller is considered: More than 90 % of all control loops are of PID type /13/. E.g., many of the multivariable control systems operate in cascade mode such that the multivariable controller is providing set point values for the lower level PID controllers. This makes the PID controllers essential building blocks of multivariable control systems and their proper tuning is a necessity for the satisfactory performance of the overall system.

When considering any alternatives to PID controller one always ends up with the same problem: Advanced control algorithms, e.g., the general linear controllers, are more troublesome to design and tune. The number of the control parameters increases as the controller structure becomes more complicated and thus the appealing simplicity of the PID controller is lost. Usually, the tuning methods for the advanced controllers assume that the whole process is fairly well known. This means that the transfer function(s), step or frequency response(s) characterizing the behavior of the system should be modeled at least to a certain precision.

3.1 Conventional tuning principles

The early tuning methods were in practice only guidelines to control engineers, who were supposed to do the tuning of the controllers, i.e., one relied on the experience of the practicing control engineers. The dominant status of the PID controller as compared to the other control strategies implemented in the industry must have contributed to the slow development and adoption of the controller tuning methods: Three intuitively understandable parameters of the PID are still quite easily adjusted in a sensible way without fancy tuning tools. This might be the case at first sight, but as the number of these simple PID controllers operating in the same system increases, the tuning becomes more and more complex task due to interactions between controllers.

3.1.1 Ziegler-Nichols tuning

Ziegler and Nichols gave the first and still commonly used tuning principles of the PID controller in 1942. These general rules of thumb are still widely used although it has been recognized that they are not able to offer acceptable performance in many cases. E.g., long dead time may be a reason for the unsatisfactory control performance if the controller is tuned with the Ziegler-Nichols principles. The reason for the popularity, despite all the disadvantages, may be the simplicity of these rules.

In the Ziegler-Nichols tuning the critical point of the frequency response is first determined. In practice, this point can be found by increasing the gain of the purely proportional controller until the controlled system reaches the stability limit and starts

to oscillate. By denoting the oscillation period with T_c and the corresponding gain of the P controller with K_c, the Ziegler-Nichols choice for the PID parameters is

$$K_P = \frac{K_c}{1.7}, \quad T_I = \frac{T_c}{2}, \quad T_D = \frac{T_c}{8}, \tag{16}$$

for a PID controller of the form

$$C_{PID}(s) = K_P\left(1 + \frac{1}{T_I s} + T_D s\right), \tag{17}$$

where K_P, T_I and T_D are gain, integration and derivation time, respectively. The disadvantages of this tuning method are quite obvious: It is rather laborious to tune many controllers one by one and, furthermore, driving the process to stability limit is neither practical nor appropriate way of enhancing the performance. And after all, the Ziegler-Nichols method does not take into account the individual control objectives of a loop but assumes that a certain controller tuning could be satisfactory for every single PID controller.

3.1.2 Internal Model Control (IMC)

Another control design method that has been used also for fixed structure controller tuning is the internal model control (IMC) method. There are many variants of this method that can handle also more general model structures, but in this context it is adequate to examine one simple example. If a system is approximated using a first order plus time delay model of the form

$$G(s) = \frac{K_{OL}}{1 + s\tau} e^{-sL}, \tag{18}$$

and the model parameters K_{OL}, τ and L (open-loop gain, time constant and time delay of the process, respectively) are determined from an open-loop step response, the IMC tuning for a PID controller of the form (17) is obtained with

$$K_P = \frac{2\tau + L}{2K_{OL}(\alpha + L)}, \quad T_I = \tau + \frac{L}{2}, \quad T_D = \frac{\tau L}{2\tau + L}, \tag{19}$$

where α is a tuning parameter corresponding to the desired closed-loop time constant /29/. In many cases α is denoted with λ which gives rise to the name *lambda tuning*. In most cases the IMC design technique results in reasonably good control structures. Also, it is quite easy to use as there is only one tuning knob. Problems may occur on processes with non-invertible zeros or long time delays that are approximated, e.g., with Padé approximations. And, especially, if the process is assumed to follow a model type that is incorrect, the resulting control performance is unlikely desirable.

3.2 Automatic tuning techniques

A few decades ago the automatic tuning of controllers became a popular research topic. In 1970's and 1980's the intensive research work resulted in a wide variety of

approaches to automatic tuning. Automatic tuning (or auto-tuning) means methods for automatic control parameter tuning on demand from the operator.

One of the first auto-tuning techniques, originally proposed by Bristol /2/, was based on pattern recognition. This method monitors the behavior of the error signal after a load disturbance or a setpoint change. The observed response is compared to the user-specified objective by using distinctive features, such as peaks and troughs, time between peaks, and the steady state error. A major drawback of this method is the need of consecutive setpoint changes or load disturbances for successful auto-tuning of control parameters. Also, determining a reasonable objective behavior for the process may be difficult.

Hang *et al.* present in /10/ an extensive state-of-the-art review of relay auto-tuning that is another common auto-tuning technique. According to them one of the first relay feedback auto-tuning methods that was commercialized was the one proposed by Åström and Hägglund in 1984. Neither open-loop tests nor large setpoint changes were required when using this auto-tuning method, which was a great improvement as compared to its predecessors such as the Bristol method. However, this method focuses on simple SISO controllers. The tuning is based on the estimation of the process frequency response at the critical frequency and applying then the Ziegler-Nichols tuning principles. This means that the relay auto-tuning shares the same disadvantages as the applied tuning principle (see Chapter 3.1.1).

Later on, many improvements on this technique have been reported, e.g., modifications for processes with long time delays and oscillating dynamics. There have also been attempts to generalize the method to more advanced controllers, such as cascade controllers and Smith predictors. E.g., Hang presents in /7/ an extension to conventional relay feedback auto-tuning that can be used to online tuning of cascade controllers.

The relay auto-tuning methods have been experimented also on multi-loop PID control systems and multivariable controllers. These attempts have been reviewed in /10/. Luyben, e.g., presented an iterative auto-tuning approach for multi-loop PID controller /30/. This method tunes the multivariable system loop by loop by using sequential relay tuning approach and Ziegler-Nichols tuning principles. One drawback of this method is that the process dynamics should be known to some extent, either in transfer function or frequency response form. This makes the method dependent of the process at hand and the generality (process independence) is lost. Another unfavorable feature is that the method is limited to open-loop stable systems only. And, finally, the stability of the system cannot be guaranteed with this auto-tuning method and therefore a heuristic “detuning factor” is introduced.

It can be concluded that common to all relay feedback auto-tuning methods is their applicability to only certain types of processes and controllers. Furthermore, they are a bit cumbersome to apply for tuning of multivariate systems.

A third way of finding the critical point of the process frequency response is described by Hang and Sin /9/. They use the cross-correlation of a pseudo-random-binary-sequence (PRBS) test signal and the process output to calculate the impulse response of the system. This is numerically transformed into frequency response and finally the controller tuning is performed with the Ziegler-Nichols rules. This procedure has some advantages, e.g., it operates on-line in closed loop system and it does not require

operation near the stability boundary. But then, the use of PRBS test signal induces extra perturbation in the controlled variable that is, of course, undesirable.

The three different approaches to auto-tuning (relay auto-tuning, pattern recognition and cross-correlation based auto-tuner) are compared with three different process models by Hang and Sin in /8/. It was discovered that relay feedback auto-tuner produced the most conservative tuning results in every case. The cross-correlation based auto-tuner using the refined Ziegler-Nichols tuning formulas gave the best responses to both setpoint changes and load disturbances.

3.3 Adaptive control

Few decades ago another progressive research area explored the adaptive control structures. The concept of adaptive control departed from the auto-tuners such that the control parameters were tuned automatically online without operator intervention. According to Åström and Wittenmark, an adaptive controller is a controller that modifies its behavior in response to changes in the process dynamics and disturbance characteristics /39/. Figure 6 presents the basic structure of the adaptive control. The reasoning concerning the adjustment of the control parameters is based on setpoint, control and output signals of the control system. In practice, a model of the plant is estimated based on the signals and the controller is tuned respectively to give the desired response. The methods can be divided into direct and indirect methods. In the indirect methods the process is identified online and the identified process parameters are used for solving the underlying controller design problem, i.e., the control parameters. A direct adaptive method means that the controller is directly parameterized in terms of the model parameters. In the following, a couple of different adaptive control schemes are outlined based on the classification presented in /39/.

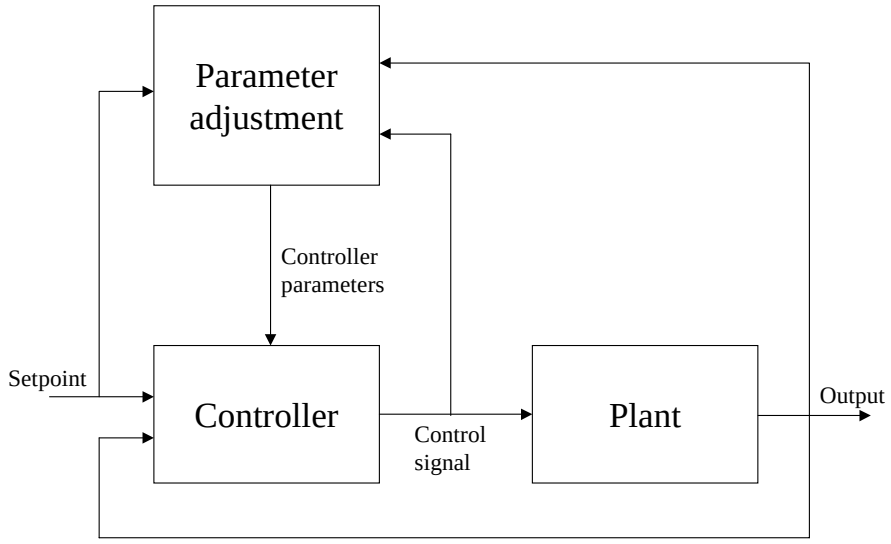

Figure 6. The basic structure of adaptive control /39/.

3.3.1 Gain scheduling

The gain scheduling was initially developed for the flight control systems. The basic idea of the control strategy is to change the control parameters as the operating point changes. Normally, it is rather easy to find a measurable process variable that indicates the changes in the operation conditions. For example, production rate of a plant can be

chosen as a scheduling variable since the time delays and the time constants of a system are typically inversely proportional to the production rate. This simple system can be implemented, e.g., as a lookup table into which suitable parameter values corresponding to different operating points are stored beforehand.

3.3.2 Model-Reference Adaptive Control (MRAC)

If the process performance specifications are given in the form of a reference model, one is able to calculate the desired system response to any input. As this desired output $y^*(t)$ is compared to the actual response of a system $y(t)$, one can use, e.g., the *MIT rule* for updating the parameters, i.e.,

$$\frac{d\theta}{dt} = -\gamma\tilde{y}\frac{\partial\tilde{y}}{\partial\theta}, \tag{20}$$

in which θ are the control parameters, γ is the update step size and $\tilde{y} = y - y^*$ is the error between actual and desired response. This parameter adjustment mechanism is the one originally used in the MRAC. The MIT rule can be interpreted as a gradient method approach to minimize the squared error $\tilde{y}^2$.

3.3.3 Self-Tuning Regulators (STR)

In the Self-Tuning Regulators the process parameters are estimated online and the corresponding control parameters are calculated by solving the controller design problem with the estimated process parameters. The so-called c*ertainty equivalence principle* is applied, i.e., the estimates are used as if they were the true parameters. The STR scheme is flexible with respect to the techniques used for identification and controller design tasks.

3.4 Iterative feedback tuning

Recently, the *Iterative feedback tuning* (IFT) method has gained much interest and many successful applications have been reported. The method was originally suggested by Hjalmarsson *et al.* in /16/. The basic idea in the IFT method is to find the control parameters θ^* that minimize the LQG type design criterion, i.e.,

$$\theta^* = \arg\min_{\theta} J(\theta) \tag{21}$$

$$J(\theta) = \frac{1}{2}\mathrm{E}\left\{\left(L_y\tilde{y}(\theta)\right)^2\right\} + \frac{\alpha}{2}\mathrm{E}\left\{\left(L_u u(\theta)\right)^2\right\}, \tag{22}$$

in which L_y and L_u are frequency weighting filters, $\mathrm{E}\{\cdot\}$ stands for taking mathematical expectation, α expresses the relative importance of the restriction on the control signal compared to the limitation of $\tilde{y}$ that is the error between the achieved $y(\theta)$ and desired y^* response:

$$\tilde{y}(\theta) = y(\theta) - y^*. \tag{23}$$

The notation $y(\theta)$ emphasizes that the achieved response is assumed to be a function of the control parameters θ as the process disturbance and the reference signal are

assumed to be realizations of independent stationary stochastic processes. This model reference problem becomes an ordinary LQG tracking problem if the desired response is set equal to the reference signal.

Typically, minimization of $J(\theta)$ with respect to θ cannot be solved analytically since $J(\theta)$ may depend on θ in a rather complicated way. Their mathematical dependency involves the expressions of the true system G and the disturbance characteristics v, both assumed to be unknown. However, the solution can be found iteratively, e.g., by means of the gradient descent algorithm, according to which the new parameters for iteration step $K+1$ are

$$\theta(K+1) = \theta(K) - \gamma_K R_K^{-1} J'(\theta(K)), \tag{24}$$

where R_K is a positive definite matrix (e.g., an identity matrix or an estimate of the Hessian of $J(\theta)$) and γ_K is the step size (the subscript K means that R and γ are not necessarily assumed to stay constant on each iteration step). The gradient of the design criterion $J'(\theta)$ in (24) is troublesome to calculate exactly and therefore it is replaced with an approximation that is calculated based on a data sample: If the signals $\tilde{y}(\theta)$ and $u(\theta)$, and their gradients $\tilde{y}'(\theta)$ and $u'(\theta)$, are known, $J'(\theta)$ can be approximated with (assuming $L_y = L_u = 1$)

$$\begin{aligned} J'(\theta) &= \mathrm{E}\{\tilde{y}'(\theta)\tilde{y}(\theta)\} + \alpha \mathrm{E}\{u'(\theta)u(\theta)\} \\ &\approx \frac{1}{k}\sum_{t=1}^{k}\left(\tilde{y}(t,\theta)\hat{\tilde{y}}'(t,\theta) + \alpha u(t,\theta)\hat{u}'(t,\theta)\right) \end{aligned} \tag{25}$$

where k is the number of the data points, t is the discrete time index and $\hat{\tilde{y}}'(t,\theta)$ and $\hat{u}'(t,\theta)$ are the estimates of the gradients of the error and control signals with the current parameters θ, respectively. These gradients cannot be solved analytically because they depend on the unknown system G. In /16/ an approach is presented to approximate these signals. A special "gradient experiment" is required on a process to obtain unbiased estimates of these gradients. Hjalmarsson *et al.* give also instructions for selecting an appropriate size for the iteration step γ.

Altogether, the above-described IFT tuning method gives a rather good control performance as compared to three conventional tuning methods (Ziegler-Nichols, ISE and IMC tuning), as is pointed out by Lequin *et al.* in /29/. According to them the IFT tuned controller can deal with setpoint changes, disturbances and model mismatch (e.g., due to changes in the process after the tuning has been accomplished) resulting in a very good performance with a reasonable control effort.

In practice the method requires experiment data generation with the true process on each iteration step. However, these experiments can be executed within normal operation of the system, although they require manipulations of the reference signal. One can use this tuning method also quite restfully, according to Hjalmarsson *et al.*, as it is proven that for a small enough iteration step size and large enough data set the method always converges towards a (local) minimum. Another advantage of IFT is that a model of the system is not required. An estimate of the gradient direction of the design criterion can be calculated based on closed loop measurement data that is obtained with an experiment arrangement described in /16/.

In /5,15/ Gunnarsson *et al.* and Hjalmarsson & Birkeland have investigated the possibilities to use IFT in tuning MIMO systems. Gunnarsson *et al.* use a 2×2 system as an example. In their approach the decoupling controllers (the non-diagonal elements of the transfer function matrix) are first tuned separately one element at a time, after which the diagonal controllers are tuned simultaneously. Hjalmarsson and Birkeland show that as only one additional gradient experiment is required to tune all the control parameters within a SISO controller, in MIMO systems the number of extra experiments rises to $n \times m$, where n and m are the number of control signals and measured outputs, respectively. In addition, in the MIMO extension of the IFT /15/ the elements of the transfer function matrix are tuned one by one. This kind of procedure obviously results in a non-optimal solution, as the overall performance of the MIMO system is not considered.

The above described IFT method has many similar features as the Iterative Regression Tuning method. E.g., the basis assumptions that the control parameters define the performance of the system in the long run (at least to a certain extent), and the interpretation of measurement signals as realizations of stochastic processes are common to both methods. Further, the iterative data based approach to find the local optimum is also similar. In a way, the IRT method presented in this report is a generalization of the IFT method. The IRT method does not restrict to LQG type design criterion but allows the user to define the tuning targets.

3.5 Iterative learning control

Batch processes are rather common especially in the chemical industry in manufacturing special chemicals, such as pharmaceutical products and polymers. The operation of a batch process differs quite a lot from the continuous processes. Running consecutive batches introduces strong nonlinearities to the plant behavior, which makes the use of a linear controller inadequate. Thus, a method called *Iterative learning control* (ILC), initially developed for training and controlling of robots and other mechanical systems under repetitive operations, have been applied to batch process control.

