Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering
Department of Mathematics

Multivariate Adaptive Controller Design with Constraints under Uncertainty

Miroslav Novák

Ph.D. Thesis

Prague, October 2006

Preface

This thesis was prepared during my study at the Department of Mathematics in the Faculty of Nuclear Science and Physical Engineering, Czech Technical University, Prague. It is based on the research work carried out in the Department of Adaptive Systems, Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic.

First of all, I would like to thank to my supervisor, Josef Böhm, for posing the control problems and for his experienced guidance throughout all of the subsequent difficulties.

Secondly, let me give thanks to Miroslav Kárný, Jan Kracík and Josef Andrýsek for their useful advices and discussions about the topics involved, and also to Alena Halousková, for her careful manuscript proof reading and corrections to my English. And finally, I would like to express my appreciation to all of the members of the Department of Adaptive Systems for their stimulating discussions, which helped me enormously to complete my thesis.

This work has been supported by grants MŠMT 1M0572 (DAR), AV ČR 1075351, GA ČR 102/03/0049, AV ČR 1ET100750401 (BADDYR) and GA ČR 102/05/0271.

Contents

List of Figures

Abbreviations, Symbols and Notation

ARX	autoregressive model with external input
LQG	linear-quadratic controller with Gaussian disturbance
SISO	single-input single-output system
MIMO	multiple-input multiple-output system
PID	proportional-integral-derivative controller
pf	probability function
pdf	probability density function
x	quantity, scalar, or vector
x^*	set of possible values of the quantity x
$\mathring{x}$	number of elements of vector x
x'	transposition of x
$\propto$	proportional sign
$\|\| \cdot \|\|$	Euclidean norm
I_n	identity matrix of dimension n
$\mathrm{tr}(\cdot)$	matrix trace
$f\cdot,\ f(\cdot\|\cdot)$	*p(d)f*, conditional *p(d)f*
$\mathcal{N}(\mu, R)$	multivariate normal distribution
$Ex,\ E[x\|y]$	expected value, conditional expected value of x
$\mathrm{var}(x)$	variance of x
$\mathcal{D}_{\mathrm{KL}}(f\\|\hat{f})$	Kullback-Leibler distance of *p(d)f*s f and $\hat{f}$
$t,\ \tau$	discrete time
x_t	quantity x at the discrete time t
$x(t)$	quantity containing all the past history $x(t) = \{x_\tau\}_{\tau=1}^{t}$
u_t	system input at time t
y_t	system output at time t
d_t	vector consisting of input and output $d_t = [y_t', u_t']'$
d_t^{ref}	reference value of data at time t
φ_t	state vector
ψ_t	regression vector
Ψ_t	data record
∂	biggest time delay of data record

$\mathbf{R}$	set of real numbers
$\mathbf{N}$	set of natural numbers
q	tuning parameter of controller
$\mathcal{F}$	decision ignorance
$\mathcal{P}$	decision experience
$\mathcal{A}$	decision action
$\mathcal{B}$	decision innovation
Θ	model parameters
T	simulation length
h	control horizon

Chapter 1

Introduction

This work concerns the task of controller design. Main effort is dedicated to tuning of the controller being designed for uncertain system with constraints. The tuning is then included in the LQG controller design algorithm.

This chapter presents an overall introduction to the problem with remarks about previous work in this topic. It also states the aims and layout of the thesis.

1.1 Motivation

Control engineering deals with dynamic systems. Dynamic system is a part of reality with defined external variables that influence the system in controlled or uncontrolled manner. The former ones are called the control or input variables and the latter ones are called disturbances. The observable variables produced by the system are called outputs or controlled variables. The dynamic system contains also an internal state. The general description of system dynamics falls into many fields of human interests. It can be found in biological, social, economical, and technical environments. Thus, the control engineering studying possibilities of controlling these systems is very important.

The need of control for technical purposes was triggered by the industrial revolution in 19th century, where increasing technical level of machines being developed required proper regulation for their operation. The control theory as a scientific discipline had limited practical use of its result due to insufficient technical devices of that time. The controllers were constructed using mechanical parts, such as the famous Watt's regulator, which did not allow the use of advanced control methods.

With the progress in the electronics, the control engineering made a further step in the feedback control. This led to achievements such as the space flight. The digital controllers supported by the progress in microprocessor technology made possible the development of a new generation of controllers such as the model based, predictive, and adaptive ones.

With increasing capability of computers, more complicated mathematical models behind the controllers could be applied. On the other hand, a new difficulty appeared. Sophisticated controllers are more difficult to apply and commit. They are dependent on many tunable parameters, also called tuning knobs, which have to be properly set up. Unfortunately, the meaning of these tuning parameters is mostly far from the user's understanding of the control task and his objective. The setting up of the controller represents an obstacle of its applicability.

This situation contributes to the gap between theory and practice, where the advanced controllers developed in the control theory are not capable to penetrate the industry. Because of the ease of use and good experience, in the practice, the classical controllers, such as proportional-integral-derivative (PID) ones, are still dominating. Nevertheless, the modern model based controllers have much higher control potential. The predictive nature of model based controllers allows to find more efficient and precise control strategy, while the capability of self-tuning [**?**] and adaptivity [**?**] keeps the internal model up to date. The advantage of the modern controllers is even more remarkable in the case of controlling systems with multiple inputs and multiple outputs (MIMO), where abilities of the classical controllers are limited.

Still the gap between theory and practice in the control engineering is not closing. Even the opposite is true. Mathematical tools used in theory offer new design technologies, more powerful analysis and reliable numerical results. General approaches available have the difficulty to consider specific requirements of the given technology and even more the specificity of individual realizations. The setting of the modern controller is still a bottleneck waiting for its solution.

1.2 State of the Art

When looking in the literature about the controller tuning, a lot of contributions can be found about the tuning of the PID controllers [1]. On the other hand, there is almost nothing available about more complex controllers like predictive, LQG or H_∞ ones. The references [5], [4], [10] and [2] can be cited as the exceptional examples.

Companies applying more complex controllers, typically predictive ones, are mostly oriented to a specific technology. This gives them a chance to gradually build up internal tuning guidelines based mainly on experience and knowledge connected with the application domain. Different toolboxes are publicly available that help with transforming some input information to the controller parameters. For instance in the LQG design, the known mathematical model of the plant, noise intensities, and input and state penalization are easily converted into the controller parameters. In the H_∞ design, the nominal plant mathematical description with some standard parameter interval definitions can be used to initialize the controller parameters [6]. Often, it is claimed that the required input information can be gained from the final user and from identification toolboxes. In reality, available pieces of expertise cannot be easily merged and the individual toolboxes are not conceptually integrated.

Only recently, an attempt to create an automated controller design has appeared [16]. This work is conceptually similar to the approach presented in this thesis. It uses data measured on the real system to identify the model and simulations to tune the controller performance. As opposed to this thesis, the controller designed in [16] is based on the robust control theory and also the identification method, control objectives, and tuning itself are different.

1.2.1 Considered Properties of Controlled System

Constraints

An important property of the majority of real systems is that they have imposed constraints on the variables that must not be exceeded. The constraints are caused by the definition of the external variables, such as the percentage of opening a valve lies between zero and one hundred, or by safety and technological reasons for temperature, pressure, and so on. Of course, the tuned controller has to comply with the constraints. Otherwise, the proposed signal has to be limited outside of the controller, which spoils the controller internal prediction of the model behavior. The tuning parameters generally influence the effective range of controller variables, and so the tuning is a possible tool for ensuring the constraint compliance by the controller. The operational constraints that are used as a guide for the user to setup a controller by specifying a working point or working range, are also considered.

The general predictive controllers [8] are able to include the constraints directly in the calculation. However these controllers are designed only for known deterministic models, thus they are unusable for the task of the topic of this thesis.

Uncertainty

In reality, the controlled system is never known completely. Its properties can vary over time, which is caused for example by bearing wear of a machine. Other reason to consider the changes in system properties is caused by the fact that reality, even if known perfectly, has to be simplified due to the lacking ability of system identification method, or the controller model has to be of a restricted form because of the limited analytical or computational complexity of the controller. In this context, the system properties are interpreted as model parameters.

Another phenomenon in system knowledge is the uncertainty of its properties in the stochastic or random meaning. Firstly, the system variables can be influenced by a random noise. This is a common situation, and this kind of uncertainty is described by the system model. There exist controllers respecting this kind of stochasticity such as LQG controller with Kalman filter [?]. Secondly, even the model parameters can be considered uncertain. The designing of a controller for this kind of model leads to a hard problem such as dual control [?]. The stochastic nature of the system properties can be caused by physical construction, or by using an identification method for limited class of models, such as the linear ones. Then the reality that does not fit exactly to the identifiable model, is transformed into uncertainty of model parameters.

There are two approaches of coping with the uncertainty in the model parameters: The robustness and the adaptivity. A controller that is able to stabilize a set of models from a certain neighborhood of a nominal model is called robust [15, 14]. A controller that is able to track changes in the real system by recursively updating the internal model using measured data is called adaptive. The adaptive controller is able to react to the changes of the slowly varying model parameters [?]. If fast change is possible, the controller has to be capable to overcome the time period before the model is adapted.

1.2.2 Basis of the Thesis

The methods of controller design presented in this thesis are based on and continue in the effort to create a tool for the complete controller design developed in the Department of Adaptive Systems in Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic in Prague. The task of designing a controller, starting with information supplied by the user and measured data and finishing with prepared controller, is thoroughly described in [?]. The basic methods used are the autoregressive model with external input (ARX model) and the adaptive LQG controller. However the presented task of setting controller tuning parameters has the form of just several hints for the user. The problem of input constraints for multidimensional constrained variables is solved by optimal signal cutoff at the bounds for one step ahead prediction according to the total loss in the sense of controller quadratic criterion. This is described in more detail in [?]. The disadvantage of the signal cutting is that it spoils the LQG prediction.

The task of prior prediction of control quality of the LQG controller is presented in [?] in the form of quality evaluation from a sample simulation run using the Monte Carlo approach. The controller quality is measured by a quadratic criterion. The model is considered to have uncertain parameters, as the natural result of identification, and the controller is adaptive to reduce the drawback of the uncertainty. In fact, the probability density function (pdf) on model parameters is transformed into the pdf on the predicted values of the controller quality. This approach did not concern the constraints in the quality evaluation.

The next progress in the development of the complete LQG controller design including the automated tuning is described in [12]. It targets the evaluation of the controller quality in the sense of input constraints of a controlled uncertain model. The quality evaluation is used for the algorithmic optimization of the controller tuning parameters employing the golden section as the optimization method. The pdf of the uncertain model parameters is sampled and for each sample a separate simulation run is performed. The resulting controller quality is a statistic from the evaluated quality of these simulations.

The developed algorithms are employed in the single-dimensional version of Matlab software toolbox Designer [**?**]. This version of Designer was successfully applied in field of biotechnology for task of yeast cultivation [**?**].

A drawback of the used golden section optimization method is that it is suitable only for controllers with one tuning parameter. This limits the usability of the tuning algorithm for just subset of the single input single output (SISO) systems. The possible extension for the MIMO models and multivariate tuning parameters is sketched in [12] by using sequential optimization in every single optimization parameter, which can result in quite a poor convergence.

1.3 Aims of the Thesis

The aim of this thesis is to solve the task of a multivariate controller set up and to improve or even to allow the use of the modern controllers in the practice.

In brief, it consists of the following items:

1. Translate the constraints imposed on the variables and other requirements on the control loop into the tuning parameter values of the designed controller.

2. Create a tuning algorithm that considers uncertainty contained in the identified model.

3. Solve the task generally for the MIMO controller with multiple constraints and multiple tuning parameters.

4. Combine the tuning algorithm with an identification method to form a unified theoretical algorithmic approach to the controller design.

5. Make the algorithm computationally feasible.

1.4 Methods Used

The presented approach of the controller tuning is based on the Bayesian decision theory and the controller quality evaluation using the Monte Carlo method [?]. The tuning as an optimization method is performed by the sample path method [11]. The computational efficiency is supported by employing the stopping rules [?] using the Kullback-Leibler divergence [?]. The uncertainty of the controlled plant is treated using the Bayesian approach for the system identification [?] of the ARX and Markov chain (MC) models. The considered controller is the LQG controller and the controller based on fully probabilistic design [?]. The resulting algorithms are coded in Matlab.

1.5 Layout of the Thesis

The thesis is divided into 6 chapters briefly described below.

The introductory chapter 1 gives an overview of the control theory and motivation for the thesis. It shows what is missing in the current state of the art of the control design and presents the aims of this work.

Chapter 2 provides the basic mathematical and Bayesian decision making background necessary for the following chapters. It describes the used formalism such as probability distributions, dynamic system description, Bayesian identification and sampling. All methods used in the rest of the thesis are introduced with respect to the Bayesian theory.

Chapters 3 – 4 presents the new assets of this work. Chapter 3 develops the solution of the controller tuning. All the aspects are firstly described generally and then the mathematical formulation and solution is given. Two functions are defined to evaluate the controller quality according the user's requirements. The evaluation is done under consideration of uncertain model parameters. The effective evaluation using the on-line stopping rules is employed and the stochastic optimization problem algorithm described.

Chapter 4 applies the controller design method to the case of the adaptive LQG controller with the ARX model. It defines the controller tuning parameters, and gives two possible initial tuning parameter estimates.

Chapter 5 illustrates the controller design process on several simulated experiments.

Chapter 6 summarizes the results of this thesis with some general remarks and it points out some open problems to be solved in the near future.

Chapter 2

Decision Making under Uncertainty

Designing a controller for an incompletely known system model obtained only from prior information and measured data influenced by random disturbance inevitably leads to the problem of decision making under uncertainty.

This chapter describes the uncertainty and the decision making using the Bayesian approach, where the uncertainty is interpreted in the same manner as randomness. The decision making is described using the probability formalism firstly in a general form, and then it is applied to particular decision tasks used in the controller design. Finally, the system identification is described using the exponential family [**?**] and ARX model.

2.1 Uncertainty and Decision Making in Controller Design

This section gives motivation for use of the Bayesian approach for the controller design. It briefly sketches the tasks of system identification and controller design by utilizing the decision making under uncertainty.