ILC is a control technique that is developed to improve the transient tracking performance of the process during identical, repeatedly executed operations /28/. The objective is to find an input signal (or profile) for the next batch run, based on the information gathered from previous batches, such that

$$\left| e_k(t) \right| \to \mathbf{0} \text{ as } k \to \infty . \tag{26}$$

In (26) $e(t)$ represents the (finite length) error signal of the controlled variable and k is the batch number. For instance, the input profile can be updated from batch to batch according to

$$u_k(t) = u_{k-1}(t) + Le_{k-1}(t), \tag{27}$$

where L is called the learning filter that is represented by some dynamic filter $L(s)$ or $L(z)$. Thus, the problem of the ILC design reduces to finding a suitable learning filter L. Initially, generic fixed structure filters were applied whose parameters where tuned to achieve the convergence of the error signal. Alternatively, model based algorithms

have also been studied, in which one tries to find the learning filter L based on the direct inversion of the plants input-output transfer function G, such that $L = G^{-1}$.

According to Lee & Lee /28/, even nowadays most industrial batch processes are operated with rather elementary sequence controls and many of them still require occasional manual operation. Introducing ILC type methods to batch process control has given promising results. However, further challenges still remain to be studied, e.g., treatment of unequal batch lengths.

The novel tuning technique that was originally presented in /21/ has certain similarities with the ILC approach. In the proposed controller tuning technique, at least when the simulator based approach is applied, the same “batches” are repeated over and over again, and the parameters are tuned accordingly. The problem setting is, however, completely different.

4 ITERATIVE REGRESSION TUNING

As the preceding review of the common controller tuning techniques in Chapter 3 showed, there exist actually rather few multivariate tuning techniques that are applicable to any multivariable control structure. And, further, it seems to be hard to find a tuning method that would take into account also the interactions of the multiple single loop controllers. In the majority of the multi-loop tuning techniques proposed in the literature the controllers are tuned one loop at a time. This is a motivating starting point when a new multivariate tuning method is to be proposed.

The underlying idea of the Iterative Regression Tuning method was already introduced in Chapter 1.2. In this chapter the method is introduced in more detail. First, the role of dynamic simulation in the controller tuning is discussed in Chapter 4.1. In Chapter 4.2, the tuning method is introduced in a framework that focuses on finding the initial tuning for the controllers during a plant start-up. This introduction is founded on the publications of Hyötyniemi /21,22,23/. Then, several other ways to apply the same tuning technique are presented in Chapter 4.3. Finally, in Chapter 4.4, the structure of the software application and the use of the resulting tuning system are considered.

4.1 On simulation and its application in controller tuning

The tuning method that is presented in the following chapter utilizes dynamic simulation, instead of real process measurements, to obtain an insight of the system behavior. This approach differs somewhat from the techniques presented in Chapter 3 that usually involve the actual process in the tuning task. Therefore the IRT method could be characterized as an “off-process” method.

The use of simulation has many benefits and, unfortunately, some drawbacks as well. When employing simulation the production on the real process is not disturbed with the experiments. This is naturally desirable, if the economical aspects and the safety of the plant are considered. It also means that the amount of the experiments is not restricted, which can be the case when experimenting on the real process. Another advantage is that running simulations can be performed faster than in real time. Further, running consecutive test cases with the simulator is much easier (and also faster) since snapshots of a certain initial state of the process can be reloaded instantaneously. In fact, it is rather impossible to conduct exactly identical repetitive experiments on a real process since some stochastic variation is always present.

On the other hand, the model of the system never exactly equals the real process and therefore the obtained tuning results are not directly applicable in practice. However, it can be assumed that nowadays the modeling precision has increased to a sufficient level thanks to advanced simulation software and increased computing capacity. Therefore, the provided results are at least highly indicative although not precisely accurate.

Indeed, the proposed methodology could be based on the actual process measurements rather than on simulations as will be discussed in Chapter 4.3. In this research project, however, the role of simulation was deliberately emphasized. One of the basic

assumptions was that the importance of the simulation will increase in the future: For instance, if the process design phase output the simulation models for the following automation system design, it would open the possibility to compare different control strategies easily by evaluating their performances in various situations.

The proposed tuning technique is an excellent example of *computationalism*. This term is used for describing the modern approach to tackle large problems with the increased computing capacity of computers, rather than employing increased human contribution. The IRT method involves iteration in three distinct levels: First, during a simulation run the state of the dynamic simulator is solved iteratively for each simulation time step t. Secondly, to obtain enough data points around the prevailing parameter values, the same simulation run is repeated k times, i.e., *k local iterations* are performed for different parameter values. And, on the highest level, a certain amount of *global iteration* steps is required to optimize the tuning of the parameters until the performance of the system meets its objectives. In the following, K is used as a symbol for the number of performed global iteration steps. As the above discussion implies, a huge amount of computation is involved when using this technique.

4.2 Method description

Let us return to Figure 1 on page 9 that illustrated a dynamic system, hereafter denoted with G, and a system for its performance evaluation. If an input signal u is introduced to the system G, it evokes the response signal y (note that G can be a MIMO system). The performance of the system can be evaluated by calculating *quality measures* q based on the input and output signals. In a statistical sense, the resulting performance expressed by means of q is more or less the same as long as the system parameters θ are held constant. In other words, the performance is assumed to be a function of the parameters, $q = f(\theta)$. This function includes some uncertainties due to the stochastic variation in the measurement signals. However, it still gives us the opportunity to improve the performance of the system G by optimizing the values of the parameters θ.

What are these quality measures in practice then? Defining the quality measures q in a sensible way requires always the knowledge and the assistance of a domain-area expert. If one uses control parameters as the qualifiers θ, it might be reasonable to measure the quality by means of control performance concepts like stability, speed, robustness, accuracy, etc. The set of chosen quality measures should somehow reflect the overall objectives of the control strategy that is being tuned. What is essentially obligatory, what is desirable and what is the behavior that one wishes to avoid?

Sensible parameter adjustment requires a model describing the dependency between the parameters and the quality measures. Because no physical model is assumed to be available, the model has to be based on the observed data. Therefore, a representative sample of parameter value combinations and the corresponding quality measure values are required.

A practical model type can be found by examining the data and its distribution more closely and by making a couple of quite realistic assumptions. First, the data is assumed to be *unimodal*, which means that it comes from a single multivariate Gaussian distribution. This assumption is motivated by the *Central limit theorem* stating that if a number of independent variables are added together the resulting distribution approximates Gaussian, no matter what the original distribution of the

variables was. It turns out that the Gaussianity assumption is justifiable also in practice (see Chapter 6.2.1). Because of the Gaussianity assumption the dependency between the variables θ and q is *linear* in the maximum likelihood sense /33/. However, it must be noted that the linear models can be applied only locally.

The unimodality assumption of the data requires the quality measures to be defined rather carefully: Their values should be smooth functions of the parameters over the whole θ axis (as was discussed in Chapter 2.4.1). When there are no abrupt changes in the values of the quality measures, the data sample tends to be more or less Gaussian (due to the Central limit theorem), and it is well justified to use local linear models for approximating the dependency $q = f(\theta)$.

After these assumptions there exist a great number of powerful tools available for modeling, such as ordinary linear algebra and multivariate statistical methods. The large enough data set that is required for the modeling can be obtained with Markov Chain Monte Carlo (MCMC) simulation. This means that the values of the qualifiers θ are varied randomly around some point and the corresponding quality measure values q are recorded. Let us assume that modeling involves n qualifiers and m quality measures that can be presented as vectors

$$\theta = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_n \end{pmatrix}, \quad q = \begin{pmatrix} q_1 \\ \vdots \\ q_m \end{pmatrix}. \tag{28}$$

After k samples of the data points are gathered (in k local iterations) they can be expressed in a matrix form

$$\Theta = \begin{pmatrix} \theta^T(1) \\ \vdots \\ \theta^T(k) \end{pmatrix}, \quad Q = \begin{pmatrix} q^T(1) \\ \vdots \\ q^T(k) \end{pmatrix}. \tag{29}$$

From now on, it is assumed that the required preprocessing of data is always taken care of without special notice. This means that the sets of data samples are locally centered and scaled to unit variance (see, e.g., /20/ for more details).

Based on a training data set a linear model F can be estimated such that

$$q = F^T \cdot \theta, \tag{30}$$

in which F^T is the mapping matrix from n-dimensional input space to m-dimensional output space

$$F^T : R^n \rightarrow R^m. \tag{31}$$

The same mapping for the data set of k samples in the matrix form can be presented as

$$Q = \Theta \cdot F. \tag{32}$$

Appendix B presents a short review over conventional and some more recent methods for finding the mapping matrix F. The review is based on /20/. Now, the matrix F

determines the dependency of the quality measures q and the parameters θ, i.e., the information that is needed to improve the performance by means of parameter tuning.

The separate quality measures can be aggregated into a single optimization cost criteria J, which, e.g., could be formulated as

$$\begin{aligned} J &= w^T q \\ &= \sum_{i=1}^{m} w_i \cdot q_i , \end{aligned} \tag{33}$$

where w is a $m \times 1$ weighting vector (see Chapter 6.4). If, e.g., the elements of w are set to unity the above cost criterion emphasizes the different quality measures equally in the optimization.

However, the estimated linear model (30) is meaningful only in the neighborhood of current parameter values, hereafter referred to as the nominal parameters $\bar{\bar{\theta}}$. This means that the optimal solution

$$\theta^* = \arg\min_{\theta} J\left(q(\theta)\right) \tag{34}$$

cannot be found with a single calculation and one has to accept the approach of taking short update steps towards the optimum. One has to apply iterative optimization approaches, e.g., gradient descent algorithm. The gradient, which indicates the direction of the maximal growth of the cost function J, is obtained by differentiating the equation (33), i.e.,

$$\frac{dJ}{d\theta} = \frac{d}{d\theta}\left(w^T F^T \theta\right) = Fw . \tag{35}$$

Thus, if it is assumed that the objective is to minimize the values of the quality measures, the parameters are updated to negative gradient direction according to

$$\begin{aligned} \bar{\bar{\theta}}(K+1) &= \bar{\bar{\theta}}(K) - \gamma \cdot \frac{dJ}{d\theta} \\ &= \bar{\bar{\theta}}(K) - \gamma \cdot F(K) w. \end{aligned} \tag{36}$$

Above, K is the global iteration step index and γ is the length of the parameter update step. The notation $F(K)$ emphasizes the fact that the matrix F is estimated over and over again, in every global iteration step. Also the values of γ and w can vary during the optimization procedure, e.g., the update step can be shortened as the optimum is approached. (Note that if the objective was to maximize the values of the quality measures, the minus sign in the parameter update formula (36) should be changed to a plus sign).

By using a short enough update step γ and large enough data set size k the algorithm becomes robust against random variations in the time series signals. In this context the word robust stands for the fact that the iterative algorithm changes the parameters inevitably towards the desired direction in the long run if only such a direction can be found in the parameter space.

4.2.1 Summary

The optimization procedure consists of *K* global iteration steps (see Figure 7). Through these steps the values of the parameters are gradually tuned towards their optimal values. Each step consists of a local iteration process, i.e., *k* simulations are run with slightly varied parameter values and the corresponding quality measures are calculated. A local linear model is estimated from the data if the Gaussianity assumption is not violated. The subsequent parameter update is based on the calculation of the gradient of the cost criterion. Global iteration steps are taken until the performance of the system meets its objectives or as long as significant improvements on performance can be observed.

In Figure 8 the same parameter optimization procedure is presented from the simulation point of view. Here, the simulation run is further split up into distinct *simulation events*. These events are either process or operator events that are of special interest as the values of the quality measures are calculated.

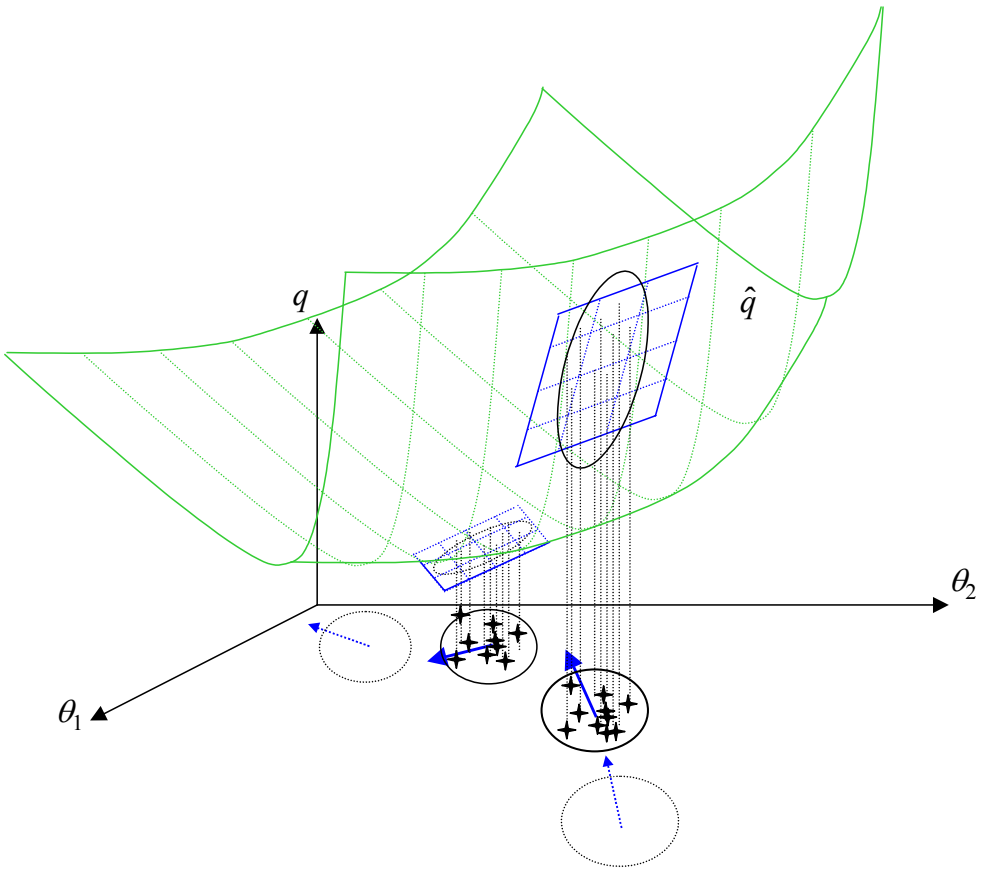

Figure 7. The optimization procedure consists of *K* successive global iteration steps (in this figure, *K* = 4). During each step, *k* data points are produced in local iteration. The data is used for modeling the local interdependence between the parameters θ and the quality measures *q*.

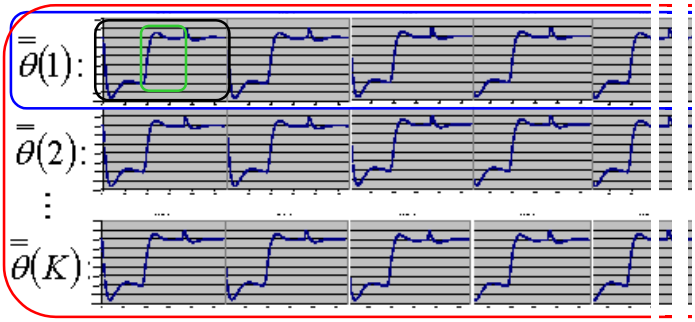

Figure 8. The tuning procedure (circled with red) consists of *K* global iteration steps (blue). A global iteration step consists of *k* local iteration steps (black), i.e., simulation runs. During a simulation, certain amount of successive simulation events (green) is run.

4.3 Alternative applications

In the previous chapter it was assumed that the IRT method was applied during the plant commissioning (or commissioning of the revised automation system). In that case the aim is to find a set of parameters that would result in a satisfactory performance, at least in the beginning. One would not have to start the plant operation with the time-consuming controller tuning task, but the tuning system could provide the user a good starting point. Next, some other examples of applying the IRT are discussed.

4.3.1 Adaptive control approach

As the operation of a plant has settled to a stationary condition, one might want to enhance the performance of the system by re-tuning the controllers. Now the required information, i.e., the qualifier – quality measure data, can be obtained directly from the process, and the tuning results obtained with the simulator can be further improved. The values of the parameters θ can be varied quietly around the nominal parameters $\bar{\bar{\theta}}$ and the performance with each combination is evaluated and recorded. After a sufficient amount of data is gathered, one can use the above-presented update paradigm to move slowly towards the optimal performance.

The biggest problem with the above approach is that the stability of the system cannot be guaranteed. Thus, this kind of automated parameter tuning system faces the same inconveniences as the conventional adaptive control schemes.

4.3.2 Gain scheduling approach

Besides the stability problem, the previous “adaptation” process is rather slow and it cannot follow any abrupt changes in the operation conditions. Thus, the idea of the gain scheduling presented in Chapter 3.3.1 could be applied here. Two distinctive control parameter sets, $\bar{\bar{\theta}}_A$ and $\bar{\bar{\theta}}_B$, e.g., appropriate for producing products A and B, can be found beforehand by using simulation. Then, by using a suitable scheduling variable, all control parameters can be changed simultaneously when necessary.

4.3.3 Multivariate ARS controller

Another interesting way of applying the IRT idea, originally presented in /23/, is illustrated in Figure 9. Industrial processes are typically controlled with several simple PID controllers. The three "tuning knobs" of the PID controller, i.e., gain, integration and derivation time, and their effect on the response of the controlled process are intuitively comprehensible. Similarly, it seems appealing to have a multivariable controller that would retain the clarity of PID and at the same time could control a large process entity with a number of controlled and manipulated variables. Further, a set of industrial controllers cannot typically be optimized once and for all, since the changing operating conditions require also changes to control parameter values.