2.1.1 Uncertainty of System Model

The system is identified using a parameterized model. The model considers uncertainty, which is caused by the natural signal disturbances of the system and by the fact that we do not know the parameters of the model exactly. The only knowledge about the system is brought by the measured data and some prior information. These information sources do not lead to an exact estimate of the model parameters values. Let us consider a point estimate of the model parameters obtained from data measured at one time interval and a second estimate using data measured from another time interval. The estimated model parameters will be almost surely different. This behavior is caused by the incomplete information contained in the measured data. The other set of data includes different driving signals or different system disturbance which emphasize slightly different properties of the identified system.

To cope with this uncertainty, we are identifying not just the point estimate but the whole probability distribution of the parameters that expresses the uncertainty. This approach acquires more complete, or more fair, information about the model parameters from the data.

Of course, the real system can change its properties, too. This is the uncertainty caused not only by the actual identification data sample, but by the reality itself.

2.1.2 Controller as a Decision Maker under Uncertainty

A controller is designed to drive the system by generating a suitable action signal in order to achieve the desired closed loop behavior. The only connection to the reality available is through the identified uncertain model described in the previous section. In fact, the controller is designed to control the identified model.

The act of the action generation is, in other words, an act of decision making under uncertainty. Because of the dynamic system model, it is the dynamic decision making task, which is a computationally hard problem. Therefore the suboptimal approach is chosen.

The tuning combines the available controller with limited capabilities and tunes its performance according to the specified aims.

2.2 Bayesian Decision Making

The Bayesian theory of the decision making under uncertainty is an advantageous tool for the controller design. The theory is based on two points. Firstly, it uses probability to describe unknown or partially known reality. Secondly, the loss function serves to judge the possible decision acts according to its, of course uncertain, consequences. The decision with smallest expected loss function is chosen as the optimal one.

This chapter describes briefly the parts of the Bayesian theory that are needed for this work. For more details on this topic see [**?**].

In the Bayesian theory the concept of uncertainty coincides with the concept of randomness. This holds for the case where it is a matter of a real randomness, such as random disturbance, as well as for the case where it is just something unknown, such as a parameter, which is known to be constant but its value is unknown to us.

This concept of uncertainty relates to the interpretation of probability [**?**]. In the Bayesian theory, probability is not interpreted in terms of relative frequencies but more generally as a subjective degree of belief of a rationally and consistently reasoning person which is used to describe quantitatively the considered uncertainty.

The connection of the uncertainty, as a subjective belief, with the reality is done by transforming the uncertainty according to measured data. It is brought into effect by using conditional probability and Bayes' rule which is described later.

2.2.1 Notation and Properties of Probability

In this section, the notation and basic properties of the probability calculus are given with remarks about its relation to the uncertainty.

Notation:

Quantity is a mapping with a numerical range, i.e. a subset of the multivariate, real-valued space.
The domain and form of the quantity are mostly unused and unspecified. The introduced notion corresponds with random variable used in the probability theory. The use of the alternative term should stress that the probability serves us as a tool adopted for the decision making under uncertainty. The term quantity stresses our orientation on numerical values that arise mostly by observing physical quantities.

Realization is a value of the quantity.
The random quantity and its realization are not distinguished, as usual. The proper meaning is determined by context.

Domain α^* denotes the set of possible values of the quantity α.

Dimension $\mathring{\alpha}$ denotes the number of elements of vector α.

Probability (density) function Symbol f is reserved both for probability functions (*pf*) of discrete quantities and probability density functions (*pdf*) of quantities of continuous type. The meaning of the *p(d)f* is given through the identifier of its argument. Implicitly all the general relations are defined for the quantities of the continuous type. One has only to keep in mind that the integration has to be replaced by regular summation whenever the argument is discrete.

Conditional pdf $f(\beta|\alpha,\gamma)$ of β conditioned by α, γ is a pdf on β^* restricting $f(\mathcal{Q})$ on the cross-section of $\mathcal{Q}^*$ given by a fixed α, γ. The following random variables are used $\mathcal{Q} = (\alpha, \beta, \gamma)$.

The conditioning symbol $|$ is dropped if just trivial conditions are considered.

Joint pdf $f(\alpha,\beta|\gamma)$ of α,β conditioned on γ is a pdf on $(\alpha,\beta)^*$ restricting $f(\mathcal{Q})$ on the cross-section of $\mathcal{Q}^*$ given by a fixed γ.

Marginal pdf $f(\alpha|\gamma)$ of α conditioned on γ is a pdf on α^* restricting $f(\mathcal{Q})$ on the cross-section of β^* given by a fixed γ with no information on β.

Probability $\mathbf{P}\{\alpha \in A\}$ denotes probability of a quantity α being inside a set $A \subset \alpha^*$. It holds $\mathbf{P}\{\alpha \in A\} = \int_A f(\alpha)d\alpha$.

Expectation of function $Z(\alpha,\beta,\gamma)$ under the condition γ

$$E[Z(\alpha,\beta,\gamma)|\,\gamma] = \int Z(\alpha,\beta,\gamma)f(\alpha,\beta|\gamma)d\alpha d\beta \tag{2.1}$$

Calculus with pdfs:

Our manipulations with the introduced pdfs rely on the following calculus. For generic random quantities α,β,γ it holds:

Non-negativity $f(\alpha,\beta|\gamma),\ f(\alpha|\beta,\gamma),\ f(\beta|\alpha,\gamma),\ f(\beta|\gamma) \geq 0.$

Normalization $\int f(\alpha,\beta|\gamma)\,d\alpha d\beta = \int f(\alpha|\beta,\gamma)\,d\alpha = \int f(\beta|\alpha,\gamma)\,d\beta = 1.$

Chain rule $f(\alpha,\beta|\gamma) = f(\alpha|\beta,\gamma)f(\beta|\gamma) = f(\beta|\alpha,\gamma)f(\alpha|\gamma).$

Marginalization $f(\beta|\gamma) = \int f(\alpha,\beta|\gamma)\,d\alpha,\ f(\alpha|\gamma) = \int f(\alpha,\beta|\gamma)\,d\beta.$

Bayes' rule

$$f(\beta|\alpha,\gamma) = \frac{f(\alpha|\beta,\gamma)f(\beta|\gamma)}{f(\alpha|\gamma)} = \frac{f(\alpha|\beta,\gamma)f(\beta|\gamma)}{\int f(\alpha|\beta,\gamma)f(\beta|\gamma)\,d\beta} \propto f(\alpha|\beta,\gamma)f(\beta|\gamma). \tag{2.2}$$

The proportion sign $\propto$ means that the factor independent of β and uniquely determined by the normalization is not explicitly written in the equality presented.

Independence equivalents

$$f(\alpha,\beta|\gamma) = f(\alpha|\gamma)f(\beta|\gamma) \Leftrightarrow f(\alpha|\beta,\gamma) = f(\alpha|\gamma) \text{ or } f(\beta|\alpha,\gamma) = f(\beta|\gamma). \tag{2.3}$$

2.3 Basic Tasks

The Bayesian decision making is presented in this section in general form. The modeled world is divided into several parts representing known, unknown, measurable, and action quantities. These quantities are used to show the base of the decision making and learning.

Let us divide the modeled world into the following generally multivariate quantities

experience $\mathcal{P}$ represents the quantities whose values are known and are available to the decision maker. The experience $\mathcal{P}$ contains usually the data already measured in the time of the decision making.

action $\mathcal{A}$ represents the quantity available for the decision maker to express his decision. It is under full control of the decision maker.

ignorance $\mathcal{F}$ represents quantities unknown to the decision maker at the time of selecting of the action. Their uncertainty is influenced by the selected action.

innovation $\mathcal{B}$ represents a part of ignorance directly measurable after an action is applied.

behavior $\mathcal{Q}^*$ consists of all possible realizations (of trajectories) $\mathcal{Q}$, i.e. values of all the quantities within the time span determined by the horizon of interest that are related to the system and considered by the decision maker.

The realization $\mathcal{Q}$ can be split with respect to any decision $\mathcal{A} \in \mathcal{A}^*$ into the relevant experience $\mathcal{P}$ and ignorance $\mathcal{F}$, formally $\mathcal{Q} = (\mathcal{P}, \mathcal{A}, \mathcal{F})$.

The specific meaning of these quantities for particular purposes is given later on.

2.3.1 Decision Making

This section concerns the problem of decision making as developing of a causal decision rule $\mathcal{R} : \mathcal{P}^* \mapsto \mathcal{A}^*$ selecting a proper action $\mathcal{A}$ using the experience $\mathcal{P}$ according to the given criterion.

The relationship between $\mathcal{F}$, $\mathcal{A}$, and $\mathcal{P}$ is described by their join pdf or the pdf on the behavior

$$f(\mathcal{F}, \mathcal{A}, \mathcal{P}) = f(\mathcal{Q}).$$

To judge the consequences, a criterion selecting the optimal decision is represented by a loss function assigning a numerical value to every possible system behavior $\mathcal{Q} = (\mathcal{F}, \mathcal{A}, \mathcal{P})$

$$\mathcal{Z} : \mathcal{Q}^* \mapsto \mathbf{R} \tag{2.4}$$

The optimal decision rule minimizes the expected loss value

$$\mathcal{R} = \arg \min_{\mathcal{R}:\mathcal{P}^* \mapsto \mathcal{A}^*} E[\mathcal{Z}(\mathcal{F}, \mathcal{A}, \mathcal{P})]. \tag{2.5}$$

It is possible to construct the decision rule value-wise [?] so that for every $\mathcal{P} \in \mathcal{P}^*$

$$\mathcal{R}(\mathcal{P}) = \arg \min_{\mathcal{A} \in \mathcal{A}^*} E[\mathcal{Z}(\mathcal{F}, \mathcal{A}, \mathcal{P}) | \mathcal{A}, \mathcal{P}]. \tag{2.6}$$

Here we assume the uniqueness and existence of the minimum. If there are more absolutely minimizing arguments, then it gives no preferences. It is possible to use for example a randomized strategy [?].

2.3.2 Learning

For the decision making, a model describing the dependency of the innovation $\mathcal{B}$ as a consequence of the performed action $\mathcal{A}$ and the past $\mathcal{P}$

$$f(\mathcal{B}|\mathcal{A}, \mathcal{P}) \tag{2.7}$$

is needed for calculation of the expected value in (2.6). The task of Bayesian learning is to identify the outer model using prior information and measured data.

To describe the unknown outer model, a set of possible models of the reality is used. The models in this class are parameterized by a parameter Θ

$$f(\mathcal{B}|\mathcal{A}, \mathcal{P}, \Theta). \tag{2.8}$$

The parameter Θ is an unknown, never observed, quantity. It is a part of the ignorance $\mathcal{F}$. The learning updates the prior knowledge about the parameter Θ by incorporating the performed action and the measured consequence.

It is assumed that there is a prior information available conditioned by the experience

$$f(\Theta|\mathcal{P}). \tag{2.9}$$

The outer model (2.7) is obtained by the chain rule

$$f(\mathcal{B}|\mathcal{A}, \mathcal{P}) = \int f(\mathcal{B}|\mathcal{A}, \mathcal{P}, \Theta) f(\Theta|\mathcal{P}) d\Theta,$$

where we assume the natural conditions of control hold

$$f(\Theta|\mathcal{P}) = f(\Theta|\mathcal{P}, \mathcal{A}).$$

The natural conditions of control express the fact that performing an action without knowing its consequence brings no information about the model parameters.

The task of the identification uses the performed action $\mathcal{A}$ and innovation $\mathcal{B}$ (measured data) to update information about the parameter Θ.

Posterior information about parameter Θ, conditioned by the action $\mathcal{A}$ and the innovation $\mathcal{B}$ is calculated using the Bayes' rule from the parameterized model (2.8), prior information (2.9), and the outer model (2.7) as the normalizing factor

$$f(\Theta|\mathcal{B}, \mathcal{A}, \mathcal{P}) = \frac{f(\mathcal{B}|\mathcal{A}, \mathcal{P}, \Theta) f(\Theta|\mathcal{P})}{f(\mathcal{B}|\mathcal{A}, \mathcal{P})}. \tag{2.10}$$

In this way, the information of the parameter is updated after the action and the corresponding innovation is available.

2.3.3 Recursive Case

In the case of dynamic systems, recursive decision making and learning is usual. Data measured in the previous control step are used to update the information about parameters Θ, which is in turn used for action selection in the next step.

Let us consider a sequence of $T \in \mathbf{N}$ successive decision making tasks. The quantities used forms sequences $\{\mathcal{A}_\tau\}_{\tau=1}^T$, $\{\mathcal{B}_\tau\}_{\tau=1}^T$. The experience sequence $\{\mathcal{P}_\tau\}_{\tau=0}^T$ starts from an empty initial experience $\mathcal{P}_0$, which contains no measured data, and corresponding prior pdf $f(\Theta|\mathcal{P}_0) = f(\Theta)$. The successive elements of the sequence accumulate the actions and innovation

$$\mathcal{P}_t = [\mathcal{B}_t, \mathcal{A}_t, \mathcal{P}_{t-1}].$$

The strategies used form a sequence $\{\mathcal{R}_\tau\}_{\tau=1}^T$, such that $\mathcal{A}_t = \mathcal{R}_t(\mathcal{P}_t)$.

The learning process follows Section 2.3.2. From the experience $\mathcal{P}_t$ we get the conditional pdf of the parameter Θ

$$f(\Theta|\mathcal{P}_t) = f(\Theta|\mathcal{A}_t, \mathcal{B}_t, \mathcal{P}_{t-1}). \tag{2.11}$$

This pdf is then used to obtain the updated outer model

$$f(\mathcal{B}_t|\mathcal{A}_t, \mathcal{P}_{t-1}) = \int f(\mathcal{B}_t|\mathcal{A}_t, \mathcal{P}_{t-1}, \Theta) f(\Theta|\mathcal{P}_t) d\Theta.$$

And finally, this pdf is then used for selecting action $\mathcal{A}_t$ by a strategy $\mathcal{R}_t$. The innovation $\mathcal{B}_t$, measured after this action is performed, is used for updated parameter pdf $f(\Theta|\mathcal{P}_{t+1})$, which closes the recursion.

Dynamic Design

The dynamic design is a multiple step ahead control. It is a recursive decision making task, where the whole sequence of decision rules is constructed in the time horizon $t \in \{1, 2, \dots, T\}$.