As was discussed in the previous chapters, the dependency between a large amount of parameters and quality measures could be captured with statistical multivariate methods. Thus, by defining the "slopes" regarding quality measures like Accuracy, Robustness and Speed, the whole set of the PID controllers could be tuned all at a time by using an upper level ARS controller whenever the operating conditions change.

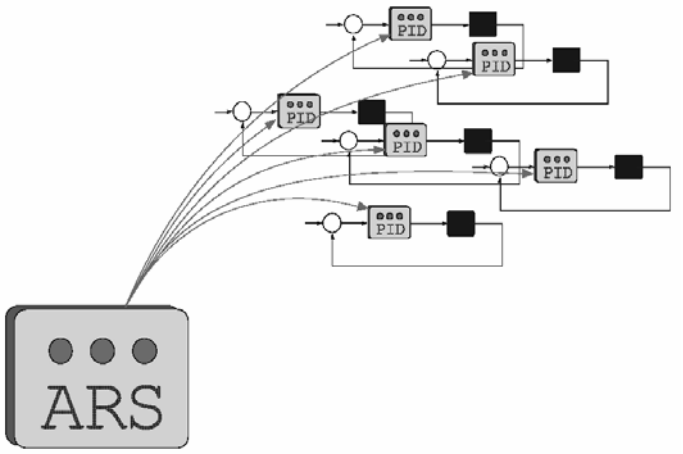

Figure 9. Multivariable ARS controller – simple as PID /23/.

4.3.4 Tuning of the simulation model

Also the simulator and its parameters can be tuned once measurement data from the actual process is available (see Figure 10). Now the objective of the tuning is in minimizing the difference between the actual and the simulated responses as the same input signals are applied. The simulator (or actually its parameters) is modified to better correspond the true behavior of the process. In other words, the optimization problem turns into a minimization task of the modeling error with respect to the model parameters.

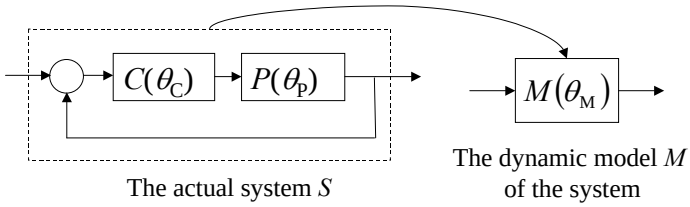

Figure 10. The data obtained from the actual system *S* can be used for the tuning of the model parameters θ_M.

4.3.5 Tuning of the process parameters

In the Introduction it was already mentioned that the object of the tuning does not necessarily have to be the parameters of the controllers. Just as well, the tuning could concentrate on the values of any other parameters, such as the setpoint values or the ramping parameters defining the operation point changes. In these cases either a dynamic simulator of the process or the actual process itself can be used for the generation of the data. The same assumptions as before hold also here regarding the properties of the parameters, i.e., they should be continuous.

4.4 Software application of the tuning method

This chapter outlines some essential issues concerning the software implementation of the IRT technique and the use of the resulting system. First, what kind of information and intervention the system requires from the user before and during the controller tuning procedure is discussed. After that, suitable techniques for presentation of the tuning results are considered.

In the following, a rather long list of initializing issues is presented and the requirements of the user interface of the tuning tool are considered. During the test case it was realized that the tuning system should have a deliberately designed user interface that would support the user in specifying the necessary initializations. However, the studies on the user interface requirements are still incomplete at this stage and therefore the following discussion remains bit summary.

4.4.1 Initializing

Before the actual tuning procedure can begin, the user is supposed to provide the tuning system with a certain amount of initial information. In Figure 11 the initial steps that precede the tuning procedure are presented (by using the same terminology as in Figure 8).

First of all, one has to specify the tuning objectives for the tuning software in terms of exact mathematical functions. It cannot be overemphasized that at this point the knowledge about the characteristic behavior of the process and the conventional disturbances and problems affecting it, is invaluable. It might be advantageous to start with the problems plaguing the existing or the conventional control structure, and think, what kind of improvements would be desirable and possible to achieve with controller tuning (i.e., without changing the process conditions or the control structure).

Regarding global iteration steps one has to specify the number of successive simulations k, i.e., the number of local iterations. The system should be able to propose a reasonable default value for k (more about topics that require further research is discussed in Chapter 7). A single simulation run consists of the events that the user specifies. These events are closely related to the quality measures. E.g., if the overshoot after a setpoint change is of particular interest it must be included in the set of simulated events. The user has to define also the order of the events in the simulation and what is the initial state from which the simulation begins. And at this stage also the variables, whose time series signals should be logged during the simulations, are specified. Further, every simulation event has to be defined in detail. This means specifying the process events, such as the disturbances, and the operator

interventions that are assumed to occur. What is also crucial is the length of the event, which naturally depends on the dynamics of the process.

Finally, the parameters that will be modified have to be selected. This can be done rather generously since the success of the tuning is not jeopardized even though the procedure would involve also some excessive parameters. However, it is essential that the user provides the system with the a priori information, e.g., about the known stability limits of the control parameters, magnitudes of the parameter values, etc. Further, one is assumed to give a somewhat sensible initial tuning for the controllers. At this point the magnitude of the parameter variation in the MCMC simulations is also specified.

Figure 11. The initializing steps that are required before the launch of the parameter tuning procedure.

4.4.2 Tuning procedure

In the following, the progress of the tuning is presented in a procedural form. The procedure consists of K global iterations. Every global iteration step begins with checking whether the stopping condition is reached. After that, the MCMC simulation is run around the prevailing nominal parameters and the values of the quality measures corresponding to the different parameter combinations are calculated. The Gaussianity of the data is ascertained and if the data violates severely the assumptions, a warning message is generated and the control of the tuning system is transferred to the user. Otherwise, the required preprocessing of data is performed and the dependency of the parameters and the quality measures is modeled and the update on the parameter values is calculated. The procedure is carried on until the stop condition is fulfilled.

```
while (stopping condition not fulfilled) {
      for (κ = 1 … k) {
            change the parameter values randomly;
            run the predefined simulation run;
            calculate the quality measures;
      }
      if (the data is not Gaussian) {
            notify the user and ask further instructions;
      }
      center and scale the data;
      construct the local parameter - quality measure model;
      determine the gradient direction;
      update values of the parameters;
}
```

4.4.3 Viewing the results

After the tuning is completed one is naturally interested in seeing the results. A flexible system should support several ways to view the achieved performance improvements. The time series plots of process variables are naturally an easy and comprehensible way to view the results. If the process responses before and after the tuning are plotted in the same figure, it is possible to judge their differences roughly. Another possibility is to inspect the plots of the cost function, the quality measure and the parameter values as functions of iteration step index. From these plots it can be concluded whether it pays to continue the tuning or has the optimization already converged. If one is interested in proving the enhancement of performance in a more formal way, statistical testing can be applied (See Appendix A).

5 POWER PLANT CASE STUDY

In this chapter the simulation model that was applied in the development of the tuning technique is presented. First, the power plant process and its Apros simulation model are presented. Then, the qualifiers and the quality measures are introduced. Finally, the practical arrangements of the test case are briefly introduced.

Apros is a professional simulation software that is designed for modeling and simulation of combustion and nuclear power plants, and pulp and paper mills. It provides large model libraries of process and automation components for construction of realistic industrial process models. For example, Apros has been used for constructing training simulators for several power plants using fossil fuels. Also, in many process analysis projects on power plants and paper mills, Apros has been applied successfully. Further information about Apros is available, e.g., in /1/.

5.1 Process description

The following short description of the process is not a comprehensive introduction to operation of a power plant. The actual aim, instead of introducing the process in detail, is more like showing that the applied process model was a rather complex system: Grasping the general view of a large and realistic simulation model of an industrial process is a demanding task. This is, however, the motivation behind the new tuning methodology. Since human mind is unable to comprehend the underlying interdependencies as the size of a system increases, advanced statistical multivariate methods are required to capture the emerging higher abstraction level concepts.

In the simulations a model of an oil burning power plant was used. The model consists of boiler (Figure 12), turbine (Figure 13) and feed water (Figure 14) sections and the related controllers (Figure 15), altogether 9 PI and 3 PID control loops. These sections are interconnected along the arrows marked into the figures.

The modeled power plant process uses oil as fuel and the heat that the combustion gases contain is first employed in the superheaters (Figure 12), from where the gas flow continues to the reheaters (between high and low pressure turbines in Figure 13). The last remains of the heat energy are used for heating of the feed water in the economizer (in Figure 12).

The feed water circulation is completely closed in the model, i.e., water that leaves the feed water tank also finally returns to it. First, the feed water is pumped from the tank (in Figure 14) through two heat exchangers to the economizer and to the superheaters. From there it continues to the high and low pressure turbines. The remaining water returns to the feed water tank through the condenser. Also the amount of water that is vaporized in the turbine section is returned to the feed water tank.

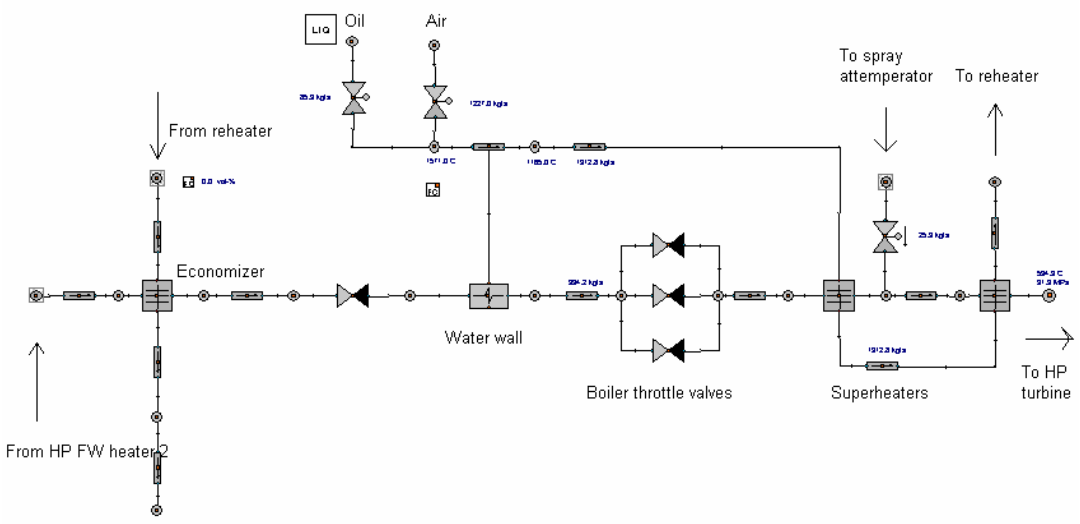

Figure 12. The boiler section of the power plant model.

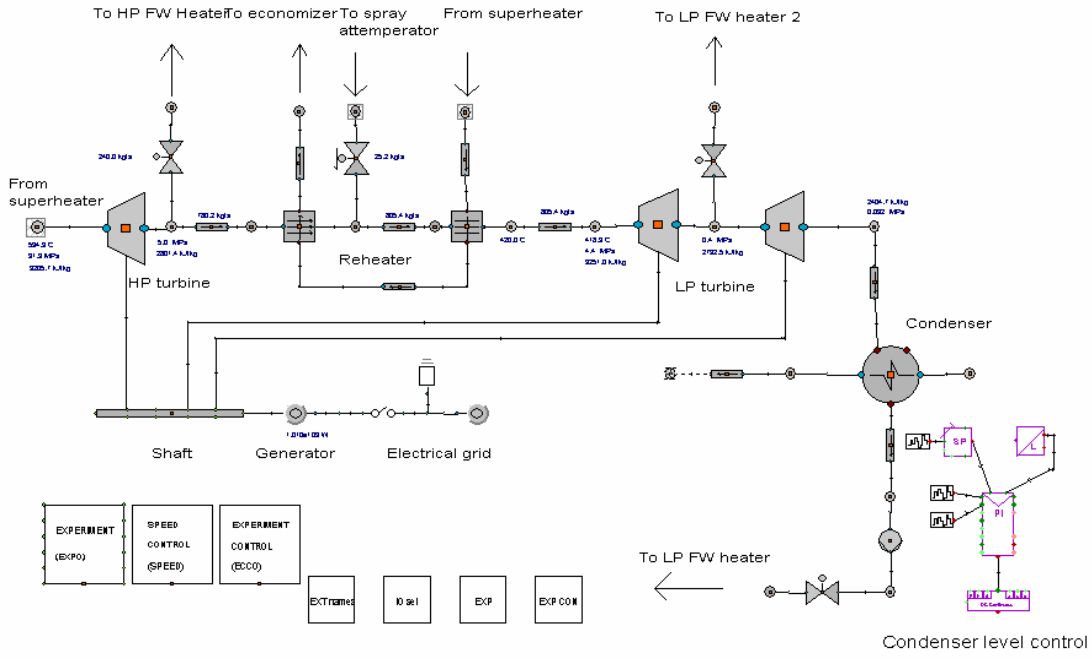

Figure 13. The turbine section of the power plant model.

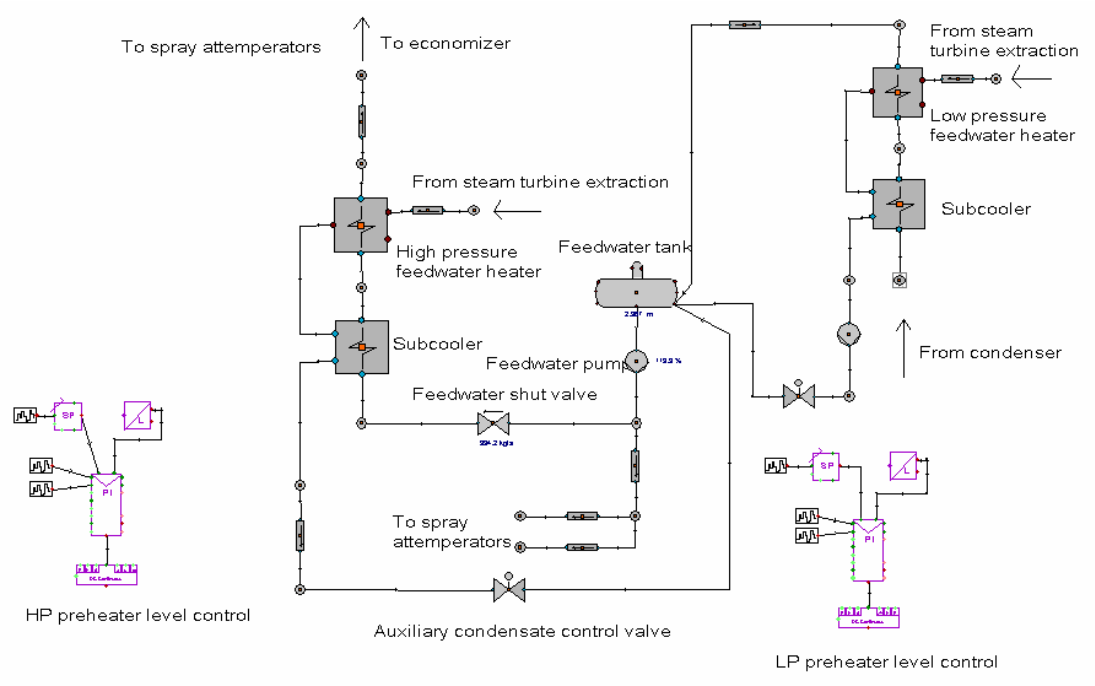

Figure 14. The feed water section of the power plant model.

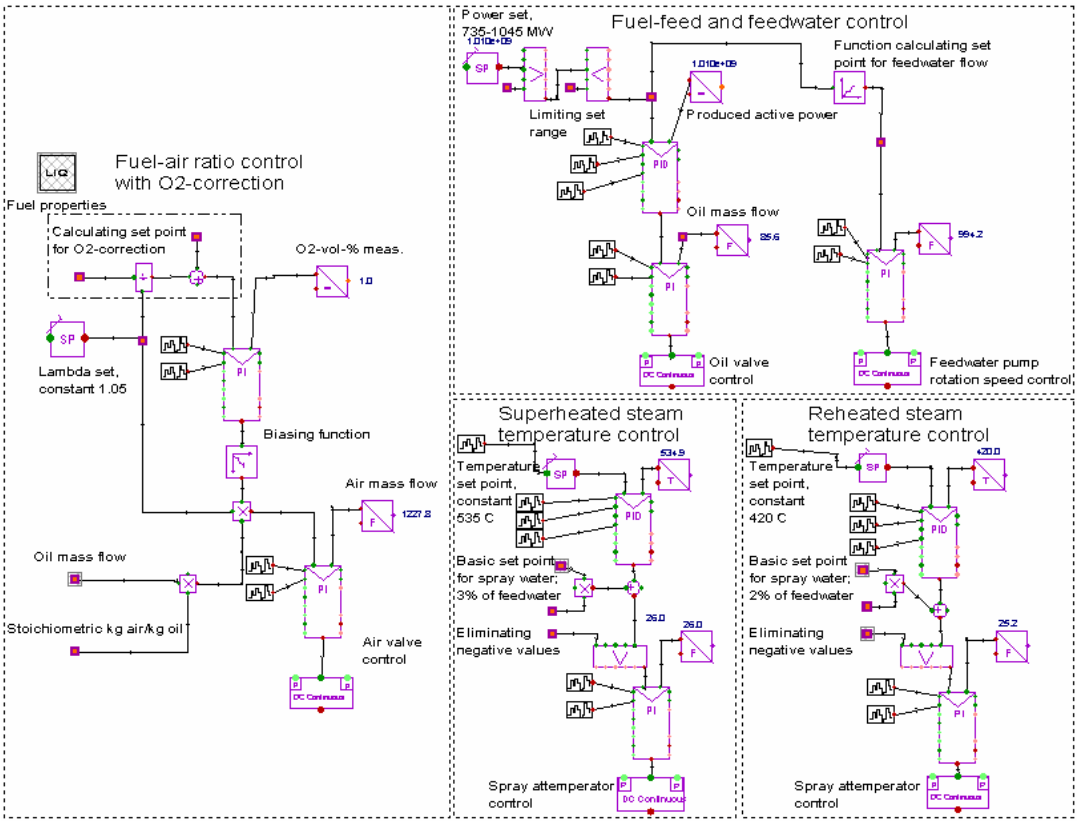

Figure 15. The primary control structures of the power plant model.