The optimal strategy can be found by using a stochastic version of dynamic programming [?]. The optimal causal strategy

$$\{\mathcal{R}_t^o : \mathcal{P}_t^* \to \mathcal{A}_t^*\}_{t=1}^T$$

acting on experience $\mathcal{P}_t$ and minimizing the expected loss function $E[\mathcal{Z}(\mathcal{Q})]$ can be constructed in the following way: For every $t = 1, 2, \dots, T$ and each $\mathcal{P}_t \in \mathcal{P}_t^*$, it is sufficient to take a minimizing argument $\mathcal{A}^o(\mathcal{P}_t)$ in

$$\mathcal{V}(\mathcal{P}_t) = \min_{\mathcal{A}_t \in \mathcal{A}_t^*} E[\mathcal{V}(\mathcal{P}_{t+1})|\mathcal{A}_t, \mathcal{P}_t], \ t = 1, 2, \dots, T \tag{2.12}$$

as the decision generated by the t-th rule of the optimal strategy, i.e. $\mathcal{A}^o(\mathcal{P}_t) = \mathcal{R}_t^o(\mathcal{P}_t)$.

The recursion (2.12) is performed in the backward manner against the course given by the increasing experience. It starts with

$$\mathcal{V}(\mathcal{P}_{T+1}) \equiv \mathcal{E}[\mathcal{Z}(\mathcal{Q})|\mathcal{P}_{T+1}]. \tag{2.13}$$

The reached minimum has the value $E[\mathcal{V}(\mathcal{P}_1)] = \min_{\{\mathcal{R}_t\}_{t=1}^T} E[\mathcal{Z}(\mathcal{Q})]$.

Adaptation and Forgetting

In Section 2.3.2, the parameter Θ was considered constant. However in reality the properties of the modeled system may change or the working point where the model was identified is moved. To allow change of the model parameters, the information accumulation by equation (2.10) is complemented also by the information forgetting. This is realized by flattening of the pdf on parameters $f(\Theta)$. This is a heuristic approach of the adaptation. For the precise formulation of the adaptation using Bayesian filtering see [**?**].

2.4 Construction Elements

The basic tasks of the Bayesian decision making were introduced in the previous section 2.3.3 in general form. This section gives an interpretation of the general decision making for various tasks connected with the real problems solved in this thesis.

The Bayesian decision making is used here in the three roles. Firstly, it is the identification of the dynamic system and its control. Second role is the controller tuning task and the last one concerns the on-line stopping rule to make the simulations and evaluation fast.

2.4.1 System Description and Identification

The primary object of the controller tuning is the system to be controlled. Its knowledge, in the form of system model, has to be as precise as possible to achieve a good controller design.

The quantities of the system (and its model) in time instant t are described in the following list and their assignment to the general form of experience, action, innovation, etc. noted in Section 2.3 is given.

System input denoted by u_t, is a quantity available for direct control – it corresponds to the action $\mathcal{A}_t$

System output denoted by y_t, is a quantity measurable after the input u_t is applied – it corresponds the innovation $\mathcal{B}_t$

Data record denoted by d_t, is a quantity containing both the input and output $d_t = [y_t, u_t]$

A special notation for the quantity containing all data from the time instant 1 to t is denoted by

$$d(t) = \{d_\tau\}_{\tau=1}^{t} \tag{2.14}$$

– it corresponds to the experience $\mathcal{P}_t$.

The class of considered models of the real system is parameterized by the parameter Θ

$$f(y_t|u_t, d(t-1), \Theta). \tag{2.15}$$

The prior information about the model is represented by a prior pdf on parameters $f(\Theta)$ that is not conditioned by the measured data.

The model is used to perform the parameter identification as described in Section 2.3.3. The estimated parameter pdf $f(\Theta|y_t, u_t, d(t-1))$ is calculated recursively according to (2.10) with substituted quantities as described above.

More specific description of the identification for the case of the exponential family models is given in Section 2.5 of this chapter.

2.4.2 Controller

The controller is defined by requirements imposed on the closed loop behavior

Control criterion is a function of the closed loop data – it corresponds to the loss function $\mathcal{Z}$

Input range available for control u^* – it corresponds to the action domain $\mathcal{A}^*$

The controller for the given model is obtained just by substituting the system inputs and outputs into the general scheme presented in the previous section and applying the decision making task as described in Section 2.3.1.

The control criterion is parameterized by a quantity q called tuning parameter. Tuning parameter shapes the controller properties through weights of particular terms in the criterion. The controller is determined by the selection of the tuning parameter value. Therefore the resulting strategy $\mathcal{R}_t$ is parameterized by the tuning parameter

$$\mathcal{R}_t : (d(t-1)^*, q^*) \mapsto u_t^*. \tag{2.16}$$

The example LQG controller and its criterion with the tuning parameter specified is described in Section 4.

2.4.3 Controller Tuning

The controller tuning is a task of searching for such a controller, or its tuning parameters that satisfy the specified requirements best.

The tuning can be interpreted again as a decision making task, where the searched tuning parameters represent the action to be made, the consequence are the resulting closed loop data, and the loss function describes the user's requirements. The tuning as the main purpose of this work is described in Section 3.

2.4.4 On-Line Stopping Rule

The controller tuning minimizes the loss function defined on closed loop data. As the minimization is done numerically, the data are obtained from simulation, which is run many times until the optimum is found, thus there are big computational demands. The need of making the controller design computationally efficient leads to the following question: How long has the simulation be in order to obtain enough information for judging the controller quality?

This task is solved by an on-line stopping rule that triggers the simulation stop when there is enough information collected. The decision is made for for the noise compensation task in Section 3.5.2 using the Bayesian decision making approach.

2.5 System Model

In this section, we describe the types of models used for the identified system as sketched in Section 2.4.1. First the general dynamic model from exponential family is given, and then the ARX model as a particular member of the family. Note that some other models are used in this work, too, but they are defined at the respective sections where they are needed.

Now, some notation needed for the following text is presented. State φ_t is a vector containing a finite number ∂ of past data

$$\varphi_t = [y'_t, u'_t, y'_{t-1}, u'_{t-1}, \ldots, y'_{t-\partial+1}, u'_{t-\partial+1}, 1]',$$

which holds sufficient past information for generating next system output y_{t+1}, provided the actual input u_{t+1} is available. The last element of the state is the constant unit which is used to simplify the notation for the case of the constant offset being present in the model. To complete the notation the regression vector φ_t and data vector Ψ_t are introduced

$$\begin{aligned}
\Psi_t &= [y'_t, u'_t, y'_{t-1}, u'_{t-1}, \ldots, y'_{t-\partial}, u'_{t-\partial}, 1]' \\
\psi_t &= [u'_t, y'_{t-1}, u'_{t-1}, \ldots, y'_{t-\partial}, u'_{t-\partial}, 1]' \\
\Psi_t &= [y'_t, \psi'_t]' = [y'_t, u'_t, \varphi'_{t-1}]'.
\end{aligned}$$

2.5.1 Exponential Family

A system model pdf (2.15) belongs to the exponential family if it can be written in the form

$$f(y_t|u_t, d(t-1), \Theta) = f(y_t|\psi_t, \Theta) = A(\Theta)\exp(\mathrm{tr}(B'(\Psi_t)C(\Theta))), \qquad (2.17)$$

where A is a non-negative real function on Θ^*, and B and C are multivariate functions with compatible dimensions defined on Θ^* and Ψ_t^*.

The corresponding conjugated pdf of Θ [**?**] is

$$f(\Theta|d(t)) = f(\Theta|V_t, \nu_t) = \frac{A^{\nu_t}(\Theta)\exp(\mathrm{tr}(V_t'C(\Theta)))f(\Theta)}{I(V_t, \nu_t)}, \qquad (2.18)$$

where $I(V_t, \nu_t)$ is normalizing factor

$$I(V_t, \nu_t) = \int A^{\nu_t}(\Theta)\exp(\mathrm{tr}(V_t'C(\Theta)))f(\Theta)d\Theta$$

and V_t and ν_t are sufficient statistics updated recursively

$$\begin{aligned} (V_{t-1}, \nu_{t-1}, \Psi_t) &\mapsto (V_t, \nu_t) \qquad (2.19)\\ V_t &= V_{t-1} + B(\Psi_t) \\ \nu_t &= \nu_{t-1} + 1. \end{aligned}$$

The symbol $\mapsto$, in this context, denotes existence of such a function that maps variable on the left side to the variable on the right side. The recursion (2.19) starts from V_0 and ν_0 determining the prior conjugated pdf $f(\Theta)$, see [**?**].

The predictive pdf of the system can be obtained from (2.17) and (2.18) by applying (2.11)

$$f(y_t|u_t, d(t-1)) = f(y_t|\psi_t, V_{t-1}, \nu_{t-1}) = \frac{I(V_t, \nu_t)}{I(V_{t-1}, \nu_{t-1})}. \qquad (2.20)$$

2.6 Gaussian ARX model

An important type of dynamic model from the exponential family is the Gaussian ARX model. This model with parameter Θ consisting of regression coefficients θ and Gaussian noise covariance R has pdf (2.17) realized in the form

$$f(y_t|\psi_t, \Theta) = f(y_t|\psi_t, \theta, R) \sim \mathcal{N}(\theta\psi_t, R), \qquad (2.21)$$

where $\Theta = (\theta, R)$ and $\theta \in \mathbf{R}^{\mathring{y}, \partial+1}$ and $R \in \mathbf{R}^{\mathring{y}, \mathring{y}}$. The relationship to the exponential family (2.17) is given by functions A, B, and C:

$$\begin{aligned} A(\Theta) &= (2\pi)^{-\frac{\mathring{y}}{2}}|R|^{-\frac{1}{2}} \\ B(\Psi_t) &= \Psi_t\Psi_t' \\ C(\Theta) &= -\frac{1}{2}\begin{bmatrix} -I \\ \theta \end{bmatrix} R^{-1} \begin{bmatrix} -I \\ \theta \end{bmatrix}'. \end{aligned}$$

The random variable Θ from pdf (2.18) has now conjugated Gauss-inverse-Wishart distribution [?]

$$f(\theta, R|V_t, \nu_t) = \alpha_t |R|^{-\frac{\nu_t}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\left(R^{-1}\left[\begin{array}{c} -I \\ \theta \end{array}\right]' V_t \left[\begin{array}{c} -I \\ \theta \end{array}\right]\right)\right\}, \quad (2.22)$$

where α_t is the normalizing constant.

The system output y_t from the single step ahead predictive pdf (2.20) has now the Student's distribution

$$f(y_t|\psi_t, V_{t-1}, \nu_{t-1}) = \frac{\kappa_t}{\left(1 + \frac{\hat{e}_t' \Lambda_{t-1}^{-1} \hat{e}_t}{1+\zeta_t}\right)^{\frac{\nu_{t-1}+\mathring{y}}{2}}}, \quad (2.23)$$

where

$$\hat{e}_t = y_t - \hat{\theta}_{t-1}' \psi_t \quad (2.24)$$

and κ_t is the normalizing constant. Variables $\hat{\theta}_t$, Λ_t, and ζ_t are obtained from the split

$$V_t = \left[\begin{array}{cc} V_{y,t} & V_{y\psi,t}' \\ V_{y\psi,t} & V_{\psi,t} \end{array}\right], \quad \text{with } \mathring{y}\text{-dimensional square } V_{y,t} \quad (2.25)$$

and

$$\begin{aligned} \hat{\theta}_t &= V_{\psi,t}^{-1} V_{y\psi,t} \\ \Lambda &= V_{y,t} - V_{y\psi,t}' \hat{\theta}_t \\ \zeta_t &= \psi_t' V_{\psi,t}^{-1} \psi_t. \end{aligned}$$

The Student's distribution (2.23) with the number of degrees of freedom going to infinity converges to the Gaussian distribution

$$f(y_t|\psi_t, V_{t-1}, \nu_{t-1}) \stackrel{t\to\infty}{\to} \mathcal{N}\left(\hat{\theta}_t' \psi_t, \frac{1+\zeta_{t-1}}{\nu_{t-1} - \mathring{\psi}} \Lambda_{t-1}\right) \quad (2.26)$$

and in simulations it is reasonable to sample this approximation instead.

The matrix V_t can be decomposed $V_t = L_t' D_t L_t$ [?], where L_t is a lower triangular matrix with unit diagonal and D_t is diagonal matrix with positive diagonal entries. This is advantageous because of better numerical stability, see [?]. Here it is noted only because a split of these matrices is referred in Section 3.5.4. The split is defined by conjugated

$$\begin{aligned} L_t &= \left[\begin{array}{cc} L_{y,t} & 0 \\ L_{y\psi,t} & L_{\psi,t} \end{array}\right] \\ D_t &= \left[\begin{array}{cc} D_{y,t} & 0 \\ 0 & D_{\psi,t} \end{array}\right] \end{aligned} \quad (2.27)$$

where the dimensions of blocks are the same as in (2.25).

Chapter 3

Controller Design Problem Formulation and Solution

This chapter covers the controller design beginning with the description of the task of controller tuning under uncertainty. Then the tuning is defined formally as a Bayesian decision making task with particular loss function proposed for specific situations. The rest of this chapter deals with the computational aspects of the tuning.

3.1 Description of Controller Tuning

Controller tuning is an off-line process aiming at correct controller set-up to fulfill given requirements and constraints. The controller is parameterized by so called tuning parameter, which is usually multidimensional. The tuning parameter has to be set properly to obtain desired control loop behavior.

The controller tuning transforms the user's requirements into the proper tuning parameter values

$$\text{user's requirements} \longrightarrow \text{tuning parameter values}$$

The task of the tuning is to solve the following two problems:

1. Incompatible user's requirements and tuning parameter – The user's requirements are considered to be expressed in a human understandable form, while the tuning parameter has its form determined by the selected class of controllers. The translation from the user's requirements to the tuning parameter can be very difficult for complex controller.

2. Incompatible identified and controlled model – A model obtained from the Bayesian identification is usually more general than the type of model acceptable for the selected class of controllers. Thus even if the

user's requirements were compatible with the tuning parameter, the resulting properties could not be guaranteed.

In the current situation, the tuning for model based controllers is usually performed manually using experience of a control engineer expert who is familiar with the controlled system and controller type. The difficulty of the manual tuning is raised in the case of multivariate controllers, where the number of tuning parameters increases. Because of possible mutual dependency of the effects of these parameters, the result of the manual tuning in this case can hardly reach the optimality. The properties of controller tuning we are dealing with are described in the following sections.

3.1.1 User's Requirements

The tuning is aiming at fulfillment of the requirements given by the user. They are divided in two kinds. First kind of requirement represents the constraints imposed on the system input generated by the controller. The typical form of this constraint is a bounding interval imposed on the input magnitude or its increments.

If there are more controllers that satisfy the constraints, we want to choose the best of them. Therefore the second kind of requirement is to minimize some loss function measuring the controller behavior such as the output error with respect to the prescribed reference setpoint.

3.1.2 Tuning Parameters

The tuning parameters influence the behavior of the considered controller, but in rather complicated way for the user. This situation is caused by the controller construction. The control strategy has to be necessarily calculated on-line for the feedback controller, so it has to be possible to evaluate it fast. The control criterion employing the tuning parameter and the system model have to be in a suitable form to make possible efficient computation, such as the linear model and a the quadratic criterion for the LQG controller. However the suitable form of the controller for computation need not be parametrized by suitable tuning parameters from the user's point of view.

3.1.3 Controlled Model Uncertainty

The controller tuning has to consider the uncertainty contained in the controlled system model. The system model is uncertain from the two reasons.

- the model is influenced by random disturbances of the state and signals
- the model parameters are uncertain

The first kind of uncertainty models the natural disturbances present in the system and also the measurement noise.

The second kind of uncertainty is not a random disturbance but it represents the fact the parameters are unknown, as the Bayesian statistics can only transform prior information about the model parameters into the posterior one rectified by the measured data. Thus the model parameters are uncertain and described by a pdf.

The selected type of model based controller can deal with only a specific type of model. Since it must be possible to evaluate the controller action fast, as described in Section 3.1.2, the possible model is the linear one with known parameters. Classical approach is to use point estimates of parameters for the controller model. However the approach used in this thesis is a different one. It uses for tuning the whole information obtained by identification in the form of pdf of the model parameters. Thus, the resulting controller is tuned to work well not only with the most likely parameters, but ensuring the proper operation over the whole parameters range with respect to their probability.

Even more interesting utilization of the uncertain knowledge of the model parameters is possible for the case of adaptive controller design. The adaptive controller uses its own internal model, whose parameters are the point estimates updated in order to reflect the varying real controlled system, see Section 3.3.2 and Figure 3.2. The controller performance is evaluated in the closed loop connected to the identified model with uncertain parameters. This model samples its parameters during the simulation. Thus the adaptiveness of the designed controller is exercised in an environment close to the class of probable real systems. Utilizing the uncertainty in this way gives to the automated controller design a new ability to find an adaptive controller with the predicted behavior close to the implementation on the real system.

3.1.4 Constraints under Uncertainty

Because of the fact that the signals and parameters are uncertain, the problem of perfect constraint satisfaction by the simulated signals is difficult to be assured. The pdfs of uncertain quantities have usually infinite supports. Therefore the constraints are redefined to be satisfied at least with a specified, high enough, probability.

Now, what happens if a constraint is in some time instant exceeded? From the user's viewpoint, the constraints can be divided into two groups. First group contains the constraints enforced by the physical composition of the system. These constraints are called hard and they have to be met at all costs. If a violation occurs, the signal is cut to fit into the bounds. However this cutting can negatively influence the model based controller prediction and the whole control quality. Nevertheless, as the tuning finds a controller that satisfies the constraints with a high probability, the signal corrections are rare and can be included into the unmeasurable disturbance of the closed loop system.

The second group of constraints, called soft, is used artificially by the user to specify his objectives in a convenient form. These constraints are not required by the system and violating them in a small allowed probability has a negligible negative effect on the control loop, so they are not cut to fit into the bounds.

3.2 Tuning as a Bayesian Decision Task

This section describes the controller tuning as a Bayesian decision making task as defined in Section 2.3 and applied for the particular case of the closed loop, user-defined constraints, controller quality and tuning parameters. The particular construction elements are described in terms of experience, action, innovation and decision making. First of all let us present the tuned closed loop.

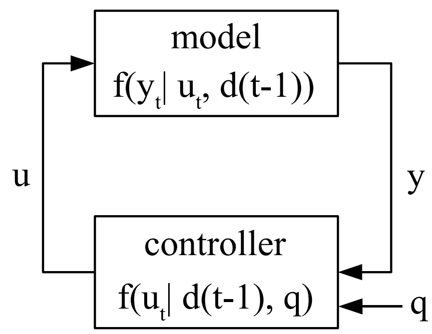

Figure 3.1: Closed loop.

The classical interconnection between controlled system and controller, see Figure 3.1, generates closed loop data $d(T)$, see (2.14), of length T. The data collect the input u_t driven by the controller and the output y_t measured on the controlled system as described in Section 2.4.1. The closed loop forms a stochastic system as the controlled model is considered to be influenced by a random disturbance. Thus the model behavior is described by pdf $f(y_t|u_t, d(t-1))$. The controller is described generally as a random one by pdf $f(u_t|d(t-1), q)$, where q denotes the tuning parameter. The closed loop data $d(T)$ are therefore also a random variable described by pdf obtained by application of the chain rule over the horizon T

$$f(d(T)|q) = \prod_{t=1}^{T} f(y_t|u_t, d(t-1)) f(u_t|d(t-1), q). \tag{3.1}$$

3.2.1 Experience

Experience as described in Section 2.3 contains known quantities. In case of controller tuning it represents the measured data used for model identification. As this work is not about the system identification, the measured data are not written explicitly. They are used to obtain a model of the controlled system $f(y_t|u_t, d(t-1))$ or in another words the pdf of its identified parameters $f(\Theta)$.

3.2.2 Action

The action in case of controller tuning is represented by the particular values of tuning parameter q. The tuning parameter defines the behavior of the controller $f(u_t|d(t-1), q)$ and hence it influences the closed loop. The particular meaning of tuning parameters depends on selected controller, for LQG controller see Chapter 4.

3.2.3 Innovation

The innovation as a quantity measurable after an action is performed is represented by the closed loop data $d(T)$ using controller with particular tuning parameter values. The data measured are in fact a sample of their respective pdf (3.1).

3.2.4 Decision Making

For the purpose of controller tuning, the user's requirements imposed on the desired closed loop behavior are represented by a pair of so called controller quality functions Z_c and Z_o defined on the closed loop data $d(T)$. The first function Z_c represents the constraints imposed on the data. It is a mapping

$$Z_c : d(T)^* \mapsto \mathbf{R}^{\mathring{c}}, \tag{3.2}$$

where $\mathring{c}$ denotes the number of independent constraints. The constraints are considered being met if the expected value of function Z_c is non-positive. The second function Z_o is a mapping

$$Z_o : d(T)^* \mapsto \mathbf{R}. \tag{3.3}$$

It represents a loss function which is decreasing with increasing controller performance with respect to the output error.

The aim of the tuning as finding such a tuning parameter value that satisfies the constraints while maximizes the performance is stated as the following optimization task

$$\begin{array}{rl} \text{minimize} & E[Z_o|q] \\ \text{subject to} & E[Z_c|q] \le 0 \\ \text{over the tuning parameters} & q. \end{array} \tag{3.4}$$

This task corresponds to the Bayesian decision making $q = \mathcal{R}(\mathcal{P})$ as described by (2.6) in Section 2.3

$$\mathcal{R}(\mathcal{P}) = \arg \min_{\mathcal{A} \in \mathcal{A}^*} E[\mathcal{Z}(\mathcal{F}, \mathcal{A}, \mathcal{P})|\mathcal{A}, \mathcal{P}]. \tag{3.5}$$

with the following assignment

- action $\mathcal{A}$ is represented by the searched tuning parameters q
- domain of action $\mathcal{A}^*$ contains all possible actions that satisfy the constraints $\mathcal{A}^* = \{q \ : \ E[Z_c|q] \le 0\}$
- loss function $\mathcal{Z}$ is represented by the function Z_o
- innovation $\mathcal{B}$ is represented by the resulting closed loop data $d(T)$
- experience $\mathcal{P}$ contains identification data and it is not written explicitly.

The controller tuning is a static decision task. There is only one time-independent action—the tuning parameters q and one innovation—the closed loop data $d(T)$. The static decision task is much simpler than the dynamic one described in Section 2.3.3. Thus it can be computed numerically off-line, without strict limitations imposed on the complexity of the model and loss function.

3.3 Construction of Closed Loop

The data pdf $f(d(T)|q)$ is generated by the controller and system model pdf (3.1). Now, these two components are investigated.

3.3.1 System Model

The system model is not known exactly, it is identified from the real data measurement, see Section 2.4.1. The assumed system model is $f(y_t|u_t, d(t-1), \Theta)$. The parameters Θ are obtained from the identification in the form of pdf $f(\Theta)$. In fact the pdf on Θ is conditioned by the measured data but this conditioning is omitted here to avoid confusion with the closed loop data.

The system model respecting the uncertainty of parameters is obtained by the chain rule (2.11)

$$f(y_t|u_t, d(t-1)) = \int f(y_t|u_t, d(t-1), \Theta) f(\Theta) d\Theta. \tag{3.6}$$

The retained uncertainty allows to model more realistic closed loop data behavior than it would be possible with point estimates.

The system model used in this thesis is the ARX model, see Section 2.6. The corresponding pdf $f(y_t|u_t, d(t-1))$ represents the Student's distribution (2.26).

3.3.2 Adaptive Controller

The complexity of the controller depends on the type of controller selected, generally when considering model based controllers the complexity is high and the controller pdf

$$f(u_t|d(t-1), q) \tag{3.7}$$

cannot be obtained in a closed form. Most of known controllers are deterministic ones, with the pdf being the Dirac delta function. The notion of pdf is used because it fits to the Bayesian approach and also a stochastic controller derived by fully probabilistic design [?] exists and it is closely related to the deterministic LQG controller.

The tuning described in this thesis is mainly focused on the adaptive LQG controller. The adaptive controller can be decomposed into a model estimator and a non-adaptive LQG controller, see Figure 3.2.

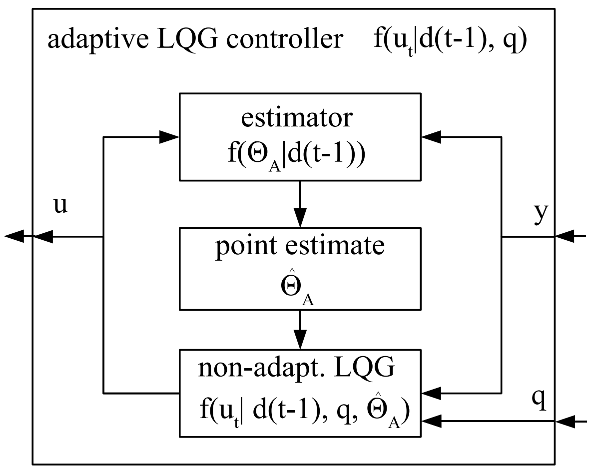

Figure 3.2: Adaptive LQG controller.

The model estimator for the ARX model $f(\Theta_A|d(t-1))$ used within the adaptive controller has the form of Gauss-inverse-Wishart distribution (2.22). The subscript A in Θ_A is used to distinguish it from the system model parameter estimated from the real measured data.

The non-adaptive LQG controller $f(u_t|d(t-1), \Theta_A)$ is derived for known, not uncertain, model parameters. It follows the Bayesian dynamic design, Section 2.3.3, by optimizing the expected value of quadratic criterion defined on specified horizon of the closed loop data. It's calculation can be done only numerically by solving the Riccati equation [**?**]. For more details see Chapter 4.

Due to the fact the non-adaptive LQG controller is derived for the known parameters only, the estimated parameters pdf $f(\Theta_A|d(t-1))$ is approximated by the Dirac delta function obtaining the maximum likelihood estimate $\hat{\Theta}_A$.

3.3.3 Monte Carlo Approach

The apparent complexity of the adaptive LQG controller disallows analytical derivation of the closed-loop data pdf $f(d(T)|q)$, which is necessary for calculation of the expected values $E[Z_o|q]$ and $E[Z_c|q]$. Therefore the optimization task (3.4) must be solved numerically Monte-Carlo. The numerical solution is composed of these three main nested evaluation loops:

- A Numerical optimization of the tuning parameter q
  - B Estimation of the expected values $E[Z_o|q]$ and $E[Z_c|q]$
    - C Simulation loop generating data samples from $f(d(T)|q)$ using the uncertain system model and the adaptive controller

The inner most loop C samples the distribution given by $f(d(T)|q)$ by performing a simulation of length T of the closed loop, Figure 3.1, for each sample. The uncertainty contained in the identified model parameters $f(\Theta)$ is used for the reasonable simulation of the adaptive controller, whose inner estimate of the parameters Θ_A is being adapted to the varying parameters Θ of the simulated system model. This approach allows to tune the adaptive controller off-line using the estimated model from the measured data.

The loop B estimates the expected values $E[Z_o|q]$ and $E[Z_c|q]$ from closed-loop data samples $d(T)$ obtained by repetitive invocation of the simulation loop C with controller tuning parameter set to the value of q. The data samples $\{d(T)_i\}_{i=1}^{N}$ are transformed by the deterministic function of data $Z_o(d(T))$ and $Z_c(d(T))$ and the sample mean is used as the estimate of the searched expected values $E[Z_o|q]$ and $E[Z_c|q]$.

The outer most loop A is an optimization method solving the tuning task (3.4). A numerical method is used. It invokes the loop B for every guess of the possible tuning parameter until it reaches the solution if it exists.

The combination of the three nested loop implies high demands on the computation time of the tuning algorithm. Therefore on-line stopping rules were proposed for the loops B and C to reduce the number of iterations while conserving given precision of the expected value estimation, see Section 3.5.2.

3.3.4 Comparison with the Former Approach

The proposal of the loops configuration given in the previous section gives several advantages in compare to the former approach of the controller tuning.

The controller tuning presented in [12] uses a different approach for the numerical evaluation. The algorithm is based on the following loop configuration

- A Sampling of model parameters Θ and accumulating the particular optimal tuning parameters q
  - B Numerical optimization of the tuning parameter q for given Θ
    - C Loss function expected value is estimated. Simulation generating data sample from $f(d(T)|q)$ using the fixed known Θ for model as well as for the controller, which is non-adaptive.

Note the loops A and B are swapped in contrast to the approach used in this thesis as described in Section 3.3.3. The sampling of Θ is in current algorithm done implicitly during simulating directly from the uncertain system (3.6).