Figure 15 presents the most important control structures of the power plant. The produced active power is controlled by calculating a suitable setpoint for the fuel flow. The oil valve position is controlled by measuring the difference of the oil mass flow and the calculated setpoint value. The PI controller responsible for the control of the rotation speed of the feed water pump follows the setpoint signal that is calculated based on the currently produced active power. The air flow to the combustion and the oxygen correction are controlled by two PI controllers. The temperatures of the superheated and reheated steam flows are controlled with similar cascade control structures, in which the operator gives the setpoint values for the temperatures and the spray attemperator flows are used as manipulated variables. Additionally, the level control of the condensers is presented in Figure 13, and the level controllers of the high and low pressure heat exchangers in Figure 14.

5.2 Qualifiers and quality measures

Already Stanfelj *et al.* /34/ came to the conclusion that the top-level process performance assessment is always closely related to the process in question. The monitored variables and their objective behavior have to be considered separately to each control system under inspection. It is difficult to find objectives that would fit for many processes. Defining a mathematical expression for the concept of quality or for the desired performance even for an individual process is a challenging task whereby the assistance of domain area experts is required.

Thus, two researchers working at the VTT who had experience on power plants were interviewed. The goal was to form a set of practical quality measures that would describe the desirable operation of the power plant. The quality measures should be chosen such that they would characterize different essential aspects on the system performance in different operating situations. Some of the measures should be designed for assessment of the steady state situation whereas the others for evaluating the system response to load disturbances, noisy measurements and setpoint changes. Resulting from certain simplifications in the modeling, (e.g., the district heating network was not modeled), selecting realistic goals for the process was a bit complicated task. Thus, some liberties were taken: It was assumed that the plant was meant to answer the changing power demand, i.e., the responses of the produced active power to setpoint changes were considered essential.

In the case study the performance (characterized with 3 quality measures) of three controllers was tried to optimize simultaneously with respect to 7 parameters. The tuning was focused on a cascade control structure, where a PID controller calculates the setpoint for the PI controller responsible for the oil flow control. The master controller calculates the setpoint based on the difference between the measured and the desired produced active power. Also the parameters of the PI controller adjusting the rotation speed of the feed water pump were tuned. The quality measures that were optimized (minimized) were

- the settling time of the produced active power after a setpoint change (into 4 percent error margin),
- the overshoot of the produced active power after a setpoint change, and
- the variance of the produced active power after a pressure stroke in the combustion.

As the results are presented and discussed in the following chapter, the above quality measures will be referred to with the symbols q_1, q_2 and q_3, respectively.

5.3 Practical arrangements of the test case

Since a ready implemented platform for simulation assisted automation tuning was not available, the corresponding operations had to be constructed for the test case. The testing arrangements consisted of Apros and Matlab software. All automation functionality was implemented on Apros together with the process model and an external automation system was not involved at all.

The simulation model was managed with command queue files in which the simulated events were specified by using the Apros command language. One file contained the specification of k repeated simulation runs, i.e., one global iteration step. The time series signals of the variables were logged to text files. By applying the basic operations offered by the Apros component libraries, a new simulation block was constructed and attached to the model to take care of the random variation of the parameters around the prevailing nominal values. Matlab was used for calculating the quality measures out of the signals, modeling the dependency of the parameters and the quality measures, and finally for determining the new parameter values for the next iteration step. This routine was repeated in every global iteration step of the tuning procedure.

6 RESULTS

In this chapter the results of the test case are presented. First, improvements on the process performance after applying the novel tuning technique are assessed. Then, the validity of the assumptions presented in Chapter 4.2 is examined. The last part of this chapter introduces comparison of the different latent variable methods and some discussion on the aggregation of the quality measures into a sensible overall cost function.

6.1 Performance improvements

In Chapter 4.4.3 the presentation of the results to the user was shortly discussed and three possible approaches to view the success of the tuning procedure were introduced. In the following, the process performance with the parameters proposed by the tuning tool is first compared to the initial performance of the system. Then the plots of the cost function, the quality measure and the parameter values as functions of the global iteration step index are presented. Finally it will be shown how the significance of the performance improvement is ascertained with the statistical testing methods.

In Table 1 and Figure 16 the initial performance of the system is compared to the result after 18 global iteration steps. The improvement of the performance is obvious as the settling time, the overshoot and the effect of the pressure stroke in the combustion are assessed. There is no perceivable overshoot left in the step response after the parameter tuning and the time that is required to settle inside the error margins has decreased also notably. Also an improvement in the disturbance rejection ability can be seen: The resulting magnitude and the duration of the perturbation have both become smaller. Along with the minimization of the settling time the tuning has succeeded to shorten the rise time as well.

Although the goals of the tuning procedure were reached in the test case, one might still question whether these goals were reasonably selected in the first place. The achieved response to the setpoint change, e.g., is at first considerably rapid after which the process variable, however, drifts notably slow to its new setpoint. This clearly indicates inappropriate definition of the tuning objectives. As will be discussed in Chapter 6.4, the overshoot, e.g., should have been determined differently. In this sense, the presented methodology helps the experts to refine their intuition concerning the process and goals of performance. This can be seen as one of the major contributions of the Iterative Regression Tuning methodology.

Table 1. The values of the settling time T_S, the overshoot *OS*, the error signal variance var{*e*(*t*)} (from the signal values $t \in$ [360s…720s]) and the cost function *J* initially and after 18 global iteration steps.

K	T_S [s]	*OS* [%]	var{*e* (*t*)}	*J*
0	183	11,7	5,11E+05	3,000
18	57	0,2	3,41E+05	0,725

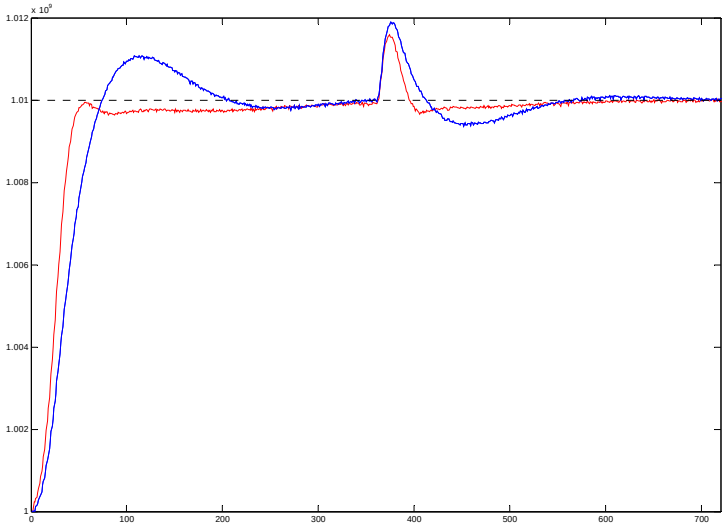

Figure 16. The responses of the produced active power to a setpoint change taking place at the time instant t = 0s and to a pressure stroke in the combustion (t = 360s) before (blue) and after (red) 18 global iteration steps.

In the following three figures (Figure 17, Figure 18 and Figure 19) the trends of the cost function, the parameter and the quality measure values are presented. It can be seen that minimization of the cost function has obviously succeeded and finally converged to a level from which any significant improvement of the performance by means of the control parameter tuning is difficult to find. The development of the parameter values during the tuning procedure of 75 iteration steps is shown in Figure 18. Each of them evolves rather consistently towards their new values in the beginning and a certain slowing-down in the update steps can be detected as the optimization procedure starts to converge. The Figure 19 presents the trends of the quality measures q_1, q_2 and q_3 during the optimization. The behavior of q_1 is heavily nonlinear due to its inconsiderate definition (see Chapter 6.2). Its influence on the values of the overall cost function is perceptible in the Figure 17. The Table 2 presents the values of the parameters at four different steps. The original parameter values were chosen to give a "good enough" performance, i.e., in their tuning any formal tuning method was not used.

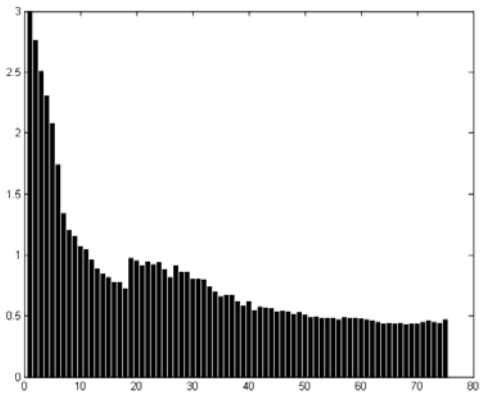

Figure 17. The development of the values of the overall cost function during 75 global iteration steps.

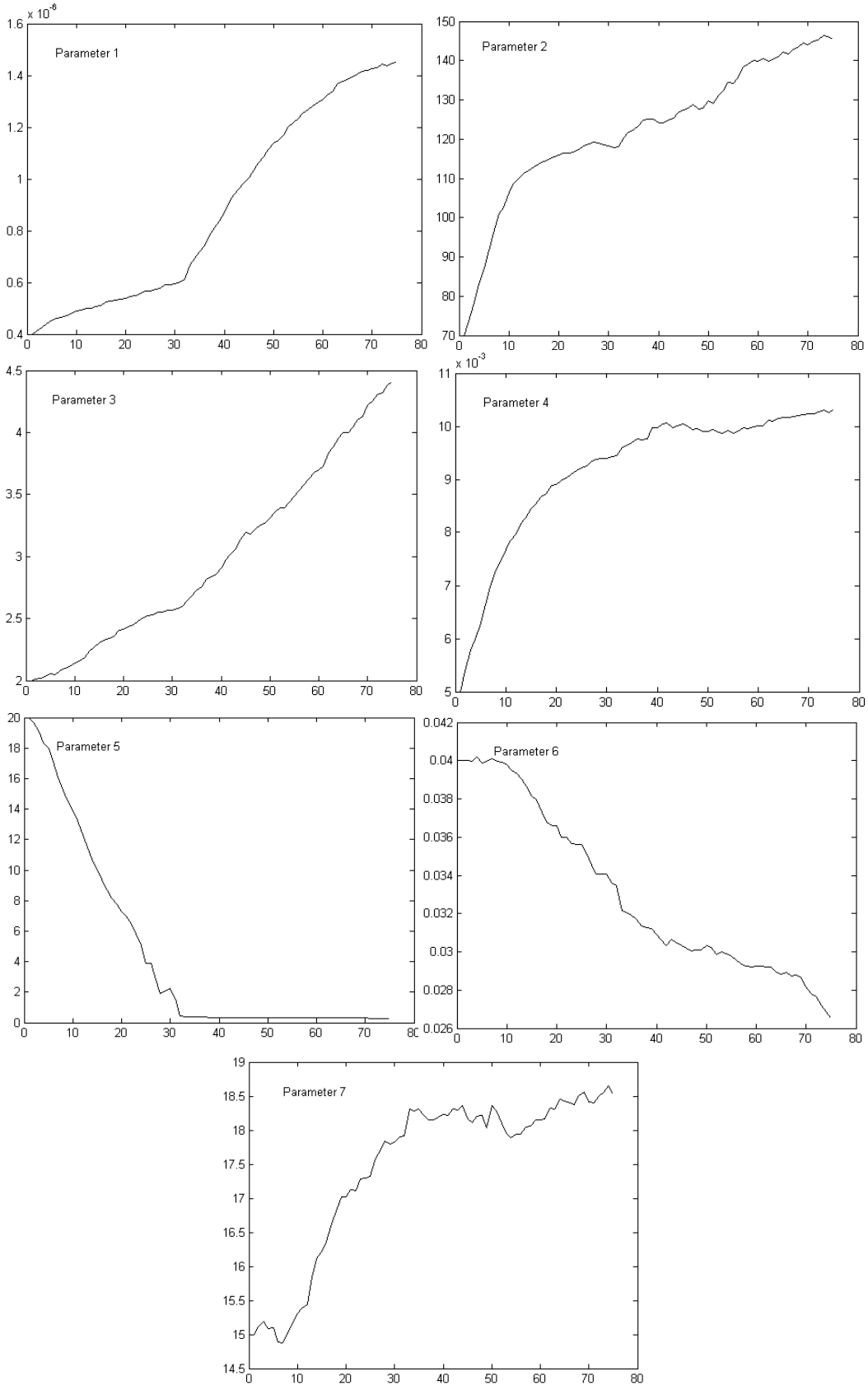

Figure 18. The trends of the parameter values, θ_1, θ_2, …, θ_7, during 75 global iteration steps.

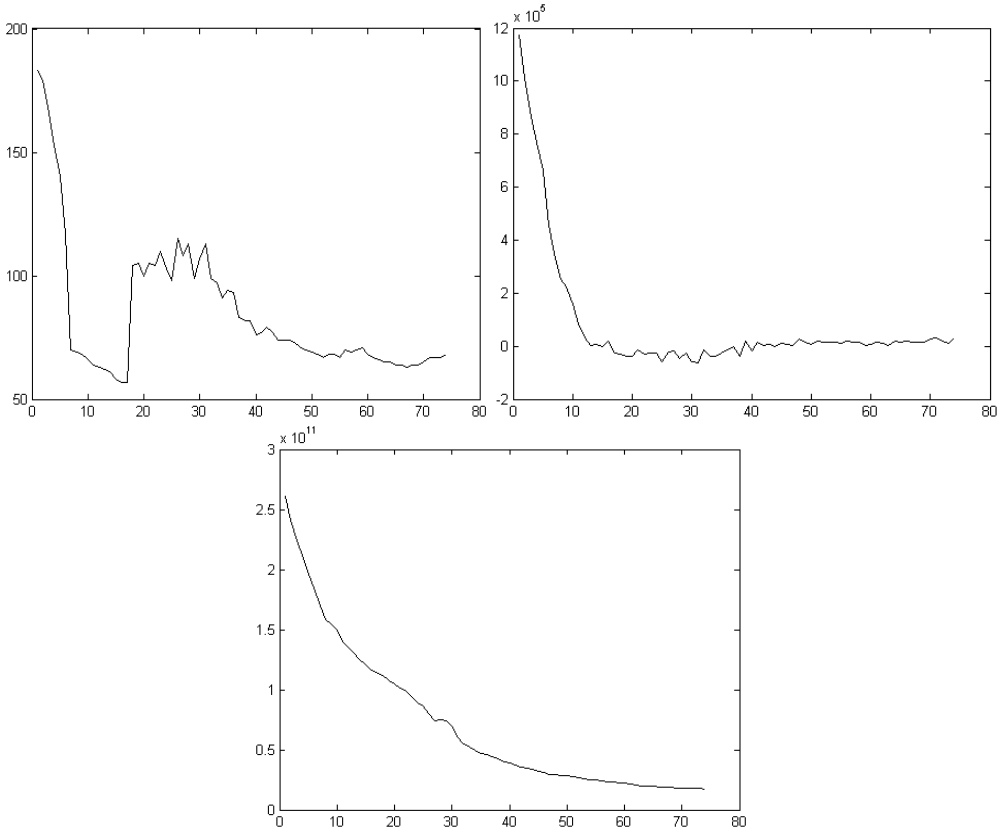

Figure 19. The development of the quality measure values, q_1, q_2 and q_3, during 75 global iteration steps.

Table 2. The nominal values of the parameters at *K*th global iteration step. θ_1, θ_4 and θ_6 are proportional gains, θ_2, θ_5 and θ_7 integration times and θ_3 derivation time of PI(D) controllers.