The result of this algorithm is a set of samples of the optimal tuning parameters pdf $f(q)$. The resulting single value from this set is chosen in order to assure the constraints be satisfied for given percentage of the samples. The tuning parameter in this approach is only single-dimensional. It assumes monotonous dependency of the constraint function on the tuning parameter, which is decreasing with increasing tuning parameter value, and also monotonous dependency of the loss function, which is increasing. Thus the optimal tuning parameter can be uniquely chosen by taking appropriate quantil of the approximated pdf $f(q)$ assuring desired probability of constraint satisfaction.

Approach presented in this thesis removes limitations of single-dimensional tuning parameter and single constraint. Even more, for noise compensation ergodic process, the loop performing multiple simulations can be removed and only one long enough simulation run is performed, see Section 3.5.2.

3.4 Closed Loop Performance Evaluation

In this section, requirements and constraints imposed on the ideal closed loop behavior are defined. Their fulfillment is measured by the controller quality functions Z_o and Z_c. The construction of these functions is described.

3.4.1 Loss Function

The control objective expresses commonly the aim assigned to the quality of the regulation process, which should be in a certain sense as good as possible subject to the present constraints, as introduced in Section 3.1.1. The typical wish on the small output error and the control effort of inputs is expressed by the objective function Z_o

$$Z_o = \frac{1}{T}\sum_{\tau=1}^{T}(d_\tau - d_\tau^{\mathrm{ref}})' W (d_\tau - d_\tau^{\mathrm{ref}}), \tag{3.8}$$

where the desired signal setpoints are described by the reference trajectory $\{d_\tau^{\mathrm{ref}}\}_{\tau=1}^{T}$ and a positive semi-definite matrix W of appropriate dimensions.

The matrix W is usually diagonal with only those elements being non-zero which correspond to signals in the data record d_t with an important prescribed reference trajectory or setpoint in d_t^{ref}. The particular values define the cost of particular signal output error.

The elements of matrix W are user's choice, but they do not substitute the proposed tuning algorithm of parameters q. The function Z_o is of a secondary importance as the primary goal of tuning is to satisfy the specified constraint.

A hint for selecting the values of the non-zero elements of W is choice of reciprocal values to the conditional variances of respective signals in d_t for outputs and zero for inputs. This approach puts more importance on tracking of the less noisy channels, while the channels with higher conditional variance take less effort of the controller. The reason is that the variance of the controlled signal in closed-loop cannot be reduced below the conditional variance of the signal in the system model. For more details see Section 4.2.2. The diagonal elements of W corresponding to inputs are left zero, because the inputs are the main concern of the constraint function Z_c.

3.4.2 Constraints

Constraints are often imposed not only on the magnitudes of input and output quantities but also on their dynamic behavior such as limited increments. To cope with these constraints uniformly, a vector variable c_t containing all constrained dynamic expressions of data quantities is introduced.

A vector c_t is extracted from data $d(t)$ by a mapping $\mathcal{C}$

$$\mathcal{C} : d(t)^* \mapsto \mathbf{R}^{\mathring{c}}, \quad \forall t = 1, \ldots, T. \tag{3.9}$$

Using this mapping, the vector c_t can be obtained for the whole time span that is denoted by $c(T) = \{c_t\}_{t=1}^{T}$. The constraints are defined by a set $C \subset \mathbf{R}^{\mathring{c}}$ of allowed values defining the constraint satisfaction in time t by $c_t \in C$.

A common example of independent time invariant constraints is formed by the cartesian product $C = \bigotimes_{i=1}^{\mathring{c}} C_i$ of intervals, where $\mathring{c}$ is dimension of constrained vector c_t. The intervals C_i are defined

$$C_i = \langle c_i^{\min}, c_i^{\max} \rangle \tag{3.10}$$

In the most practical tasks, vector c_t contains magnitudes and increments of data records. The corresponding function $\mathcal{C}$ is

$$c_t = \mathcal{C}(d(t)) = [d_t, d_t - d_{t-1}].$$

The constraint function Z_c introduced in (3.2) is now described using the constraint vectors c_t.

$$Z_c : c(T)^* \mapsto \mathbf{R}^{\mathring{c}}, \tag{3.11}$$

This redefinition does not change the meaning of the function because the constraint variable $c(T)$ is function (3.9) of the data $d(T)$.

Two variants of function Z_c for servo control Z_{c_M} and noise compensation Z_{c_P} tasks are used as described in the rest of this section.

3.4.3 Servo Control Task

The constraint function Z_{c_M} collects information about maximal constraint violation during the simulation run

$$Z_{c_M,i} = \max_{t=1,\ldots,T} \operatorname{dist}(c_{i,t}, C_i) - \operatorname{dist}(c_{i,t}, \operatorname{comp}(C_i)), \tag{3.12}$$

where $\operatorname{comp}(C_i)$ is a set complement of C_i, $Z_{c_M,i}$ is i-th element of Z_{c_M}, and $\operatorname{dist}(x, X)$ denotes a distance between point x and set X. This definition of function is suitable mainly for transient processes, where the constrained signals have one or just a few important peaks, such as servo control tasks. The time T is selected big enough to cover all the instants with significant signal changes.

3.4.4 Noise Compensation Task

The second function Z_{c_P} evaluates proportional amount of time where constraints are violated over the total length of simulation with some allowed tolerance, see Section 3.1.4. In the discrete case, it is the relative frequency of constraint satisfaction

$$Z_{c_P,i} = \alpha_{\min} - \frac{1}{T} \sum_{t=1}^{T} \chi_{C_i}(c_{i,t}), \tag{3.13}$$

where χ_{C_i} is characteristic function of the set C_i, and number $\alpha_{\min} \in \langle 0, 1\rangle$ relaxes the requirement of constraint satisfaction to a specified level.

This definition is suitable for situations where the constraints can be violated any time during the simulation. This is the case of noise compensation control, where the control loop generates an ergodic process. Then it holds

$$Z_{c_P,i} \overset{T\to\infty}{\longrightarrow} \alpha_{\min} - \mathbf{P}(c_i \in C_i) \quad \text{in probability,}$$

where $\mathbf{P}(\cdot)$ denotes probability and c_i has dropped the time index because of ergodicity of the process.

3.5 Numerical Evaluation

In this section, a numerical approach to estimation of expected value from samples is described. The computational complexity is reduced by introducing stopping rules shortening the simulations.

3.5.1 Expected Value Estimation

The controller tuning, formulated as the optimization task (3.4), acts on the conditional expectation of the controller quality functions Z_c and Z_o. However their pdf is not known in a closed form, because of the complexity of the dynamic system model (3.6) and adaptive controller (3.7). Thus the expected value has to be estimated by sampling. To unify the notation in the following text, let $Z_\bullet$ denote all the quality functions distinguished by the content of the placeholder "$\bullet$" for "c_M", "c_P" or "o". The expectation $E[Z_\bullet|q]$ is estimated as sample mean

$$Z_\bullet^N(q) = \frac{1}{N}\sum_{s=1}^{N} Z_{\bullet,s}(q) \overset{N\to\infty}{\longrightarrow} E[Z_\bullet|q], \tag{3.14}$$

Sequence $\{Z_{\bullet,s}(q)\}_{s=1}^N$ denotes N samples of $Z_\bullet$ from $f(Z_\bullet|q)$.

3.5.2 Number and Length of Simulations

The quantity $Z_\bullet^N$ is evaluated using N independent simulation runs. The length of each run is determined by T. Increasing these two numbers N and T increases precision of the expected value approximation $Z_\bullet^N$ of the controller quality functions. On the other hand, it also increases the computational demands of the evaluation, thus the lengths have to be limited. To solve this tradeoff, the on-line stopping rules are employed. First, the properties of the quality functions with respect to number and length of simulations are described.

The variance of the quantity $Z_\bullet^N$ is indirectly proportional to the number of independent simulation runs N, which is clear from its evaluation (3.14).

The similar situation occurs for length of simulation T, which has to be long enough in order to:

1. Contain all important reference trajectory changes.
2. Allow the uncertain parameters to vary in order to simulate the controller adaptiveness.
3. Decrease the variance of the controller quality functions.

The item 1 is straightforward. It is used for transient processes, where a kind of constraint measure Z_{c_M} is used. Of course, all responses related to reference trajectory changes have to be included, too.

The situation of items 2 and 3 is more complicated. Both of the items contribute to the precision of the expected value estimate. Even more, the item 2 can be substituted by item 3, because if the variance is low, it means that further parameters changes bring no more information on the controller quality functions.

Increasing the simulation length T for the ergodic case, such as the noise compensation, has the same effect as increasing the number N of the simulations. Thus, one long simulation is sufficient.

The proper values of N and T are decided on-line during simulation using the Chebyshev inequality and the Kullback-Leibler divergence. The on-line stopping is advantageous in comparison with the off-line determination of the length and number of simulations, because it considers the contribution of the actual data and thus stopping is optimal for the current simulation unlike for all possible simulation runs as in the case of a priori selected N and T values.

3.5.3 On-line Stopping Rule for Number of Simulations

The independent simulation runs are connected mainly with non-stationary servo-control tasks. It is hard to find a reasonable distribution of the quality functions $Z_\bullet$ for different variants of reference trajectory. Thus, a simple non-parametric stopping rule is used. It is activated when the following inequality is satisfied

$$\mathbf{P}(|Z_\bullet^N - EZ_\bullet^N| \geq \gamma) \leq \beta, \tag{3.15}$$

where parameters β and γ determine the sensitivity of the stopping.

The stopping is based on variance of $Z_\bullet^N$ as shown below. The independency of averaged quality functions (3.14) resulting to $Z_\bullet^N$ used with Chebyshev inequality yields

$$\mathbf{P}(|Z_\bullet^N - EZ_\bullet^N| \geq \gamma) \leq \frac{\mathrm{var}(Z_\bullet^N)}{N\gamma^2}. \tag{3.16}$$

As covariance $\mathrm{var}(Z_{\bullet}^{N})$ is unknown, its estimate $Z_{\sigma,\bullet}^{N}$ is used

$$Z_{\sigma,\bullet}^{N} = \sum_{s=1}^{N} \frac{(Z_{\bullet,s})^2 - (Z_{\bullet}^{N})^2}{N}, \tag{3.17}$$

where variable $Z_{\bullet,s}$ has the same meaning as in (3.14). Then the stopping is triggered after certain minimal number of simulations is performed and when the following inequality is satisfied

$$\frac{Z_{\sigma,\bullet}^{N}}{N\gamma^2} \leq \beta. \tag{3.18}$$

3.5.4 On-line Stopping Rule for Simulation Length

A rule for on-line simulation stopping for the noise compensation task is described here. The function $Z_{\bullet}$ contains a sum (3.8) or (3.13), but the summed terms are correlated, so the approach using the Chebyshev inequality from Section 3.5.3 cannot be applied. Let the summed terms (3.23) of Z_o and $\chi_{C_i}(c_{i,t})$ of Z_{c_P} forming the controller quality functions be called partial controller quality and be denoted by v_t. For the noise compensation task we assume the closed loop signals be ergodic and thus also the partial losses are ergodic.

To find a reasonable stopping rule, a simple dynamic model of v_t

$$f(v_t|v(t-1), \Xi). \tag{3.19}$$

is being estimated in Bayesian way. Let the parameters of the model be denoted by Ξ. When the estimated pdf $f(\Xi|v(t))$ of the model parameters Ξ stabilizes, the stopping takes place. The stabilization of pdf $f(\Xi|v(t))$ is measured by the Kullback-Leibler divergence $\mathcal{D}_{\mathrm{KL}}$ of two successive pdf estimates [**?**]. It is defined by

$$\mathcal{D}_{\mathrm{KL}}(f(\Xi|d(T))\|f(\Xi|d(T-1))) = \int f(\Xi|d(T)) \ln \frac{f(\Xi|d(T))}{f(\Xi|d(T-1))} d\Xi. \tag{3.20}$$

When this divergence, labeled U_T, becomes smaller than some threshold value ε

$$U_T = \mathcal{D}_{\mathrm{KL}}(f(\Xi|d(T))\|f(\Xi|d(T-1))) \leq \varepsilon, \tag{3.21}$$

the computation is stopped. At this moment T, the pdf $f(\Xi|d(T))$ is considered to reach the steady state. The stationarity means that more data would not bring significantly more information for the estimate. The dynamic model of variable v (3.19) is used just for determination of the stopping time while the loss function is calculated by its original defining equation (3.8) or (3.13). This approach was mentioned in [**?**].

Yet there is a better opportunity of calculating the loss function value from the estimated dynamic model of the partial loss v by evaluating its stationary pdf. This approach is used in the next paragraph with ARX model. It is shown that $f(Z|d(T))$ is stabilizing as $f(\Xi|d(T))$ is stabilizing. In other words the divergence

$$\mathcal{D}_{\mathrm{KL}}(f(Z|d(T))\|f(Z|d(T-1))) \tag{3.22}$$

is decreasing as $\mathcal{D}_{\mathrm{KL}}(f(\Xi|d(T))\|f(\Xi|d(T-1)))$ is decreasing.

The definition of the quantity v_t and the construction of the particular models for the functions Z_o and Z_{cP} is described in the following paragraphs using ARX and Markov chain models. The stopping rule for whole simulation is triggered when the conditions for both loss and constraint function stopping are activated.

Approximation by ARX Model

This section describes a suitable model type (3.19) of the partial quality v_t used for determination of the stopping time when evaluating the loss function Z_o. The quantity v_t for the function Z_o as the summed term in (3.8) is the weighted distance between the data variable d_t and its referential value d_t^{ref} in time t

$$v_t = (d_t - d_t^{\mathrm{ref}})' W (d_t - d_t^{\mathrm{ref}}). \tag{3.23}$$

For purpose of stopping quite a rude dynamic approximation of v_t by a simple autonomous ARX model is used.

$$v_t = a v_{t-1} + k + e_t, \quad e_t \sim \mathcal{N}(0, R). \tag{3.24}$$

The parameters a, k, and R are collected into the variable Ξ, where $[a,k] = \Xi_\theta$, $R = \Xi_R$ and $\Xi = [\Xi_\theta, \Xi_R]$.

The Bayesian identification of the parameters Ξ leads to the self reproducing Gauss-inverse-Wishart prior/posterior pdf

$$\begin{aligned} f(\Xi|v(t)) &= f(\Xi_\theta, \Xi_R|V_t, \nu_t) = \\ &= \alpha_t |R|^{-\frac{\nu_t}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\left(\Xi_R^{-1}\begin{bmatrix} -I \\ \Xi_\theta \end{bmatrix}' V_t \begin{bmatrix} -I \\ \Xi_\theta \end{bmatrix}\right)\right\}, \end{aligned} \tag{3.25}$$

where α_t is a normalizing constant. Statistics ν_t and V_t and parameter elements Ξ_θ and Ξ_R written only as θ and R without Ξ are described in Section 2.5.