K	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7
0	4,00E-07	70,00	2,00	5,00E-03	20,00	4,00E-02	15,00
18	5,31E-07	114,89	2,36	8,74E-03	8,18	3,68E-02	16,81
50	1,11E-06	128,00	3,27	9,91E-03	0,32	3,01E-02	18,04
70	1,42E-06	144,40	4,13	1,02E-02	0,30	2,86E-02	18,56

The achieved performance improvement is obvious already based on the above examination. However, the change in performance can be proved to be significant also in the statistical sense and not only heuristically. By using the methodology presented in Chapter A.1.1 in Appendix A the test can be expressed as follows: The significance level for the test is set to $\alpha = 0.0005$ (i.e. the probability that the new achieved value of the quality measure q_i would be an observation from the initial distribution is 0.0005 at the maximum) and the hypotheses are formulated such that

$$\begin{aligned} H_0 &: q_i(K) = q_i(0) \\ H_1 &: q_i(K) < q_i(0) \end{aligned} \tag{37}$$

where $q_i(K)$ is the obtained value of the quality measure q_i after K global iteration steps and $q_i(0)$ is the initial (or conventional) value of the same quality measure. The initial values of q_1, q_2 and q_3 were estimated by calculating the mean value of hundred

observations, although the methodology assumes the null value to be known exactly. The estimates for K = 18 were calculated in the same way and the variances of the quality measure observations were estimated. The T-distributed normalized test statistic was defined as in equation (A5) resulting in the values

$$\hat{T}(q_1(18)) \approx 69$$
$$\hat{T}(q_2(18)) \approx 1520$$
$$\hat{T}(q_3(18)) \approx 1090.$$

Since with 99 degrees of freedom the probability of obtaining the values of the test statistic $T \geq 3.4$ is 0.0005, one can conclude that the observed values of $\hat{T}(q_1(18))$, $\hat{T}(q_2(18))$ and $\hat{T}(q_3(18))$ are far too rare to be from the initial distribution. Thus the null hypothesis can be rejected with a minimal probability of committing the Type 1 error (see Chapter A.1) and the alternative hypothesis stating that the process performance is better after 18 iterations can be accepted.

Although the statistical testing offers an established way of conducting conclusions, it is an ill-founded approach in this case. Since the variation in the obtained values of the quality measures is not originated from the actual process and its disturbance sources, but from the mathematical imprecision of the solver in the dynamic simulator, the resulting quality measure distribution does not correspond to any realistic situation. That is one explanation for the exceptionally high values of the test statistics, and thus the results of the hypothesis testing can be considered truthful only in the mathematical sense. However, as the tuning results are applied into practice, the above methodology offers an uncompromising way to compare the improved performance to the initial state.

6.2 Validity of the assumptions

In Chapter 4.2 some assumptions were made related to IRT method. First it was assumed that a linear model describes the dependency of θ and q with appropriate precision. For instance in /33/ it is shown that within a multinormal distribution the unknown variables (the dependent variables) can be modeled as linear functions of the known ones (the regressors). Here, the Gaussianity of the data is used as justification for linearity.

It has to be assumed that the data is unimodal, i.e., it forms a single distribution rather than separate clusters. In the following, it will be discussed how to define the quality measures so that their values conflict the Gaussianity and unimodality assumptions as little as possible (some bad examples of quality measure definitions are given to depict the possible consequences). However, it can be claimed that the above assumptions are quite realistic in practice assuming that the quality measures are defined carefully.

6.2.1 Unimodality

The definitions of the three quality measures q_1, q_2 and q_3 were presented in Chapter 5.2, and at first glance they seem to be reasonably selected. However, two of these quality measures could have been determined more wisely. This points out how the conventional performance characteristics can be totally inappropriate in this context. Therefore, the objectives of the optimization must be defined really carefully since the

results can be only as good as the objectives set to the problem. (However, as presented in Chapter 6.1, notable improvements on the performance were achieved in the test case, despite the deficiencies.) Here, while discussing the properties of the data distribution, we will study the reasons behind the first one of these flaws whereas the other will be treated in Chapter 6.4.

Let us first examine the distributions of the quality measures. The histograms of the centered and scaled quality measures are presented in Figure 20. It is clear that the distribution of q_1 (the settling time) does not fulfill the assumed normality.

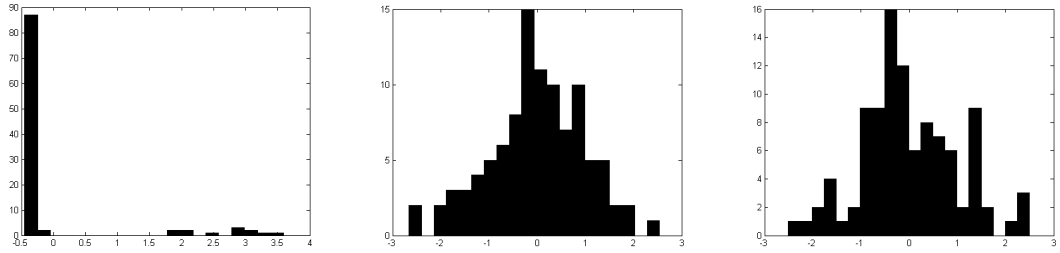

Figure 20. Histograms of the three quality measures, q_1, q_2 and q_3, k = 100.

If the dominating bar in the settling time histogram is examined a bit closer, it turns out that it actually hides a unimodal distribution, see Figure 21. The unsatisfactory nature of the original distribution is due to the inconsiderate definition of the quality measure. The problems result from the 4 percent tolerance boundaries that make the quality measure behave in this manner.

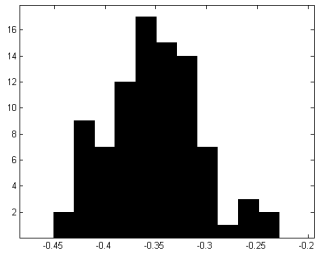

Figure 21. The histogram of the settling time values after the diverging data points are removed.

Let us study the consequences with an example: In Figure 22 six step responses of second order systems with different damping coefficients are presented. If the settling time is defined with tolerance boundaries, its values divide into separate clusters instead of forming a unimodal distribution. The quality measure becomes discontinuous and thus non-differentiable as illustrated in Figure 23. Obviously, this kind of behavior is not acceptable and it severely violates the assumptions made in the previous chapters. Thus, instead of using crisp boundaries, some kind of time weighted error signal integral of the form (9) could lead into better results, when the concept of settling time is of importance.

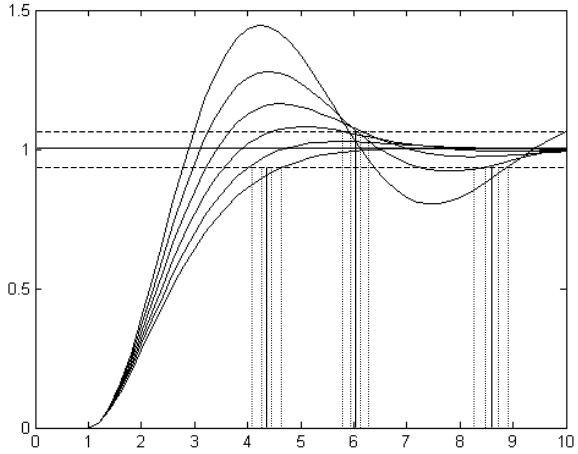

Figure 22. Six step responses of a second order system with different values of damping coefficient. The settling time distribution, if defined through tolerance boundaries, divides into three separate clusters.

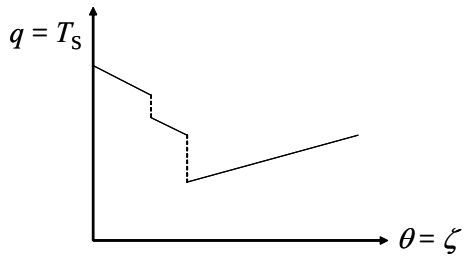

Figure 23. The settling time is a discontinuous function of the damping coefficient if it is defined through crisp tolerance boundaries.

Figure 24 presents distributions of two quality measure values from the same global iteration step. On the left, a distribution of a well behaving quality measure q_3 is shown (variation caused by a pressure stroke in the combustion), resulting in rather normally distributed values, and on the right, the distribution of q_1 (settling time) is shown. Three separate clusters are visible for the reason explained above. A linear model is not sufficient to explain this behavior, which can be seen also from Figure 25, where a similar data distribution is presented as a projection onto a plane.

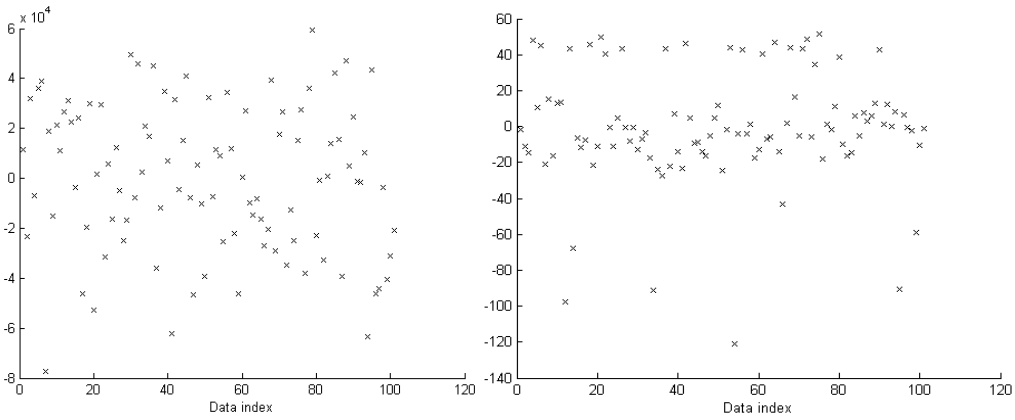

Figure 24. The values of quality measures that are approximately normally distributed (on the left) and divided into three clusters (right).

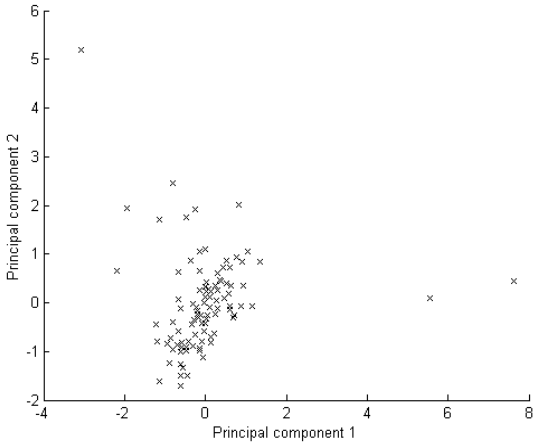

Figure 25. A quality measure distribution from an iteration step that involves some diverging data points. The data is projected to a plane spanned by its two major principal components.

6.2.2 Properties of the quality measures

In Chapter 2.4 the properties of a good quality measure q were considered and in Figure 5 some examples were presented. The optimization of smooth and monotonically decreasing quality measures, like q_3 and q_4 in Figure 5, would be rather straightforward. Unfortunately, the quality measures are more troublesome in practice. The difficulties that arose when a quality measure was not continuous over the whole θ axis were described above. Figure 26 illustrates another situation that is undesirable but still rather realistic. Many of the quality measures behave in this manner: The values of the function evolve rather moderately as the optimum is approached, but quite sharply after the optimal point the values of the quality measure explode. For instance, the settling time as a function of the proportional gain behaves in this manner: By increasing the controller gain, faster responses are obtained until at some point the system starts to oscillate, and, finally, the stability limit is reached, settling time becoming infinite.

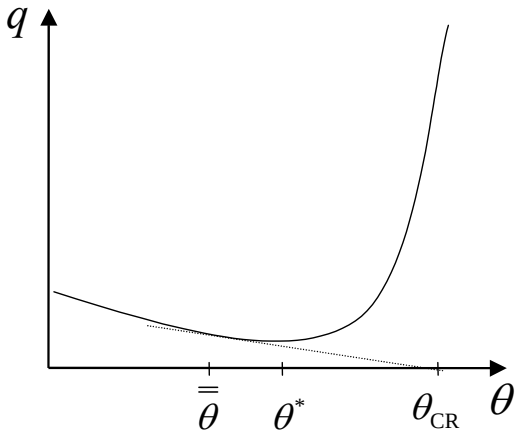

Figure 26. The optimal value of the quality measure q is achieved with the parameter value θ^*. As θ^* is passed by and θ_{CR} is approached the performance starts to degrade fast.

In practice, one faces the biggest problems when the varied parameter is close to some boundary, e.g., when the integration time has very small (but still positive) value. If the value of the parameter is accidentally changed to negative the process response "explodes" and so do the values of the most quality measures like the error signal integrals. The inappropriate parameter values may in some cases even cause divergence of the simulator, which will most probably terminate the whole tuning procedure.

To avoid these problems within the MCMC simulations, the range of the parameter variation should be considered. The magnitude of the parameter value variation can be bound to the parameter value itself (as was done in the test case) or it can be proportioned to the *feasible region* of the possible parameter values. In the latter case the boundaries of the feasible region should be known beforehand. This is, of course, impossible in many cases. Standard deviation of 5 percents (relative to the absolute value of the parameter) was used which turned out to be a practical choice.

The outlier detection and removal is an essential task when real process data are employed, e.g., for process modeling, because faulty measurements or missing values cause inaccuracy to the model. And when the data is obtained from a real plant it always contains some errors. But when the data is produced with a dynamic simulator all data points are valid (in theory) and should be used within the model estimation. However, the simulated data can contain rare values that cause some problems. The effects of these diverging observations on the success of the update step were studied in the test case. It did not seem to have any drastic consequences whether the out of line data were employed in the modeling or not. The resulting update step proposal was more or less the same in both cases, at least regarding the direction of the update. Typically, the direction of the proposed parameter update was not changed and only the magnitude of the update step was slightly affected.

6.2.3 Reliability of the parameter update

In the simulations the parameter updates were based solely on the estimated model. The estimation was assumed to be successful during every update step without any verification. The number of local iterations in an update step was fixed to an amount considered "certainly enough", and for that reason checking of the modeling precision was not considered necessary. However, as the previous discussion implies, check on the unimodality of the data distribution could prevent from the difficulties described above.

By examining the data distribution it can be predicted whether the estimation of the qualifier – quality measure model and therefore the calculated parameter update will be successful. If the unimodality assumption is severely violated at some global iteration step, there is no reason to believe that the resulting update based on the linear model would be towards optimal direction. Several methods are available for testing the normality of the distribution (see Appendix A).

If the tuning is performed with a simulator, a somewhat easier methodology can be used. The new parameters can simply be tested with the simulator without any risks of damaging the actual system. If the resulting performance is unsatisfactory or worse than before the update step, the modeling around the previous parameter values can be repeated.

6.3 On MVR models and parameter updates

The applicability of the PCR, PLS, CCR and CR methods (see Appendix B) was studied in the test case. In practice, during every iteration step these four different models were estimated such that the best latent structure (i.e., the number of latent basis vectors) was defined for every one of them. Here, the word “best” refers to the best estimation capability when an independent validation data was used. The evaluation of the estimation capability was done by comparing the mean estimation errors per data point. Being precise, the best latent structure was defined for 14 models on every iteration step since the CR model was estimated with eleven different values of the parameter α, i.e., $\alpha \in [0, 0.1, 0.2, \ldots, 1]$.

6.3.1 Comparison of the MVR techniques

Generally, it can be claimed that any major differences in the applicability of the different latent variable regression methods were not perceived. Only when the estimation capabilities of the different models were evaluated by comparing the mean estimation errors per data point, the CCR and CR seemed to outperform the PCR and PLS methods. It should be kept in mind, however, that it requires more calculation to define the best CR model than any of the others (about ten times more in this case), since it has one parameter more than the competing methods. The modeling has to be done separately for every value of the parameter α and thus the computational effort increases.

The figures below (Figure 27 and Figure 28) depict how accurately the four regression methods are able to model the slope of the settling time on the first and on the 20th global iteration step. The observations are arranged in the order of magnitude with respect to the true perceived values. As can be seen, the estimation of the descent direction from the validation data becomes much more difficult when the data set used for the modeling consists of separate clusters rather than a unimodal distribution.

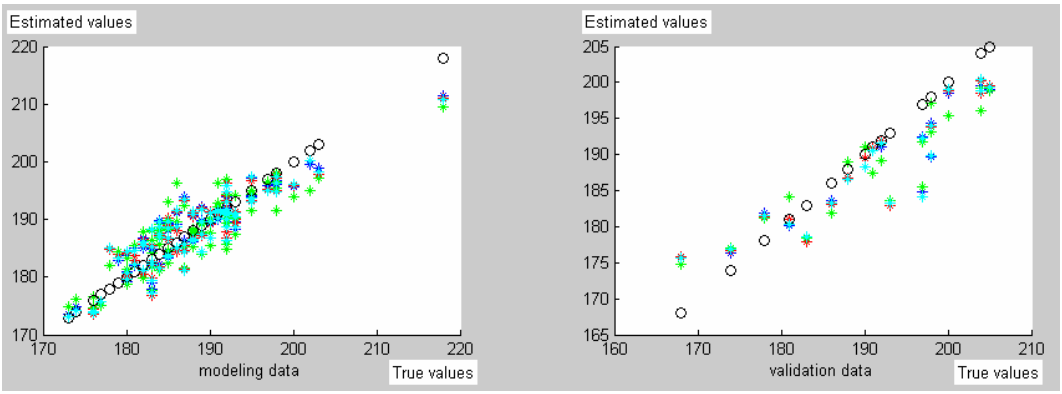

Figure 27. Settling time values from the first iteration step. The PLS estimates are denoted with blue asterisks, PCR with red, CCR with green and CR estimates with cyan. Black circles denote the true obtained values of settling time.

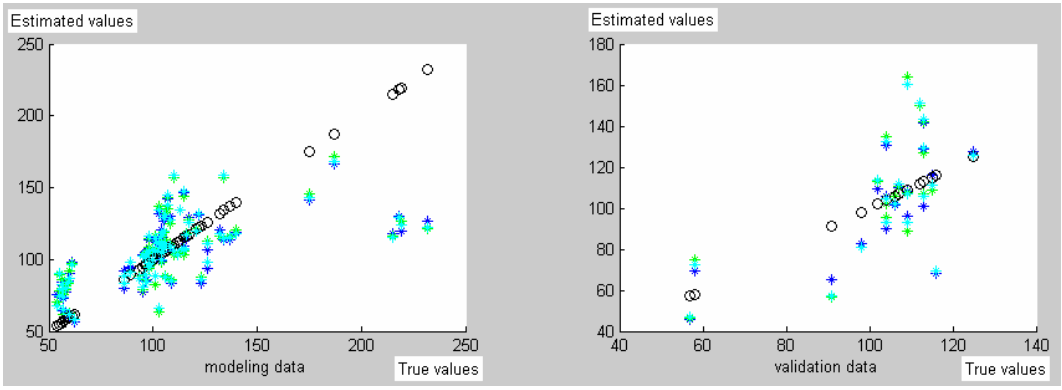

Figure 28. Settling time values from the 20th iteration step. The PLS estimates are denoted with blue asterisks, PCR with red, CCR with green and CR estimates with cyan. Black circles denote the true obtained values of settling time.

6.3.2 Observations on parameter convergence

First of all, in most of the cases every MVR method resulted in a more or less identical parameter update proposal. Especially the direction of the update was usually common to the different methods and differences were perceivable only in the length of the update steps.

Figure 29 presents the convergence of the proposed parameter update for θ_1 using different MVR techniques in four different global iteration steps. Note that the vertical axes are not in the same scale. During the tuning procedure the parameter was gradually changed from $\theta_1(1) = 4.00 \cdot 10^{-7}$ to $\theta_1(70) = 1.42 \cdot 10^{-6}$ (see Table 2), i.e., the desired update direction is increasing. As can be seen, the necessary amount of data points, k, varies quite a lot. The largest data set (until the correct update direction is found with every MVR method) is required in the step $K = 20$. In that step $k \geq 40$ data points are necessary. The reason is most likely the clustered data as discussed above. On the contrary, in the step $K = 1$ the right update direction seems to be clear to all MVR methods already after 10 simulations. Further, as $K = 70$ the update procedure is nearly converged and the update with every method is rather conservative (see also Figure 18).

At first it sounds a bit strange that seven parameters could be adjusted in a sensible way with respect to three quality measures by using only about ten data points. The answer lies in the idea of the latent variables. Put heuristically, if the number of the input variables can be reduced for instance to three and the number of the outputs to two, one has to estimate only $3 \times 2 = 6$ parameters instead of the original amount ($7 \times 3 = 21$).

Since the parameter update was calculated based on the formula (36) the length of the resulting update step is directly relative to the magnitude of the coefficients in the F matrix. Strong correlation between some qualifier – quality measure pair shows up as a large value of the corresponding element in F. If the elements on one row in F have different signs the objectives are contradictory with respect to the corresponding parameter. In such case defining an unambiguous update turns into a troublesome task whereby the decision making has to be involved. Smaller coefficients in F are indications of weaker correlation between qualifier – quality measure pairs. If some parameter does not have any correlation between the quality measures all elements in

the corresponding row in F matrix are near zero (at least their expected value in the long run is zero). Thus, including extra parameters to the tuning procedure has no unfavorable effects either on the success of the tuning or on the values of these parameters.

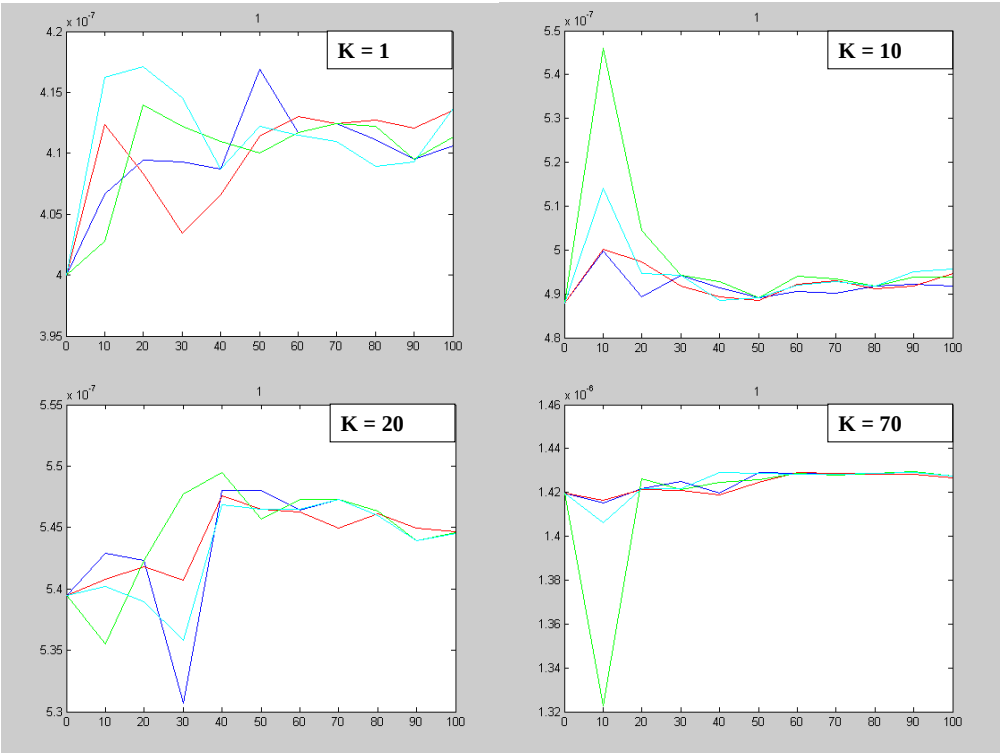

Figure 29. The convergence of the parameter update for θ_1 on global iteration steps K = 1, 10, 20 and 70 subject to the number of the data points. (Red line is the PCR, blue PLS, green CCR and cyan CR proposal.)

6.4 On optimality and multiple objectives

In the simulations the overall cost function to be minimized was defined as a weighted sum of the quality measures

$$J(K) = \sum_{i=1}^{m} \left(w_i \cdot \frac{q_i(K)}{q_i(0)} \right), \tag{38}$$

in which K is the index for the global iteration step and w_i is a weight for the quality measure q_i. The initial values of the quality measures are scaled to unity to make the progress of the tuning procedure easier to follow. If the update is performed according to (36), in which the mapping matrix F is estimated from centered and scaled data, all the quality measures are automatically equally weighted in the optimization despite their original magnitudes. However, comparing the importance of different quality measures (even if their values have been scaled) can be somewhat arbitrary. The decision making between multiple objectives that is required is discussed in more details, e.g., in /31/.

In Chapter 6.2.1 it was implied that in addition to the badly defined q_1 (the settling time) also one of the remaining two quality measures could have been determined

more wisely. This refers to the minimization of the overshoot after the stepwise setpoint change. The minimization was determined by the expression

$$OS = \max_t\left(y(t)\right) - r \tag{39}$$

rather than, e.g., by

$$OS = \left|\max_t\left(y(t)\right) - r\right|. \tag{40}$$

Above, r refers to the reference signal. The difference between the two minimization tasks is obvious: The first one tries to "push" the response signal $y(t)$ below the reference signal r all along the inspection period whereas the second expression minimizes the deviation of $y(t)$ from r. The definition (39) would be applicable if also an opposite minimization goal was defined. Thus, the best possible optimization results were not obtained although evident improvements in performance were reached, as discussed in Chapter 6.1.

The problem of incommensurable objectives is one of those arising when multiobjective optimization is considered. The above described method for mastering the multiple objectives of the optimization, that is an example of *scalarization* attempt, is extremely simplified. Scalarization means converting a multiobjective problem into single objective form. Adding the values of the optimized functions together gives a practical although slightly dubious way to manage several targets of interest with a single scalar valued function. In a way, the weighting of the quality measures in (38) represents *decision making* between the objectives: Preferences and further knowledge about the phenomena being optimized can be used to weight the relevance of the quality measures. More general and better justifiable methods for multiobjective optimization can be found from the literature, e.g., /31,35/.

The multiobjective optimization can be performed without the scalarization such that every single quality measure is optimized at the same time. This results in a *Pareto optimal* solution, which means that improvement on any of the optimized attributes is impossible without deteriorating the others. The Pareto optimal solution may be different depending on the initial state of the optimized system. Formulating such multiobjective optimization problems is rather ambiguous. Too many conflicting objectives can prevent the optimization of the quality measures as a whole. The mathematical machinery is unable to find any direction from the regressor space that would result in better values of quality measures, if the objectives include contradictory goals. Thus, the objectives have to be stated carefully and certain amount of realism should be kept in mind (or one can always abandon the Pareto optimality and apply scalarization approaches).

It is yet another question, whether any of the Pareto optima is "good enough" or whether it pays to spend time on finding "the very best" Pareto optimal solution. It depends crucially on the application at hand. If the optimization is continued within the Pareto optimal solutions, one has to use decision making between separate objectives. In some cases this is unavoidable as the set of Pareto optimal solutions may enclose major parts of the parameter space (i.e., the objectives are heavily contradictory). In practice, the solution that gives the most desirable overall behavior might as well lie outside the set of Pareto optimal parameter combinations. This might be due to

imprecise mathematical formulation of the objectives and the stochastic nature of the phenomena or deficient objective specification.

7 CONCLUSIONS

In Chapters 2 and 3 several techniques for PPA and control parameter tuning were introduced. This naturally raises the question what is the motivation of developing yet another method. In the beginning of 1990's the computational power of computers used in process control was much lower than today. This fact resulted in industrial implementation of extremely economical algorithms only. This can be seen for instance in the hierarchical PPA approach proposed by Stanfelj *et al.* /34/ in which the problem of process monitoring is split in several levels. The overall process performance is monitored separately from the controller performance and controller tuning, which are considered only if a major deterioration of process performance is perceived. Due to the explosion of the computing capacity construction and simulation of large and detailed process models is not an issue anymore. Therefore one has the opportunity to overcome the earlier problems and implement just as demanding algorithms as desirable.

The influence of dozens of control parameters on the process performance is a good example of a large complex system. Typical (and only possible) solution has been over decades to split the problem into pieces of manageable size. This kind of approach inefficiently takes into account the interactions between the control loops as they are tuned separately. The same disadvantage appears also in the iterative tuning algorithms of the multi-loop control systems /10,30/. The statistical multivariate methods proposed in /21,22,23/ instead enable the direct tuning of the control parameters against the objective behavior of the process. Applying these methods makes new concepts, covering the whole process and its performance, *emerge* from the amount of conventional single loop control performance characterizations. On the higher abstraction level, terms like accuracy, robustness and speed of the control system can be applied in the overall process performance context.

The Iterative Regression Tuning method can be applied also to other tuning tasks. For example, instead of control parameters, other continuous process parameters such as setpoints can be tuned to enhance the systems performance. Alternatively, after the plant is operating the actual measurement data from the process can be used to tune the parameters of the simulator so that it corresponds better with the reality.

Even though this report focused on the controller tuning during the commissioning of an automation system, the same tuning procedure could be used during the normal operation as well. E.g. changes in the production rate, raw material properties or other circumstances may require different tuning of the controllers. If the variations in the raw materials are known to be larger than normally for some reason or another, it might be advantageous to change the tuning of the controllers into more robust direction. Or if the plan of the near future operation consists of several setpoint changes (e.g. different grades are to be produced), the tracking ability of the overall control system is of particular interest. Now, if the different process performance objectives could be organized under a couple of intuitively understandable terms, the simultaneous tuning of all controllers could be as easy as the tuning of an easy-to-understand PID controller. Although more and more intelligent algorithms can be

implemented for controller tuning purposes it does not necessarily mean that the complexity of the tuning system would increase as well at the expense of the usability.

The future work on the development of the IRT method will consist of, e.g., defining more accurately the properties of quality measures that are best applicable in this context. A sort of function library approach could be a practical solution for the implementation of the tuning system. A collection of ready-implemented functions that are known to behave in an acceptable manner would decrease the possibility of facing severe difficulties caused by inconsiderate definition of quality measures. Similarly, a few different cost function types could be offered to user from which an appropriate one could be selected. For instance, one could decide whether to apply multiobjective optimization or decision making approach in the optimization.

The different alternative parameter update techniques should be studied in more detail, since the applied gradient descent algorithm is only one (simple) possibility to perform the iterative optimization. Also the number of required local iterations, which seems to have a certain connection to the number of significant latent variables in the data space, needs to be studied further.

Some sort of control on the reliability of the update step (i.e., the correctness of the proposed update direction) might by reasonable to implement as well, e.g., in form of normality testing. Whether the assumption of Gaussianity is severely violated, the system could at least provide the user with a warning of possible failure in the estimation of the parameter – quality measure model.

In the experiments, more or less identical “batches” were repeated in the local iterations to eliminate random effects and to speed up the parameter convergence. It can be questioned whether the obtained results can directly be applied in cases where the simulation runs cannot be controlled in the same way. However, there exist plenty of applications also for the presented simplified approach. For example possibilities of applying the methodology for optimizing the grade changes (the transient periods between operating points) in a paper machine are being studied.

REFERENCES

1. Anon., APROS - The Advanced Process Simulation Environment, VTT, [referred 20.4.2004]. www.vtt.fi/tuo/63/apros/index.htm.

2. Bristol, E.H., Pattern recognition: An alternative to parameter identification in adaptive control. Automatica 13(1977)2, p. 197-202. Ref. IEEE Transaction on Industrial Electronics 38(1991)6, p. 428-437.

3. Campi, M.C., Lecchini, A. & Savaresi, S.M., Virtual Reference Feedback Tuning (VRFT): a new direct approach to the design of feedback controllers. Proceedings of the 39th IEEE Conference on Decision and Control. Sydney, Australia, December 12-15, 2000. p. 623-629.

4. Glad, T. & Ljung, L., Control Theory – Multivariable and Nonlinear Methods, Taylor & Francis, London, 2000, p. 467.

5. Gunnarsson, S., Collignon, V. & Rousseaux, O., Tuning of a decoupling controller for a 2 × 2 system using iterative feedback tuning. Control Engineering Practice 11(2003)9, p. 1035-1041.

6. Halmevaara, K., Iterative latent variable based tuning technique for multiparameter systems, Master's thesis. Helsinki University of Technology, Department of Forest Products Technology, Espoo, 2004, p. 109.

7. Hang, C.C., Relay feedback Auto-Tuning of Cascade Controllers. IEEE Transactions on Control Systems Technology 2(1994)1, p. 42-45.

8. Hang, C.C. & Sin, K.K., A Comparative Performance Study of PID Auto-tuners. Control Systems Magazine 11(1991)5, p.41-47.

9. Hang, C.C. & Sin, K.K., On-Line Auto Tuning of PID Controllers Based on the Cross-Correlation Technique. IEEE Transactions on Industrial Electronics 38(1991)6, p. 428-437.

10. Hang, C.C., Åström, K.J. & Wang, Q.C., Relay feedback auto-tuning of process controllers – a tutorial review. Journal of Process Control 12(2002)1, p. 143-162.

11. Harris, T.J., Assessment of control loop performance. Canadian Journal of Chemical Engineering 67(1989)5, p. 856-861. Ref. Control Engineering Practice 4(1996)9, p. 1297-1303.

12. Harris, T.J., Boudreau, F. & MacGregor, J.F., Performance Assessment of Multivariable Feedback Controllers. Automatica 32(1996)11, p. 1505-1518.

13. Harris, T.J., Seppala, C.T. & Desborough, L.D., A review of performance monitoring and assessment techniques for univariate and multivariate control systems. Journal of Process Control 9(1999)1, p. 1-17.

14. Harris, T.J., Seppala, C.T., Jofriet, P.J. & Surgenor, B.W., Plant-wide feedback control performance assessment using an expert-system framework. Control Engineering Practice 4(1996)9, p. 1297-1303.

15. Hjalmarsson, H. & Birkeland, T., Iterative feedback tuning of linear time-invariant MIMO systems. Proceedings of the 37th IEEE Conference on Decision and Control. Tampa, FL, December 16-18, 1998. p. 3893-3898.

16. Hjalmarsson, H., Gunnarsson, S. & Gevers, M., A Convergent Iterative Restricted Complexity Control Design Scheme, Proceedings of the 33rd IEEE Conference on Decision and Control. Orlando, FL, December 14-16, 1994. p. 1735-1740.

17. Huang, B. & Shah, S.L., Performance assessment of control loops, Springer Verlag, London, 1999, 255 p.

18. Huang, B., Shah, S.L. & Kwok, E.K., Good, Bad or Optimal? Performance Assessment of Multivariable Processes. Automatica 33(1997)6, p. 1175-1183.

19. Huang, B., Shah, S.L. & Miller, R., Feedforward Plus Feedback Controller Performance Assessment of MIMO Systems. IEEE Transactions on Control Systems Technology 8(2000)3, p. 580-587.

20. Hyötyniemi, H., Multivariate Regression – Techniques and Tools. Helsinki 2001, Helsinki University of Technology Control Engineering Laboratory, Report 125, 207 p.

21. Hyötyniemi, H., Towards New Languages for Systems Modeling. Proceedings of 42'nd Scandinavian Simulation Conference SIMS'02. Oulu, Finland, September 26-27, 2002.

22. Hyötyniemi, H., On Emergent Models and Optimization of Parameters. Proceedings of 42'nd Scandinavian Simulation Conference SIMS'02. Oulu, Finland, September 26-27, 2002.