The stationarity measure for the Z_o function denoted by $U_{o;t}$, by means of the Kullback-Leibler divergence of two successive estimated pdfs of Ξ, has the form [**?**]

$$U_{o;t} = \mathcal{D}_{\mathrm{KL}}(f(\Xi|v(t))\|f(\Xi|v(t-1))) = \frac{F(\nu_t) + G(\zeta_t) + H(\nu_t, \varrho_t, \zeta_t)}{2}, \tag{3.26}$$

where

$$
\begin{aligned}
F(\nu_t) &= 2\ln\left(\Gamma\left(\frac{\nu_t - 1}{2}\right)\right) - 2\ln\left(\Gamma\left(\frac{\nu_t}{2}\right)\right) + \frac{\partial \ln(\Gamma(\frac{\nu_t}{2}))}{\partial \frac{\nu_t}{2}} \\
G(\zeta_t) &= \ln(1+\zeta_t) - \frac{\zeta_t}{1+\zeta_t} \\
\varrho_t &= \frac{\hat{e}_t^2}{D_{y,t-1}(1+\zeta_t)} \\
H(\nu_t, \varrho_t, \zeta_t) &= (\nu_t - 1)\ln(1+\varrho_t) - \frac{\nu_t \varrho_t}{(1+\varrho_t)(1+\zeta_t)}.
\end{aligned}
$$

The quantities ζ_t, $\hat{e}_t$, and $D_{y,t-1}$ are defined in Section 2.6.

When the divergence $U_{o;T}$ is less than threshold ε in time T then it is assumed that enough information has been collected and the loss function Z_o (3.8) is precise enough.

Loss Evaluation from Dynamic Model It is possible to evaluate the mean value of loss function Z_o directly from the dynamic stopping model of v_t (3.19) instead of its original definition (3.8), were the stabilization property of EZ_o is implied by stabilization of the dynamic model parameters.

This is obtained by transforming the dynamic model (3.19) into a static one. First, the transformation for deterministic parameters Ξ is given and then the distribution of uncertain ones is transformed.

Suppose now that the parameters a, k, R of dynamic model (3.19) are known and stable, $|a| < 1$, then the corresponding static model is given by pdf

$$v_t = \mathcal{N}(p, q), \tag{3.27}$$

where parameters p, q are given by

$$p = \frac{k}{1-a} \tag{3.28}$$

$$q = \frac{R}{1-a^2} \tag{3.29}$$

The new parameter p is a suitable estimate of Z_o, as $Z_o = \frac{1}{T}\sum_{t=1}^{T} v_t$. Thus

$$p = Ev_t \doteq EZ_o,$$

where the $\doteq$ sign means approximately equal as the stopping model (3.24) is just an approximation.

If the model (3.19) is unstable, $|a| \geq 1$, the loss Z_o is infinity.

Now we drop the assumption of certain parameters. As the model (3.19) is estimated in Bayesian way, its parameters are uncertain. Thus parameters p and q of the static model (3.27) are uncertain, too. The estimate of Z_o is therefore selected as expected value of p

$$EZ_o \doteq Ep \tag{3.30}$$

The pdf of p is obtained by transforming quantities k, a and R according to (3.28). Unfortunately, the posterior pdf of parameters Ξ is Gauss-inverse-Wishart and it has infinite support for parameter $\Xi_a = a$. Situation when $|a| \geq 1$ and the estimated model (3.24) is unstable has non-zero probability. This conforms to the reality where a system model with uncertain parameters connected in closed loop can be with some probability unstabilizable.

As this situation is generally unavoidable, we have to accept that the stopping model (3.24) is unstable with some low probability $\mathbf{P}(|a| \geq 1)$. However this makes the estimate of EZ_o infinite. When evaluating the Z_o directly from simulation by (3.8) and the closed loop shows to be unstable, the estimated model is rejected by the tuning algorithm. Thus the results are limited to the stabilizable models only. So when we approximate the EZ_o from stable stopping models only $|a| < 1$ we obtain the same result. Therefore we may restrict the transformation (3.28) to $|a| < 1$.

Stopping properties of transformed quantity The stopping property with respect to the Kullback-Leibler divergence of parameter pdf $f(\Xi)$ (3.21) implies the same property for transformed quantity p, which is used to estimate EZ_o (3.30).

$$\mathcal{D}_{\text{KL}}(f(p|d(T))\|f(p|d(T-1))) \leq \mathcal{D}_{\text{KL}}(f(\Xi|d(T))\|f(\Xi|d(T-1))) \leq \varepsilon$$

This can be proven by writing the transformation (3.28) restricted on $|a| < 1$, lets denote it G, as a composition $G = S \circ H$ of regular transformation $H : p = \frac{k}{1-a}$, $k = k$ and projection S selecting only element p from result of S.

The Kullback-Leibler divergence remains unchanged when transforming the quantity by a regular transformation. Let $f(x)$ and $g(x)$ be pdfs on quantity x. Transformed quantity $y = H(x)$ has pdf $\tilde{f}(y) = f(H^{-1}(y))|J_{H^{-1}}(y)|$ and similarly for pdf g. Then it holds

$$\mathcal{D}_{\text{KL}}(\tilde{f}(y)\|\tilde{g}(y)) = \int f(H^{-1}(y))|J_{H^{-1}}(y)| \ln \frac{f(H^{-1}(y))}{g(H^{-1}(y))} dy = \\ = \int f(x) \ln \frac{f(x)}{g(x)} dx = \mathcal{D}_{\text{KL}}(f(x)\|g(x)).$$

The projection transformation decreases the value of the Kullback-Leibler divergence

$$\mathcal{D}_{\text{KL}}(f(a)\|g(a)) \leq \mathcal{D}_{\text{KL}}(f(a,b)\|g(a,b)) \tag{3.31}$$

This is proven by

$$\mathcal{D}_{\text{KL}}(f(a,b)\|g(a,b)) = \\ = \int\int f(a,b) \ln \frac{f(a,b)}{g(a,b)} da db =$$

$$
\begin{aligned}
&= \int f(a) \int f(b|a) \ln \frac{f(b|a) f(a)}{g(b|a) g(b)} db da = \\
&= \int f(a) \ln \frac{f(a)}{g(a)} da + \int f(a) \int f(b|a) \ln \frac{f(b|a)}{g(b|a)} db da = \\
&= \mathcal{D}_{\text{KL}}(f(a) \| g(a)) + \int f(a) \mathcal{D}_{\text{KL}}(f(b|a) \| g(b|a)) da \geq \mathcal{D}_{\text{KL}}(f(a) \| g(a))
\end{aligned}
$$

Approximation of Transformed Expected Value Applying Taylor series expansion of transformation $p = G(\Xi)$ in point $E\Xi$ we obtain

$$
p = G(E\Xi) + (\Xi - E\Xi)\nabla G(E\Xi) + \frac{1}{2}(\Xi - E\Xi)\nabla^2 G(E\Xi)(\Xi - E\Xi)' + \cdots, \quad (3.32)
$$

which is in expected value

$$
Ep = EG(\Xi) = G(E\Xi) + \frac{1}{2}\text{tr}(\nabla^2 G(E\Xi)\text{cov}\Xi) + \cdots. \quad (3.33)
$$

Using just the first order approximation we obtain

$$
Ep \doteq \frac{Ek}{1 - Ea}
$$

The expected value Ea should be evaluated from the distribution with support only on $|a| < 1$, as described above. Nevertheless, the probability $\mathbf{P}(|a| > 1)$ is low. So using the expected value from original Gauss-inverse-Wishart distribution on Ξ is sufficient. This approximation gives good results comparing to calculation directly from definition of Z_o (3.8) according to experimental testing.

Markov Chain Estimation

The calculation of the constraint function Z_{c_P} (3.13) includes an estimate of constraint satisfaction probability using characteristic function of the allowed set. To determine precision of this estimate, the task is slightly extended.

Given the i-th element of the constraint quantity $c_{i;t}$ from Section 3.4.2 and the corresponding constraining interval C_i from (3.10), let $\{v_t\}_{t=1}^{T}$ be a sequence indicating the relative position of $c_{i;t}$ to C_i

$$
v_{i;t} = \begin{cases} 1 & c_{i;t} > C_i \\ 0 & c_{i;t} \in C_i \\ -1 & c_{i;t} < C_i \end{cases}, \quad (3.34)
$$

where the inequality symbol is understood as it holds for all the elements of the interval on its right side.

The dynamic model (3.19) of the discrete variable $v_{i;t}$ is represented by Markov chain

$$f(v_{i;t}|g_{i;t-1}, \Xi) = \Xi_{v_i|g_i}, \text{ where } \Xi_{v_i|g_i} \geq 0 \text{ and } \sum_{v_i} \Xi_{v_i|g_i} = 1. \tag{3.35}$$

The notation of $\sum_{v_i}$ denotes a sum over the whole set of possible values v_i^*, the analogous situation holds also for the product in the following text. As the quantity v_t is now discrete, the symbol f represents a probability function now. The quantity $g_{i;t-1}$ contains the past values of $v_{i;t}$

$$g_{i;t-1} = [v_{i;t-1}, v_{i;t-2}, \ldots, v_{i;t-\eta}].$$

The number η denotes the order of the Markov chain. The parameter $\Xi_{v|g}$ has $3^{\eta+1}$ entries. The following derivations are done for single element of v_t only and the element index i is omitted for the sake of simplicity.

Using the Bayes' rule and the conjugated prior on $f(\Xi)$ defined by the statistic $V_{0,v|g}$

$$f(\Xi) \propto \prod_g \prod_v \Xi_{v|g}^{V_{0,v|g}-1},$$

we obtain the posterior pdf of the parameters Ξ

$$f(\Xi|v(t)) = \frac{\prod_g \prod_v \Xi_{v|g}^{V_{v|g;t}-1}}{B(V_t)},$$

where

$$V_{v|g;t} = V_{0,v|g} + \sum_{\tau=1}^{t} \delta(v, v_t)\delta(g, g_t)$$

with $\delta(\cdot,\cdot)$ being the Kronecker delta and the normalizing factor

$$B(V_t) = \prod_g \frac{\prod_v \Gamma(V_{v|g;t})}{\Gamma(\sum_v V_{v|g;t})}.$$

The stopping rule uses the Kullback-Leibler divergence to determine if there is collected enough information about the constraint function Z_{c_P}. The calculation is stopped whenever the divergence of two successive pdfs, denoted by $U_{c;T}$, is less or equal to threshold ε

$$U_{c;T} = \mathcal{D}_{\mathrm{KL}}(f(\Xi|v(T)) \| f(\Xi|v(T-1))) \leq \varepsilon. \tag{3.36}$$

Derivation of this divergence for the Markov chain model is done through converting it to the Dirichlet model, for which the divergence is analyzed in [?].

Parameters $\Xi_{v|g}$ are independent for different past data g. Thus

$$f(\Xi|v(t)) = \prod_g f(\Xi_{\bullet|g}|v(t)),$$

where the particular factors

$$f(\Xi_{\bullet|g}|v(t)) = \frac{\Gamma(\sum_v V_{v|g;t})}{\prod_v \Gamma(V_{v|g;t})} \prod_v \Xi_{v|g}^{V_{v|g;t}-1}$$

are distributed by the Dirichlet distribution. In each time step, only one of these factors is updated—that one with corresponding past data $g = g_{t-1}$. The other factors remain unchanged.

As it holds

$$\mathcal{D}_{\mathrm{KL}}(f_1(x)f(y)\|f_2(x)f(y)) = \mathcal{D}_{\mathrm{KL}}(f_1(x)\|f_2(x)),$$

thus

$$\mathcal{D}_{\mathrm{KL}}(f(\Xi|v(t))\|f(\Xi|v(t-1))) = \mathcal{D}_{\mathrm{KL}}(f(\Xi_{\bullet|g_t}|v(t))\|f(\Xi_{\bullet|g_t}|v(t-1))) \quad (3.37)$$

is a divergence of two Dirichlet distributions. Now, the divergence of the two Dirichlet distributions derived in [?] can be used in (3.37) and the stopping rule (3.36) then yields

$$U_{c;t} = -\ln \frac{V_{v_t|g_t;t-1}}{\sum_v V_{v|g_t;t-1}} + \frac{\partial}{\partial V_{v_t|g_t;t}} \ln \Gamma\left(V_{v_t|g_t;t}\right) - \frac{\partial}{\partial \sum_v V_{v|g_t;t}} \ln \Gamma(\sum_v V_{v|g_t;t}). \quad (3.38)$$

At the stopping time T, determined by (3.36), a stabilized MC model is obtained.

For the stopping purposes only first order, $\eta = 1$, Markov chain is used. Its steady state probability $\mathbf{P}(v_t = 0)$ of state number zero in (3.34) can be used for obtaining the value of Z_{c_P}. The steady state p evaluation for Markov chain with certain parameters Ξ requires calculation of vector p such that $\sum_{i=1}^{3} p_i = 1$ and $\Xi p = p$.

For uncertain Ξ with Dirichlet distribution it is difficult to calculate the distribution of steady state p. Also another problem arises when using smooth optimization technique for constraints satisfaction measure evaluated from only finite number of samples, see Section 3.6.2. Thus the Markov chain is evaluated only for stopping purposes.