23. Hyötyniemi, H., Emergence and Complex Systems – Towards New Practices for Industrial Automation?. Proceedings of Intelligent Processing and Manufacturing of Materials IPMM'03, Sendai, Japan, May 18 - 23, 2003. (CD-ROM format)

24. Hägglund, T., A control-loop performance monitor. Control Engineering Practice 3(1995)11, p. 1543-1551.s

25. Hägglund, T., Automatic detection of sluggish control loops. Control Engineering Practice 7(1999)12, p. 1505-1511.

26. Jämsä-Jounela, S-L., Poikonen, R., Vatanski, N. & Rantala, A., Evaluation of control performance: methods, monitoring tool and applications in a flotation plant. Minerals Engineering 16(2003)11, p. 1069-1074.

27. Kalivas, J. H., Basis sets for multivariate regression. Analytica Chimica Acta 428(2001)1, p. 31-40.

28. Lee, K.S. & Lee J.H., Iterative learning control-based batch process control technique for integrated control of end product properties and transient profiles of process variables. Journal of Process Control 13(2003)7, p. 607-621.

29. Lequin, O., Gevers, M., Mossberg, M., Bosmans, E. & Triest, L., Iterative feedback tuning of PID parameters: comparison with classical tuning rules. Control Engineering Practice 11(2003)9, p. 1023-1033.

30. Luyben, W.L., Simple Method for Tuning SISO Controllers in Multivariable Systems. Ind. Eng. Chem. Process Des. Dev. 25(1986)3, p. 654-660.

31. Miettinen, K.M., Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, Norwell MA, 1999, 298 p.

32. Milton, J.S. & Arnold, J.C., Introduction to probability and statistics, 3rd edition, McGraw-Hill, New York, 1995, 811 p.

33. Pindyck, R.S. & Rubinfeld, D.L., Econometric Models and Economic Forecasts, 3rd edition, McGraw-Hill, New York, 1991, 596 p.

34. Stanfelj, N., Marlin, T.E. & MacGregor, J.F., Monitoring and diagnosing process control performance: The single-loop case. Industrial & Engineering Chemistry Research 32(1993)2, p. 301-314.

35. Steuer, R.E., Multiple Criteria Optimization: Theory, Computation, and Application, 2. edition, Robert E. Krieger, Malabar FL, 1989, 546 p.

36. Thornhill, N.F. & Hägglund, T., Detection and diagnosis of oscillation in control loops. Control Engineering Practice 5(1997)10, p. 1343-1354.

37. Tyler, M.L. & Morari, M., Performance Monitoring of Control Systems using Likelihood Methods. Automatica 32(1996)8, p. 1145-1162.

38. Åström, K.J. & Hägglund, T., The future of PID control. Control Engineering Practice 9(2001)11, p. 1163-1175.

39. Åström, K.J. & Wittenmark, B., Adaptive control, 2. edition, Addison-Wesley, Reading MA, 1995, 574 p.

APPENDIX A: STATISTICAL TESTING

In this chapter an (extremely brief) overview of the statistical testing methodologies is presented. The introduction is based on /32/, where the subject is discussed in more details. The concept of Statistical Process Monitoring (SPM) is also introduced shortly. At the end of the chapter some ideas are proposed how to utilize the statistical testing methods in the context of the IRT method and how to test the normality of a data distribution.

First, let us define some terminology used in this chapter: A distribution of a random variable X and a sample of k observations, X_1, X_2, …, X_k, is studied. The true population parameter q can be approximated with a sample estimate $\hat{q}$ that is calculated based on the observations. The population parameter is assumed to describe some essential characteristic of the whole population or the underlying distribution from which the sample is drawn. For example, this parameter might describe the expected value of the random variable $q = \mathrm{E}\{X\}$ that can be approximated with the arithmetic mean of the sample $\hat{q} = \overline{X}$.

A.1 Hypothesis and significance testing

Normally statistical testing is directed to an estimation problem of an (unknown) population parameter q. The classical hypothesis testing involves always two competitive and contradictory hypotheses related to the value of the estimated population parameter $\hat{q}$. The first one of these, H_0, is the *null hypothesis* that is related to the *null value* of the population parameter q_0 whereas H_1 is the *alternative* or *research hypothesis*. The experimenter tries to untangle whether the value of the sample estimate $\hat{q}$ refers to H_0 or H_1.

A.1.1 Tests on the mean value

The hypotheses are usually defined such that H_0 is tried to reject and H_1 stands for something desirable. Further, the equality of q with some preconceived value (null value q_0) is attached with the null hypothesis. For example, if the testing is conducted on mean value of the sample, i.e., the *test statistic* is $\hat{q} = \overline{X}$, the hypotheses could be stated as

$$\begin{aligned} H_0 &: q = q_0 \\ H_1 &: q > q_0. \end{aligned} \tag{A1}$$

The goal of the experiment is to show that with a certain level of significance it can be stated that the observed sample mean $\hat{q}$ represents an expected value of q that is higher than its preconceived value q_0. The preceding form of the hypotheses is called right-tailed test and the test is called left-tailed if the research hypothesis is formulated as

$$H_1 : q < q_0 , \tag{A2}$$

or two-tailed if

$$H_1 : q \neq q_0 . \tag{A3}$$

It must be pointed out that also the estimate of the population parameter $\hat{q}$ that is used as a test statistic is a random variable and its value varies from one sample to another with a certain mean value and variance. In other words, its values are determined by some distribution. To be able to determine any probability levels concerning the values of the test statistic $\hat{q}$ the underlying distribution should be known, at least approximately. If the test statistic reaches a value that is considered significantly rare as the null hypothesis is assumed to hold, one can reject the null hypothesis H_0 in favor of H_1. On the other hand, if the observed value of the test statistic is common under the assumption of true null hypothesis, the rejection of the null hypothesis fails.

Hypothesis testing involves two possible errors that can be encountered. Type 1 error means that one incorrectly rejects the null hypothesis. Type 2 error occurs if the rejection of the null hypothesis is failed although the research hypothesis is true.

In hypothesis testing the values of the test statistic that will lead to rejection of the null hypothesis are set beforehand by fixing the size of the test α, e.g., $\alpha = 0.05$. It means that the probability of the observed value of the test statistic that is considered small enough to lead to rejection of the null hypothesis is fixed to α. The values of test statistic that are less probable than α form the *critical* or the *rejection region* for the test.

In significance testing it is studied what is the probability or the *P value* of the observed value of the test statistic, if it is drawn from the null distribution. Small *P* values suggest that the observations are probably from another distribution than the one determined by the null hypothesis, i.e., the null hypothesis should be rejected.

As it was already mentioned the distribution of the test statistic has to be known. If the random variable X is normally distributed with mean $q_0 = \mathrm{E}\{X\}$ and variance σ^2 and a random sample of size k is observed then the *normalized test statistic*

$$\frac{\overline{X} - \mathrm{E}\{X\}}{\sigma / \sqrt{k}} \sim N(0,1) \tag{A4}$$

is normally distributed with zero mean and unit variance. In above it is assumed that the variance of X is known which is not true in many cases. If the variance is approximated with the sample variance $\hat{\sigma}^2$ it can be shown that the respective normalized test statistic

$$\frac{\overline{X} - \mathrm{E}\{X\}}{\hat{\sigma} / \sqrt{k}} \sim T_{k-1} \tag{A5}$$

follows the T_{k-1} distribution, i.e., T distribution with k-1 degrees of freedom. If X is not normally distributed T tests should be performed only if the sample size is large, i.e., k

≥ 25. Then the probabilities of Type 1 and 2 errors are not significantly increased /32/. Tests based on the statistic (A5) are commonly called *T* tests.

A.1.2 Tests on the variance

Statistical tests on the variance can be performed with the same general form of the hypothesis testing as in (A1). The variance is compared to its preconceived value σ_0^2. Now the distribution of the test statistic is however different. When sampling from a normal distribution, the test statistic

$$\frac{(k-1)\hat{\sigma}^2}{\sigma_0^2} \sim \chi^2(k-1) \tag{A6}$$

is known to follow a chi-squared distribution with *k*-1 degrees of freedom. Check on the normality must be made and if the data turns out to be non-Gaussian the preceding testing method cannot be applied.

A.2 Testing for normality

Several methods are available for testing the normality of distributions. For smaller samples visual inspection methods are usually proposed in the literature. A couple of these graphical methods are presented, e.g., in /32/. For bigger data samples also more sophisticated methods have been developed.

One that is recommended for data sets, in which $k \geq 50$, is called *Chi-square test for normality* /32/. The goal of the test is to find out whether there is enough evidence that the tested sample of random variables, $X_1, X_2, \ldots, X_k$, is not from a normal distribution. This is done by means of hypothesis testing.

First, the real axis is divided into *N* mutually exclusive categories. The lower and the upper boundaries of the *i*th category are denoted here with $b_{i,\mathrm{L}}$ and $b_{i,\mathrm{U}}$, respectively. (The first and the last one are open-ended in practice). Then estimates for sample mean and variance are calculated based on the formulas

$$\hat{\mu} = \sum_{i=1}^{N} \frac{O_i M_i}{k} \tag{A7}$$

$$\hat{\sigma}^2 = \frac{\left(k\sum_{i=1}^{N} O_i M_i\right)^2 - \left(\sum_{i=1}^{N} O_i M_i\right)^2}{k(k-1)}, \tag{A8}$$

where O_i is the number of observations falling into *i*th category, M_i is the midpoint of the *i*th category and *k* is the sample size. After that, the probabilities p_i that an observation falls into category *i* are estimated. For example, for the *i*th category

$$\begin{aligned} \hat{p}_i &= P\left[b_{i,\mathrm{L}} \leq X \leq b_{i,\mathrm{U}} \mid X \text{ is normal}\right] \\ &= P\left[\frac{b_{i,\mathrm{L}} - \hat{\mu}}{\hat{\sigma}} \leq Z \leq \frac{b_{i,\mathrm{U}} - \hat{\mu}}{\hat{\sigma}}\right] \end{aligned} \tag{A9}$$

where Z is obtained by a normalization similar to (A4) and it is $N(0,1)$ distributed. Then, the expected number of observations falling into category i is estimated by

$$\hat{E}_i = k\hat{p}_i . \tag{A10}$$

Finally, the hypotheses

H_0 : The data is drawn from a normal distribution.

H_1 : The data is drawn from a distribution that is not normal.

are tested using the test statistic

$$\sum_{i=1}^{N} \frac{\left(O_i - \hat{E}_i\right)^2}{\hat{E}_i} \tag{A11}$$

that follows approximately the chi-square distribution with N - 3 degrees of freedom. If the probability to obtain the value of the test statistic or a bigger value, when sampling from a normal distribution, is considered too small (smaller than the beforehand chosen test size α), the null hypothesis is rejected and the distribution of the random variable is concluded to be non-Gaussian.

A.3 Statistical process monitoring

Classical statistical testing can be used in the monitoring and detection of process performance problems /13/, e.g., one can determine whether changes have occurred in some characteristic figures of the system in the long run. Such changes could be, for instance, increased steady state variance of a controlled variable or change in any performance index. This kind of approach is called Statistical Process Monitoring (SPM).

Tyler and Morari /37/ have proposed a method for detection of process performance deterioration based on statistical testing. In their approach the acceptable process performance is expressed as constraints on closed loop impulse response coefficients. These constraints are derived from the performance specifications, e.g., for settling time, decay rate and output variance. The performance is evaluated by choosing between two alternative hypotheses:

H_0 : The current performance satisfies the objectives.

H_1 : The current performance violates the objectives.

Tyler and Morari claim that by selecting properly the performance objectives, this testing method outperforms, e.g., the Harris index as a monitoring tool. The proposed method is claimed to be insensitive to irrelevant changes in the noise dynamics but at the same time it is capable of sensing changes in the system model under examination.

As in SPM one tries to detect the change of process performance to the undesired direction, the same methodology can be applied to detect the improvements on performance as well. If a control system is tuned and the performance is evaluated against quality measures q, statistical testing can be used to detect the favorable changes in the plant behavior.

For example, it is rather difficult to judge whether the variance of a signal has decreased along with the controller tuning or does it only seem to have done so. With significance testing the changes in the performance can be related to some probability level. For instance, one can suggest that with 95 % confidence the expected value of the variance has decreased from its initial value.

In practice the testing can be done by estimating the quality measure $\hat{q}$ k times from independent samples. If one can assume that the values of $\hat{q}(1), \hat{q}(2), ..., \hat{q}(k)$ are normally distributed, the tests introduced in Chapter A.1.1 can be applied directly.

APPENDIX B: MULTIVARIATE REGRESSION METHODS

In this chapter a few Multivariate Regression (MVR) methods for constructing linear models of MIMO systems are presented. The presentation follows rather closely the report written by Hyötyniemi /20/ concentrating on the issues that are relevant when applying these methods. Therefore, most of the proofs and derivations are omitted.

In the following chapters the necessary data preprocessing is always assumed without extra emphasis. This includes centering and scaling of the data to zero mean and unit variance. More about these practical steps preceding the actual modeling task can be read from /20/.

B.1 On multidimensional data and linear models

In this short chapter the notation used for the multidimensional data is introduced. In order to retain the brevity on symbols in this report, the input and output variables are denoted here with θ and q, respectively, rather than, e.g., with u and y or x and y as conventionally on the system engineering literature. This choice makes it possible to use unified notations throughout the chapters of the report.

Let us assume that a system has n input variables θ_i, $i = 1, \ldots, n,$ and m output variables q_j, $j = 1, \ldots, m$. θ and q without any subscripts are used as symbols for multidimensional input and output column vectors, respectively, such that

$$\theta = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_n \end{pmatrix} \quad q = \begin{pmatrix} q_1 \\ \vdots \\ q_m \end{pmatrix}. \tag{B1}$$

If k samples of inputs θ and corresponding outputs q are available, the data set can be expressed in a matrix form

$$\Theta = \begin{pmatrix} \theta^T(1) \\ \vdots \\ \theta^T(k) \end{pmatrix} \quad Q = \begin{pmatrix} q^T(1) \\ \vdots \\ q^T(k) \end{pmatrix}. \tag{B2}$$

Note that vector (and scalar) signals and variables are denoted with the lower case symbols whereas the upper case symbols denote a collection of k samples of these vector valued variables. Furthermore, it will be assumed that k is much higher than n or m.

The purpose of the modeling is to find a linear model F such that

$$q = F^T \cdot \theta, \tag{B3}$$

or for the data in matrix form,

$$Q = \Theta \cdot F \,. \tag{B4}$$

In the following, different techniques to find linear models for the MIMO systems are presented.

B.2 Multilinear regression

If a linear model structure between input and output variables is assumed the technique that is usually applied in the modeling is the Least Squares (LS) method proposed by Gauss already in the 1800's. On many applications it results in feasible models but as the number of variables grows, and especially if the data is not well conditioned in numerical sense, the method gets into troubles. Let us first study a MISO (multiple inputs, single output) model where only the *i*th output is considered. Naturally, the estimates based on a regression model never coincide perfectly with the true output values but there is a modeling error E_i present such that

$$Q_i = \Theta F_i + E_i \,. \tag{B5}$$

Q_i and F_i are the *i*th columns of the Q and F matrices and E_i is a column vector that contains k error terms corresponding to k observations of Q_i. Naturally for a good model F_i the errors E_i should be rather small. Thus, finding such parameters F_i that minimize the sum of the squared errors should lead to a sound model:

$$\sum_{\kappa=1}^{k} E_i^2(\kappa) = E_i^T E_i = \left(Q_i - \Theta F_i\right)^T \left(Q_i - \Theta F_i\right), \tag{B6}$$

in which $E_i(\kappa)$ denotes the modeling error of *i*th output on the κth data sample. By differentiating the equation (B6) with respect to model parameters F_i and setting the result equal to zero (vector), gives an equation whose solution is an extremum point of the sum of the squared errors:

$$\frac{d\left(E_i^T E_i\right)}{dF_i} = -2\Theta^T Q_i + 2\Theta^T \Theta F_i \equiv \mathbf{0} \,. \tag{B7}$$

For a second order polynomial the extremum is unique and thus it is a global extremum. And because the second derivative matrix,

$$\frac{d^2\left(E_i^T E_i\right)}{dF_i^2} = 2\Theta^T \Theta \geq 0 \,, \tag{B8}$$

is positively semidefinite, the extremum is a global minimum. Thus, solving the equation (B7) gives the optimal parameters F_i in the least squares sense:

$$F_i = \left(\Theta^T \Theta\right)^{-1} \Theta^T Q_i \,. \tag{B9}$$

If there are m output variables, the above expressions can be combined as

$$F = (F_1 \mid \cdots \mid F_m) = (\Theta^T\Theta)^{-1}\Theta^T(Q_1 \mid \cdots \mid Q_m) = (\Theta^T\Theta)^{-1}\Theta^T Q\,. \qquad \text{(B10)}$$

The obtained model F is called Multilinear Regression (MLR) model.

B.2.1 Problems and improvement ideas

In the MLR approach the ordinary assumptions relative to linear regression model should hold /33/. First of all, the relationship of the regressors and the dependent variables is assumed to be linear as in (B5). Secondly, the regressors should be non-stochastic. Furthermore, it is assumed that there exists no exact linear dependency between any of them. The error terms E_i in (B5) are assumed to have a zero expected value and a constant variance for all observations (i.e. homoscedasticity assumption). The errors of the outputs Q_i and Q_j at samples κ_1 and κ_2, $E_i(\kappa_1)$ and $E_j(\kappa_2)$, are also assumed to be statistically independent for all i, j, κ_1, κ_2 (also for the case $i = j$, if only $\kappa_1 \neq \kappa_2$).

In practice the above assumptions are violated more or less and so the MLR has some deficiencies. For instance the linear model type turns out to be the optimal solution only if the data is Gaussian. As MLR attempts to model all of the variation in the output variable direction, it accidentally models also some of the random variations in the dependent variable values. This causes troubles when the model is used for forecasting values of Q with an independent input variable set Θ. Another and maybe even more severe problem is the *collinearity* in the variables. E.g., if there are two nearly identical variables in the Θ matrix, one of the eigenvalues of the corresponding covariance matrix $1/k \cdot \Theta^T\Theta$ approaches zero, i.e., the matrix becomes rank deficient. Now the matrix inversion in equation (B10) is no longer defined. Even though one succeeded to calculate the inversion of a nearly singular covariance matrix, the model would become very sensitive to noise and quite useless for the estimation purposes.

In the following, some techniques are presented to overcome these inconveniences. These methods share the same fundamental idea of how to enhance the regression model and they are known as *statistical multivariate regression methods* or *multivariate subspace projection methods*. Let us study their philosophy a bit closer. It is a fairly realistic assumption that all measurement signals are more or less noisy and redundant (collinear) in practice, but on the other hand all of them carry some fresh information as well. If some of the variables are omitted in order to remove the redundancy, crumbs of information are also lost. And naturally one wants to get rid of the redundancy and noise only. The problem is solved if the data dimension can be reduced in a sophisticated way, i.e., an appropriate *subspace* can be found from the original data space.

Let us study an example: Figure 30 presents a collinear two-dimensional data set in its original coordinate system, i.e., using variables θ_1 and θ_2. Unit vectors θ_1° and θ_2° with the same directions as θ_1 and θ_2 constitute the so-called natural basis, Φ_θ, of the variable space. However as can be seen from the figure, most of the variation is explained by the variable ϕ_1, which is a linear combination of θ_1 and θ_2. If the variation in the direction of ϕ_2 is assumed to be nothing but noise, one is able to describe the relevant information using only one variable. Thus, Φ = (ϕ_1) can be selected as a basis vector set (in this case consisting of only one vector) that spans a (one-dimensional) subspace in the original two-dimensional space spanned by Φ_θ = (θ_1° |

θ_2°). Now, if the original data set is first projected onto this subspace, the linear mapping to output space can be done from Φ without any problems caused by collinearity, i.e., one can use the MLR approach for the regression. The modeling is divided into two linear mappings:

$$\Theta \xrightarrow{F^1} Z \xrightarrow{F^2} Q\,, \tag{B11}$$

where Z are the variables in the subspace spanned by Φ. In the following these new variables are called *latent variables*. In practice the mapping can be combined finally into a single calculation such that

$$Q = ZF^2 = \left(\Theta F^1\right)F^2 = \Theta F\,. \tag{B12}$$

Thus, to result in a satisfactory regression model F one has to find suitable basis vectors spanning the subspace Φ. This will be elaborated in the following chapters and several techniques will be presented. The properties of bases and linear mappings are discussed in more details, e.g., in /20/.

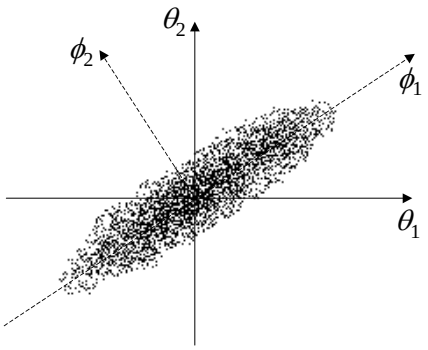

Figure 30. A data distribution in two dimensions: variables θ_1 and θ_2 are mutually correlated, i.e., collinear. The same data distribution can be expressed also as a function of ϕ_1 and ϕ_2.

B.3 Latent variable methods

In the following subchapters four different regression methods are presented. The overall idea of dimension reduction is common to all of them. Only the principles how this is obtained differ from one method to another. Here these methods are presented through the same eigenproblem approach although their customary definitions in the literature usually might look quite different.

B.3.1 Principal Component Regression

The Principal Component Regression (PCR) is based on Principal Component Analysis (PCA), which assumes that the information content in a data set is carried by the covariation. Therefore, in PCR the basis vectors spanning the subspace Φ are selected such that they explain the major part of the variation in the input data (like in the example presented in Chapter B.2.1). The purpose is to omit the directions of the data variation where the signal-to-noise ratio is smallest, and use the latent variables for regression purposes. This is motivated by the assumption that the uncorrelated noise is evenly distributed in every direction, whereas real underlying phenomena are

typically reflected in various original variables. E.g., in Figure 30 ϕ_1 points to the direction of the major variation whereas the direction of ϕ_2 is of minor significance. A reasonable selection for the subspace basis in this case would be $\Phi = \phi_1$, as discussed earlier.

Let us assume that ϕ_i is an arbitrary direction in the data space. If the input data Θ is projected onto this vector, i.e.,

$$Z_i = \Theta\phi_i, \tag{B13}$$

the variance of the projections (latent variables) is

$$\mathrm{var}\{Z_i\} = \frac{1}{k} Z_i^T Z_i = \frac{1}{k} \phi_i^T \Theta^T \Theta \phi_i. \tag{B14}$$

If the aim is to find the direction of the maximum variance, one ends up in a constrained optimization (maximization) problem with respect to ϕ_i:

$$\begin{cases} f(\phi_i) = \frac{1}{k} \phi_i^T \Theta^T \Theta \phi_i \\ g(\phi_i) = 1 - \phi_i^T \phi_i = 0. \end{cases} \tag{B15}$$

Above $f(\phi_i)$ is the objective function that is maximized with the restriction $g(\phi_i)$. If the length of the vector ϕ_i were not restricted the values of the function $f(\phi_i)$ would grow without limit. The optimization problem can be solved using the method of Lagrange multipliers resulting in a cost function

$$J(\phi_i) = f(\phi_i) - \lambda_i \cdot g(\phi_i) \tag{B16}$$

in which λ_i is a Lagrange multiplier. By differentiating the above cost function with respect to the parameter vector ϕ_i (mapping parameters from input data to latent variables), and setting it to zero,

$$\frac{dJ(\phi_i)}{d\phi_i} = \frac{d}{d\phi_i}\left(f(\phi_i) - \lambda_i \cdot g(\phi_i)\right) \equiv \mathbf{0}. \tag{B17}$$

By substituting (B15) in (B17), calculating the derivatives and reorganizing the terms the equation can be expressed as an eigenvalue problem

$$\frac{1}{k} \Theta^T \Theta \cdot \phi_i = \lambda_i \cdot \phi_i. \tag{B18}$$

The above equation is the eigenvalue formulation of the data covariance matrix and any of its eigenvectors ϕ_i corresponding to an eigenvalue λ_i solves the equation (B18). Thus, all the eigenvectors of the data covariance matrix represent extremum points for the original optimization problem (B15). For the $n \times n$ symmetric covariance matrix, there are n orthogonal eigenvectors (corresponding to real and non-negative eigenvalues, see /20/) which are all possible solutions to the above equation. However, it can be shown that maximum is only reached with the eigenvector corresponding to the largest eigenvalue.

The eigenvectors of the covariance matrix are called the principal components of the data. Choosing the most relevant principal components to the basis set Φ gives many advantageous properties to the regression model. The orthogonality of the eigenvectors assures uncorrelatedness of the latent variables. This means that the covariance matrix of the latent variables is always invertible if only the eigenvalues are not zero. Further, the eigenvectors of the covariance matrix can be extracted now one by one.

If all the *n* eigenvectors are included to the basis set Φ, all the information content of original Θ data can be expressed with the latent variables *Z*. However, in PCR it is appropriate to omit some of the minor principal components from the basis. If *N* most significant eigenvectors are included to the basis $\Phi_{\text{PCA}} = (\phi_1 \mid \ldots \mid \phi_N)$ the PCR model is implemented as

$$\begin{aligned} \hat{Q}_{\text{est}} &= \Theta_{\text{est}} F_{\text{PCR}} = \Theta_{\text{est}} F^1 F^2 \\ &= \Theta_{\text{est}} \cdot \Phi_{\text{PCA}} \left(\Phi_{\text{PCA}}^T \Theta^T \Theta \Phi_{\text{PCA}} \right)^{-1} \Phi_{\text{PCA}}^T \Theta^T Q, \end{aligned} \tag{B19}$$

where $\hat{Q}_{\text{est}}$ is the estimated output corresponding to input Θ_{est}.

B.3.2 Partial Least Squares

PCR tried to express the covariation of the multidimensional input data in lower dimensional subspace with latent variables. The attempt is understandable but instead of maximizing the variation of the input variables of the regression model, it sounds even more reasonable to maximize the correlation between input and output variables. This approach is taken in Partial Least Squares (PLS) regression to find the latent variables. Here, it should be emphasized that PLS is usually defined in an algorithmic form. However, in this context an eigenproblem approach is taken. It can be proved that only the most significant eigenvector directions found with the eigenproblem formulation coincide exactly with the results of the algorithmic form.

In the PLS the latent structure is searched not only from the input variable space but also from the output space. Thus the regression can be thought to consist of a three-step procedure:

$$\Theta \xrightarrow{F^1} Z^1 \xrightarrow{F^2} Z^2 \xrightarrow{F^3} Q\,. \tag{B20}$$

Above Z^1 and Z^2 are the variables in the input and output oriented subspaces, spanned by the bases $\Phi^1 = (\phi_1 \mid \ldots \mid \phi_N)$ and $\Phi^2 = (\varphi_1 \mid \ldots \mid \varphi_M)$, respectively. Maximization of the correlation of the latent variables Z^1 and Z^2 results similarly as in the PCA case in a constrained optimization problem, now with two constraint equations:

$$\begin{cases} f(\phi_i, \varphi_i) = \dfrac{1}{k} \cdot \phi_i^T \Theta^T \cdot Q \varphi_i \\ g_1(\phi_i) = 1 - \phi_i^T \phi_i = 0 \\ g_2(\varphi_i) = 1 - \varphi_i^T \varphi_i = 0. \end{cases} \tag{B21}$$

Using the method of Lagrange multipliers again results in a pair of eigenvalue problems

$$\begin{cases} \frac{1}{k^2} \cdot \Theta^T Q Q^T \Theta \cdot \phi_i = \lambda_i \cdot \phi_i \\ \frac{1}{k^2} \cdot Q^T \Theta \Theta^T Q \cdot \varphi_i = \lambda_i \cdot \varphi_i . \end{cases} \tag{B22}$$

The eigenvectors solving the above equations reveal the directions of the correlation extrema between input and output oriented latent variables. Again, the magnitude of the eigenvalue defines the significance of the corresponding latent variable. The three-step regression procedure in (B20) reduces in practice to a single matrix operation, because all the necessary dimension reduction is achieved in the first projection to the basis $\Phi^1 = (\phi_1 \mid \ldots \mid \phi_N)$. Therefore the PLS regression can be expressed as

$$\begin{aligned} \hat{Q}_{\text{est}} &= \Theta_{\text{est}} F_{\text{PLS}} \\ &= \Theta_{\text{est}} \cdot \Phi_{\text{PLS}} \left(\Phi_{\text{PLS}}^T \Theta^T \Theta \Phi_{\text{PLS}} \right)^{-1} \Phi_{\text{PLS}}^T \Theta^T Q, \end{aligned} \tag{B23}$$

where $\Phi_{\text{PLS}} = (\phi_1 \mid \ldots \mid \phi_N)$ and $N \leq \min\{n, m\}$.

B.3.3 Continuum Regression

The above two methods originate in slightly different foundations. In the PCA the modeling emphasis is exclusively on the input data whereas in the PLS both input and output data are considered. In both methods the latent variables are searched with an optimizing procedure: In PCA, the variance of the latent variables in the input space is maximized, and in PLS the correlation between input and output oriented subspace variables is maximized. Now one might ask which solution is better. The question is hard to answer since there is no physically optimal solution for the latent structure. And as it turns out, there are also several other ways in addition to the preceding two to define the latent basis structure.

The Continuum Regression (CR) attempts to combine the MLR, PCR and PLS methods into a single framework. Actually an innumerable amount of different regression methods can be defined with the same idea that will be presented in the following (see /20/ for more precise derivation). The latent variables can be defined as solutions to the eigenproblem

$$\frac{1}{k^{1+\alpha_1(1+\alpha_2)}} \cdot \Theta^T \left(Q \left(Q^T Q \right)^{\alpha_2} Q^T \right)^{\alpha_1} \Theta \cdot \phi_i = \lambda_i \cdot \phi_i . \tag{B24}$$

The resulting latent basis coincides with

- PCR if $\alpha_1 = 0$ and α_2 is an arbitrary number
- PLS if $\alpha_1 = 1$ and $\alpha_2 = 0$ and
- “MLR” if $\alpha_1 = 1$ and $\alpha_2 = -1$.

Above the MLR is in quotes because it actually is not a subspace projection method. However, it can be thought that in the MLR the emphasis is on explaining the maximal amount of the output variation with the regression model. The eigenproblem should not be solved in the preceding form since the matrix $Q \left(Q^T Q \right)^{\alpha_2} Q^T$ is huge in

dimension ($k \times k$) but the matrix power should be calculated with the singular value decomposition.

Both α_1 and α_2 can be expressed as a function of a parameter α such that the above "definitions" hold. There are several possibilities to define the functions $\alpha_1(\alpha)$ and $\alpha_2(\alpha)$, e.g.,

$$\begin{aligned} \alpha_1(\alpha) &= -2\alpha^2 + \alpha + 1 \\ \alpha_2(\alpha) &= 2\alpha - 1. \end{aligned} \tag{B25}$$

With these selections MLR is given with $\alpha = 0$, PLS with $\alpha = ½$ and PCR with $\alpha = 1$. Now we have a continuum between the before-mentioned methods and thus we are not forced to confine to them only, but any value of α can be chosen. After the best parameters N and α for the particular modeling task are found and the resulting latent basis $\Phi_{CR} = (\phi_1 \mid \ldots \mid \phi_N)$, $N \leq \min\{n, m\}$, is constructed, the final regression model based on the CR looks once again quite familiar, i.e.,

$$\begin{aligned} \hat{Q}_{\text{est}} &= \Theta_{\text{est}} F_{\text{CR}} \\ &= \Theta_{\text{est}} \cdot \Phi_{\text{CR}} \left(\Phi_{\text{CR}}^T \Theta^T \Theta \Phi_{\text{CR}} \right)^{-1} \Phi_{\text{CR}}^T \Theta^T Q. \end{aligned} \tag{B26}$$

B.3.4 Canonical Correlation Analysis

In the Canonical Correlation Analysis (CCA) the basic idea is very close to that of PLS: The aim is to find the subspace basis vectors such that the correlation of the input and the output variables is maximized. The difference is that the length of the basis vectors is not constrained in the optimization but the length of the projected data vectors, $Z_i = \Theta\phi_i$, is kept constant instead. This leads to the following constrained optimization problem

$$\begin{cases} f(\phi_i, \varphi_i) = \dfrac{1}{k} \cdot \phi_i^T \Theta^T Q \varphi_i \\ g_1(\phi_i) = 1 - \dfrac{1}{k} \cdot \phi_i^T \Theta^T \Theta \phi_i = 0 \\ g_2(\varphi_i) = 1 - \dfrac{1}{k} \cdot \varphi_i^T Q^T Q \varphi_i = 0. \end{cases} \tag{B27}$$

Proceeding similarly as above, the problem results in a pair of eigenproblems,

$$\begin{cases} \left(\Theta^T\Theta\right)^{-1} \Theta^T Q \left(Q^T Q\right)^{-1} Q^T \Theta \cdot \phi_i = \lambda_i \cdot \phi_i \\ \left(Q^T Q\right)^{-1} Q^T \Theta \left(\Theta^T \Theta\right)^{-1} \Theta^T Q \cdot \varphi_i = \lambda_i \cdot \varphi_i, \end{cases} \tag{B28}$$

that can be expressed also as a generalized eigenproblem:

$$\begin{cases} \Theta^T Q \left(Q^T Q\right)^{-1} Q^T \Theta \phi_i = \lambda_i \cdot \Theta^T \Theta \phi_i \\ Q^T \Theta \left(\Theta^T \Theta\right)^{-1} \Theta^T Q \varphi_i = \lambda_i \cdot Q^T Q \varphi_i. \end{cases} \tag{B29}$$

In (B28) it is assumed that both $\Theta^T\Theta$ and Q^TQ are invertible, which is clearly against the discussion in Chapter B.2. Thus, if either of them is close to singular, one of the generalized eigenproblems in (B29) should be solved.

Contrary to PCA and PLS basis vectors the CCA eigenvectors may have physical explanations as well. Also, the CCA bases are generally not orthogonal. Therefore the resulting regression formula reduces to a different form than before: With a subspace basis $\Phi_{\mathrm{CCA}} = (\phi_1 \mid \ldots \mid \phi_N)$, $N \leq \min\{n, m\}$, the estimation is performed as

$$\begin{aligned} \hat{Q}_{\mathrm{est}} &= \Theta_{\mathrm{est}} F_{\mathrm{CCR}} \\ &= \Theta_{\mathrm{est}} \cdot \Phi_{\mathrm{CCR}} \left(\Phi_{\mathrm{CCR}}^T \Phi_{\mathrm{CCR}} \right)^{-1} \left(\Phi_{\mathrm{CCR}}^T \Phi_{\mathrm{CCR}} \right)^{-1} \Phi_{\mathrm{CCR}}^T \Theta^T Q. \end{aligned} \tag{B30}$$