Properties of the Stationarity Measures Illustrated on an Example

To show the properties of the stationarity measures $U_{c;t}$ (3.38) and $U_{o;t}$ (3.26) using ARX and MC stopping models, a simple illustrative experiment is presented. The results can be seen in Figure 3.3. The data used for the

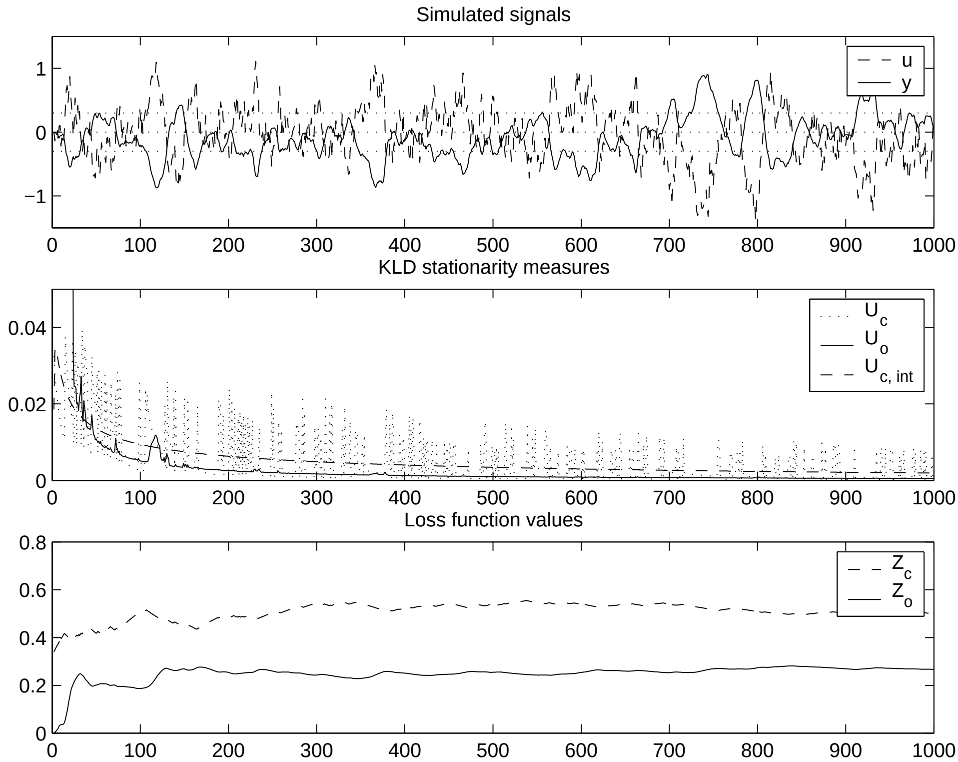

Figure 3.3: Properties of stationarity measures. The symbols u and y denote inputs and outputs, U_c and U_o are the stationarity measures obtained by Markov chain and logarithm ARX model approximations. U_c^{int} is interpolation of U_c. Z_c and Z_o are controller quality functions of constraint violation and output error. Horizontal axis represents the time.

evaluation of the loss and constraint functions were generated using the linear system with transfer function

$$\frac{0.00468 + 0.00438z^{-1}}{1 - 1.81z^{-1} + 0.8178z^{-2}}$$

which was driven by zero mean white noise with variance one. This model was obtained by discretization of a simple continuous model with transfer function

$$\frac{1}{(1+s)^2}$$

with sampling period 0.1.

The squares of the generated output samples were used as a partial quality function for the stopping by using ARX model stabilization, see (3.23) with $W = 1$. The stationarity measure $U_{o;t}$ for the ARX model is seen in the second part of the figure and the evolution of the mean value estimation is in its third part.

The constraining interval $[-0.3, 0.3]$ is used on the generated data to obtain the discrete three-state indicator (3.34) for the purpose of stopping through the MC model (3.35). The resulting stationarity measure $U_{c;t}$ and the corresponding estimation of probability of the state zero are shown in the second and third part of the figure.

It can be seen that the measure $U_{c;t}$ is rather fuzzy. This complicates the decision whether to stop simulation, because the rule to stop whenever the measure is below the threshold is quite unsatisfactory as the several next samples immediately increase the value above this threshold. To solve this problem, an interpolation is performed using the approximation by the following model

$$U_{c,t}^{\text{int}} = a_0 + a_1 t^{-1/2} + a_2 t^{-1}, \tag{3.39}$$

were the coefficients are obtained by linear regression. The interpolated measure, denoted by $U_{c;t}^{\text{int}}$, is shown in the figure. The interpolation is, up to a tiny peak close to the origin, satisfactory for the stopping purposes.

It is possible to think about stopping for the interpolating regression, too, and trigger the stopping when the interpolated measure is below the threshold as well as the interpolation itself has been stabilized.

The threshold for the measures $U_{c;t}^{\text{int}}$ and $U_{o;t}$ need not be of the same value. The stopping models are different and have a different number of identified parameters. From the particular example presented in this section a reasonable threshold for $U_{c;t}^{\text{int}}$ is roughly 0.008 and for $U_{o;t}$ it is 0.004.

3.6 Optimization

The controller tuning as the optimization task (3.4) requires to evaluate expected values of the controller quality functions. These functions are data-dependent and the data are random quantities whose distribution is available through samples only. As the expected value is estimated from a finite number of samples, it is also a random quantity. The optimization (3.4) is a stochastic constrained optimization task, which is generally difficult to solve. From possible approaches to the solution of this kind of optimization we tried two of them. The stochastic approximation [13] and the sample path method [11]. We found from the experiments that the sample path method is much more suitable for this controller tuning in a sense of convergence speed. We describe it here briefly.

3.6.1 Sample Path Method

Suppose we are solving the controller tuning task for an identified system model with given quality functions. The values, or more precisely samples, of the quality functions depend on data, which in turn depend on the tuning parameter value and a particular randomness realization used in the simulation. The expected values of the quality functions are approximated by sample mean as described in (3.14). Now we rewrite this formula with different quantities. The length of the simulation T is determined by the stopping rule and so it is not given explicitly. The additional quantity is a sequence of independent identically distributed quantities $\xi = \{\xi_i\}_{i=1}^{N}$ that covers the randomness used during the i-th simulation run. The rewritten formula

$$\hat{Z}_{\bullet}^{N}(q,\xi) = \frac{1}{N}\sum_{i=1}^{N} Z_{\bullet}(q,\xi_i), \tag{3.40}$$

where the quality function $\hat{Z}_{\bullet}^{N}(q,\xi)$ is now deterministic, because everything random is covered by ξ. In other words the data $d(T)$ are a deterministic function of the tuning parameters q and the randomness ξ_i, thus the quality function sample defined by $Z_{\bullet}(q,\xi_i) = Z_{\bullet}(d(T))$, is also deterministic. Note that the samples of the quality function in (3.14) are the same but with unspecified randomness.

When the N is going to infinity, the limit $\hat{Z}_{\bullet}^{\infty}(q)$ does not depend on the particular randomness ξ any more. This fact is used by the sample path method, where the random sequence ξ is sampled only once at the beginning of the optimization and then it remains fixed when evaluating (3.40) for all possible tuning parameter values q. The expected value $\hat{Z}_{\bullet}^{\infty}(q)$ is still approximated by $\hat{Z}_{\bullet}^{N}(q,\xi)$, where N is high enough to obtain good approximation of the expected value. The number of samples is obtained by the stopping rule described in Section 3.5.3. Because of the fixed sequence ξ it is possible to use the well developed deterministic optimization methods. This is advantageous especially for the constrained case that we solve. For details on properties of the sample path method see [11].

The randomness in the case of controller tuning takes place in the system model pdf sampling (3.6). A possible form of the randomness ξ is a sequence of independent samples of the uniform distribution $\mathcal{U}(0,1)$. The pdf (3.6) conditioned by actual data is then obtained by respective transformation of particular element of ξ.

The particular solution of the randomness fixing for the case of ARX model with uncertain parameters with modeled pdf (2.23) is to us the approximation using the normal distribution (2.26). The sequence ξ can be now based on the normal distribution $\mathcal{N}(0,1)$ which is scaled and shifted to match the required normal distribution (2.26). An easy way of algorithmic implementation of the sample path method is to set the state of the pseudo-random number generator to a predefined value in the beginning of every simulation.

A deterministic quasi-newtonian optimization method called `fmincon` was selected from the Matlab Optimization toolbox [9]. The used optimization method is a local one and therefore it depends on the starting point. The choice of the starting point is dependent on the type of controller used and its tuning parameters. For the LQG controller the approximation of the searched tuning parameter values that can serve as the starting point is described in Sections 4.2.1 and 4.2.2.

3.6.2 Constraint Function Smoothing

The selected quasi-newtonian optimization method requires smooth loss and constraint functions. But it is not always met in the case of numerical approximation of these functions. This section describes a solution of the problems caused by non-continuity of the approximated constraint function $Z_{c_P,i}$.

A sample of i-th element of random quantity $Z_{c_P,i}$ (3.13), evaluated from simulation, obtains only a finite number of discrete values

$$Z_{c_P,i} \in \{\alpha_{\min} - \frac{j}{T} : j \in \{0, 1, \ldots, T\}\},$$

because there is the sum of finite number of samples evaluated by characteristic function in (3.13). Thus the value of $Z_{c_P,i}$ as a function of the tuning parameter q for the deterministic simulation with fixed randomness is piecewise constant. Even the expected value estimation (3.40) calculated from N sample simulations does not help, because it only extends the number of possible discrete values. Therefore it is difficult to calculate the improvement of small perturbation of tuning parameters to the constraint satisfaction. This fact kills the numerical approximation of gradient used by the deterministic optimization method.

To eliminate the discrete valued function $Z_{c_P,i}$ a continuous piecewise linear interpolation $\tilde{c}_{i,t}$ of the original constrained signal $c_{i,t}$ on the discrete time span $t = 1, \ldots, T$ is used instead. The measure of the constraint satisfaction is obtained as a volume of the continuous time where the interpolation $\tilde{c}_{i,t}$ lies inside the bounds. The interpolation is defined as

$$\tilde{c}_{i,t} = \sum_{k=1}^{T-1} (c_{i,k} + (c_{i,k+1} - c_{i,k})(t - k))\chi_{[k,k+1)}(t),$$

where each summed term represents the affine combination of two adjacent discrete values of $c_{i,k}$ and $c_{i,k+1}$. Each affine combination is limited to the particular interval by the characteristic function $\chi_{[k,k+1)}(t)$ of continuous time in the half-closed interval $[k, k+1)$. The sum just puts these particular interpolation into a series. See Figure 3.4.

Accumulation of ratio $Z_{\tilde{c}_P,i}$ of continuous time variable $\tilde{c}_{i,t}$ where constraints are satisfied, see the grayed intervals in Figure 3.4, is evaluated from

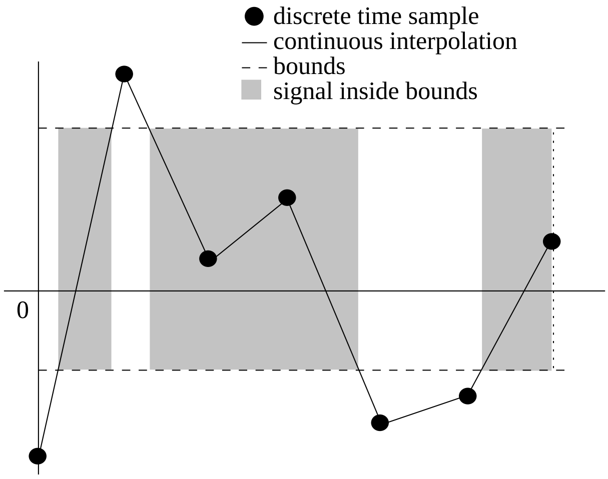

Figure 3.4: Linear interpolation of discrete signal.

the original discrete time samples $c_{i,t}$ by

$$Z_{\tilde{c}_P,i} = \alpha_{\min} - \frac{1}{T-1}\sum_{t=1}^{T-1} w_{i,t},$$

where the non-integer relative indicator $w_{i;t}$ of constraint satisfaction of continuous interpolated variable $\tilde{c}_i$ in the particular interval between two discrete samples $[t, t+1)$ evaluates to

$$w_{i,t} = \begin{cases} \frac{\max(c_i^{\min},\min(c_i^{\max},c_{i,t+1}))-\max(c_i^{\min},\min(c_i^{\max},c_{i,t}))}{c_{i,t+1}-c_{i,t}} & \text{for } c_{i;t} \neq c_{i;t+1} \\ 1 \quad \text{for } c_{i;t} = c_{i;t+1} \in C_i \\ 0 \quad \text{for } c_{i;t} = c_{i;t+1} \notin C_i \end{cases}$$

The fraction in the first case measures the “vertical” interval of the line interpolation limited to the bounds and divided by the total unlimited “vertical” interval length. Since the interpolation is linear, this ratio is the same as length of the “horizontal” interval, which represents the amount of time where the interpolation lies inside the bounds between the particular integer time instants. The first case covers all possible adjacent constraint values combination up to a situation where both are of the same value, this situation is covered by the second and third case.

3.6.3 Handling Quality Functions Shape

The two previous sections solved problems with randomness and non-smoothness of the quality functions Z_o and Z_c for purpose of quasi-newtonian constrained optimization which is suitable for deterministic and smooth functions. Still there are some adverse situations making the optimization method difficult to succeed.

Shape

The typical shape of the quality functions Z_o and Z_c for a case of noise compensation task is shown in Figure 3.5 for the case of one-dimensional tuning parameter and one constraint. The function Z_o is low close to the origin because the tuning parameter as a penalization weight for small values enables the controller to perform intensive control actions to compensate the disturbance well. Therefore the function Z_c is high close to the origin. As the penalization increases the control actions are attenuated so the function Z_c decreases and Z_o increases as the disturbance is less compensated. At some point the function Z_c becomes non-positive and from this point towards bigger penalization the constraint is satisfied. This situation in Figure 3.5 was shown for system with low parameter uncertainty for that the LQG controller is capable to overcome this uncertainty without loss of stability.

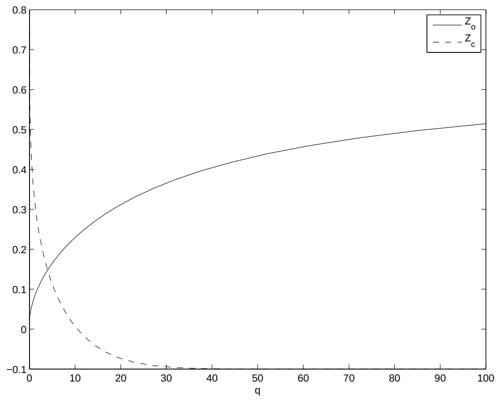

Figure 3.5: Closed loop quality Z_o and Z_c for SISO case depending on tuning parameter q for system with low model parameter uncertainty

In Figure 3.6 there is the same situation drawn where the parameter uncertainty is high. For big penalization values the situation is almost same as in Figure 3.5 but for small penalization the controller–system model mismatch leads to worse noise compensation up to a loss of closed-loop stability close to the origin. This property is visible on the increase of function Z_o close to the origin in comparison with figure 3.5. The function Z_o is non-monotonous in this case. The effect on function Z_c also increases its value close to the origin but this function remains monotonously decreasing.

The closed loop results for two-dimensional penalization with low parameters uncertainty are shown in Figure 3.7.

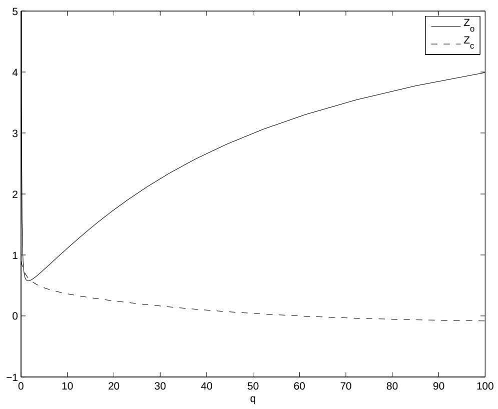

Figure 3.6: Closed loop quality Z_o and Z_c for SISO case depending on tuning parameter q for system with high model parameter uncertainty

Flatness Problem

A difficulty for the optimization method takes place in region far from the origin where both the quality functions Z_o and Z_c are very flat. The imprecision caused by iterative evaluation of the loss functions comparing to the low gradient is high enough to disable the gradient approximation by finite perturbations. As a result the quasi-newtonian method fails to get out of this flat region.

Since the shape of the quality functions is not known a priori, it is not possible to avoid the optimization method to get to this flat region. Even starting from a point very close to the origin does not solve this problem, because the optimization method often makes the first step rather big, thus ending far from the origin again. Also the problem of instability for models with higher parameter uncertainty and low control action penalization, see Figure 3.6, complicates the selection of too low penalizations.

Suitable Starting Point

To avoid the problem in the flat region a simple preliminary optimization is done to find a good starting point for the quasi-newtonian constrained optimization. The preliminary optimization is based on the monotonicity property of the function Z_c and it finds a zero crossing point of this function on a single dimensional set given by a ray starting from the origin and crossing a given guess of the suitable tuning parameter value, for LQG controller initial approximation of tuning parameter see Section 4.2.

The zero finding on a single-dimensional space for monotonous function is done without need of gradient evaluation thus avoiding possible problems in the flat regions.

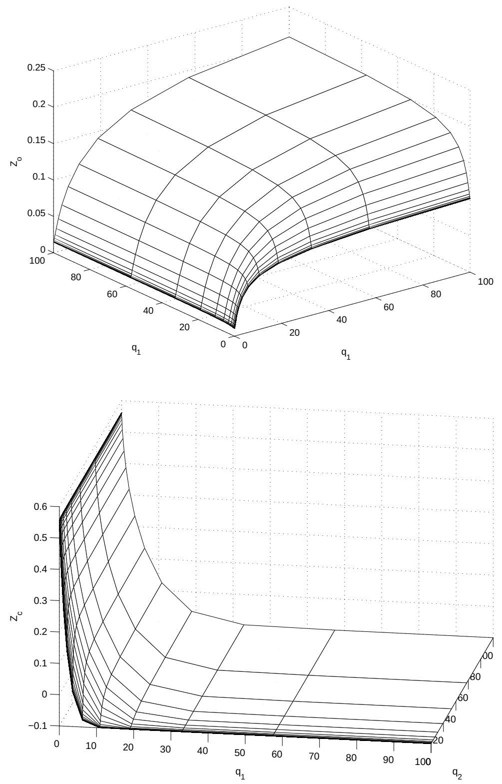

Figure 3.7: Closed loop quality Z_o and Z_c for MIMO case with two dimensional tuning parameter q

At the found zero-crossing point of function Z_c the constraints are satisfied, thus the optimization method does not attempt to make far step towards flat regions. This point is also used as a hint for size of finite perturbations for gradient estimation to make it big enough with respect to the calculation errors and small enough to obtain the good gradient approximation. It is chosen as $1/10$ of the norm of the zero-crossing point.

The one-dimensional zero-crossing search is faster than the multidimensional optimization, thus it saves the much of the computation time and the multidimensional optimization is run finally to find the optimum starting from the rude approximation.

3.7 Controller Tuning Implementation

During the work on this topic the software implementation of the tuning was developed. In this section, the algorithm of the tuning is presented first and then the whole composition of the Matlab toolbox Designer including its integration with the other toolboxes the Mixtools and Jobcontrol is described briefly.

3.7.1 Controller Tuning Algorithm

The particular methods used for the task of controller tuning described in this chapter are now summarized and ordered in the form of an illustrative algorithm of possible implementation for the ARX system model. However the algorithm is proposed for a general controller, this thesis is concerned in the LQG controller, see Section 4. Some remarks in the algorithms refers explicitly to the LQG controller.

The optimization and evaluation loops are described only, as they are the main topic of this work. The results of the system identification are supposed to be already available from the toolbox Mixtools, for details see for example [**?**].

The algorithm follows:

1. Obtain the task specification
   - Model parameters pdf $f(\theta, R|V_0, \nu_o)$, see (2.22). Known from the identification.
   - Desired reference trajectory and the respective loss function Z_o. Given by the user.
   - Constraints imposed on input signals and the respective constraint function Z_c, see Section 3.4. Given by the user.
2. Select initial estimate of tuning parameter value q for particular type of controller used. The initial estimates for the case of LQG controller are described in Section 4.2.
3. Optimization loop
   (a) Initialize statistics Z_c^N and Z_o^N of multiple simulations by zeros for the summation (3.14).
   (b) Initialize pseudo-random number generator by a fixed value in order to assure fixed randomness ξ, see Section 3.6.
   (c) Multiple simulation loop
      i. Initialize simulation
         - Initialize simulation statistics Z_o and Z_c by zeros for purpose of calculation (3.8) and (3.12) or (3.13).
         - Set initial values of signals to the mean values of identification data.
         - Set time counter t to one.

- Initialize the adaptive controller estimate of the controlled model parameters $f(\Theta_A)$, see Section 3.3.2, of the adaptive controller to be equal to the off-line estimated pdf $f(\Theta)$ from identification.

(d) Simulation loop.

i. Perform one output data y_t sampling y_t from the distribution described by the pdf $f(y_t|u_t, \varphi_t)$ (2.23).

ii. Update the statistics of Z_c and Z_o according to (3.8) and (3.12) or (3.13) using newly available data.

iii. Exit the simulation loop and continue from 3e, if the estimate of the functions Z_c and Z_o is representative. This is decided by a stopping rule, see Section 3.5.4, whenever:

- Enough information is collected – for noise compensation
- All important changes in reference trajectory passed – for servo control task

iv. Update the controller estimate of the controlled model $f(\Theta_A)$ using the last sampled data, see (2.19).

v. Design a controller K_t based on the current tuning parameters q and the point estimate $\hat{\Theta}_A$ of the adaptive model parameters pdf $f(\Theta_A)$.

vi. Set $t = t + 1$, generate a new system input u_t using the controller K_t.

vii. Continue with the next simulation step 3d.

(e) Update the multiple simulation statistics Z_c^N and Z_o^N using the simulation results in form of samples of Z_c and Z_o.

(f) Exit the multiple simulation loop, if there is enough information collected according to the stopping rule described in Section 3.5.3.

(g) Otherwise continue with next multiple simulation loop 3c.

4. Optimization loop branching

- If the selected deterministic optimization method finds the optimum, return the optimal tuning parameter value q as the result.
- If no feasible solution can be found report an error.
- Otherwise the optimization algorithm chooses a new optimum approximation q according to newly obtained values Z_c^N and Z_o^N and continues with a new optimization loop 3.

For the case of LQG controller, the initial estimate of tuning parameters, the point 2, and controller design and action, points 3(d)v and 3(d)vi, are described in Chapter 4.

3.7.2 Designer Toolbox

The presented controller tuning algorithm was implemented as the core part of the Designer toolbox. However, the real implementation is of course much more complex than several nested loops.

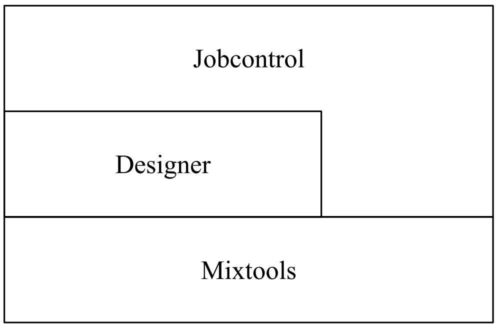

Figure 3.8: Structure of the Jobcontrol toolbox

The Designer—a toolbox responsible for the controller tuning—forms a part of the toolbox Jobcontrol [?] that handles the whole process of the controller design with utilizing the basic algorithm of the toolbox Mixtools [?]. The composition of these toolboxes is shown in Figure 3.8.

The Jobcontrol processing starts with the identification data measured on the real system and finishes with the designed controller and its verification. The complete processing is divided into several steps, see Figure 3.9. Let us present all the steps of the controller design briefly.

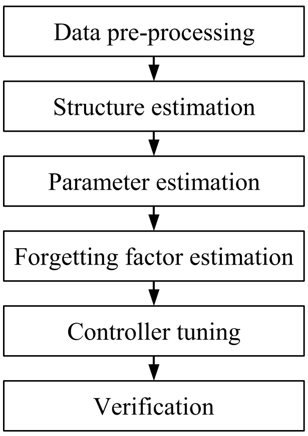

Figure 3.9: Steps of controller design

Data Pre-Processing

Data forms the basic source of information about the process. Data pre-processing is the starting step of the design. The raw data are sampled and/or grouped according to the expected range of the control period, scaled to similar numerical level, outlying data are removed, and high frequency noise is suppressed. These simple standard signal-processing actions are vital for numerical treatment and potential validity of the model. Data pre-processed in this way are used in steps that follow. The same scaling has to be used whenever appropriate, for instance for scaling of the set point. The scaling is stored for purposes of reporting results of the whole design in the original scale.

Structure Estimation

Model structure characterizes significant data input-output relationships. The structure is connected with the used parametric system model, such as ARX one, where some parameters are selected as important characteristics of the system. The rest of parameters marked as unimportant is considered to have zero value.

Essentially, the most probable structure conditioned by the measured data and prior information is searched for within a rich space of model structures. The search is locally guided in a direction of the best structure probability. This is based on the general theory of Bayesian structure estimation [**?**]. Other details can be found in [7].

Parameter Estimation

This step estimates unknown model parameters selected as possibly nonzero by the structure estimation. The Bayesian set up is used for this purpose. The real data, fictitious data reflecting prior knowledge, control period, and model structure serve as input to this step. The adopted estimation qualifies the range of possible models by probabilities reflecting processed knowledge. Consequently, it provides uncertainties of the resulting estimates even after processing of a limited amount of the data. The uncertainties are considered in the closed loop behavior evaluation. For the ARX model, the Bayesian estimation leads to an algorithm formally equivalent to recursive least squares. It is known to be numerically sensitive so its numerical robust factorized version [**?**] is implemented. The estimation results, used in the rest of the design, serve for initialization of the on-line estimation of the controller adaptation process and provide the alternative model needed in stabilized forgetting.

Forgetting Factor Estimation

Forgetting is a process applied in the on-line use of the considered adaptive controller. It suppresses the obsolete information on model parameter estimates. Its use converts self-tuning controllers into truly adaptive ones. Forgetting is known to increase the sensitivity of the parameter estimator to informative content of the processed data. This sensitivity has inhibited applications of adaptive control for a long time. The advanced stabilized forgetting [**?**] is used within the controller design task. This forgetting is controlled by the model obtained in the off-line estimation and by an optional forgetting factor that is estimated in this step. Essentially, the most probable forgetting factor is searched among finite set of alternative values. Details about the estimation of forgetting factor can be found in [**?**].

Controller Tuning

This step forms the central topic of this work. The algorithm described in Section 3.7.1 is placed here.

Verification

The controller verification step evaluates performance of the closed loop formed by the controller and controlled system or realistic model. Use of the controller in closed loop is the final test of design success. Discrepancies between expected and real design results have to be carefully considered and if the results are not acceptable, the whole design should be repeated with a changed specification. The possible discrepancies are caused either by unrealistic wishes and constraints or by a bad model quality. The latter one is caused by the lack of informative data that was not substituted by a sufficient prior knowledge.

Chapter 4

Design of the Adaptive LQG Controller using ARX Model

The adaptive LQG controller is suitable for control of stochastic systems. Although the presented approach of the controller tuning is not derived for any particular controller, the LQG one was selected because it is derived directly for the systems with Gaussian noise. In this chapter, the description of the LQG controller is given, the tuning parameters determining the control criterion are explained, and finally two techniques to obtain an approximation of the proper tuning parameter values are described.

4.1 LQG Controller Principle

The classical principle of the LQG controller is shown in this section for the state space model. Let us consider the state space model

$$\begin{aligned}\varphi_{t+1} &= A\varphi_t + Bu_t + e_t^{\varphi} \\ y_t &= C\varphi_t + Du_t + e_t^{y},\end{aligned}$$

where e_t^{φ} and e_t^{y} are white Gaussian noise signals.

The aim of the control is to minimize the expected value of the following quadratic criterion

$$J_t = \varphi'_{t+h} Q_{\varphi} \varphi_{t+h} + \sum_{\tau=t}^{t+h-1} (\varphi'_{\tau} Q_{\varphi} \varphi_{\tau} + u'_{\tau} Q_u u_{\tau})$$

determined by the positive semi-definite matrices Q_{φ} and Q_u.

The solution is found in the form of the input signal generated by the linear control law

$$u_t = -K'(\hat{\theta}_{t-1}, Q_u, Q_{\varphi})\varphi_{t-1}. \tag{4.1}$$

Matrix gain $K(\cdot) \in \mathbf{R}^{\mathring{u},\mathring{\varphi}}$ results from the dynamic programming, Section 2.3.3, that reduces to the solution of the Riccati equation [?]

$$K = (Q_u + B'S_0B)^{-1}B'S_0A. \tag{4.2}$$

The positive semi-definite Riccati matrix S_0 computed from the Riccati equation

$$S_i = A'S_{i+1}A - A'S_{i+1}B(Q_u + B'S_{i+1}B)^{-1}B'S_{i+1}A + Q_\varphi, \tag{4.3}$$

starting from known $S_h = Q_\varphi$.

4.1.1 Adaptive LQG Controller for ARX Model

The LQG controller, used in this thesis, is based on the ARX system model (2.21). This model is in the adaptive case recursively updated, see Section 3.3.2, and its parameters are denoted by subscript A

$$f(y_t|\psi_t, \Theta_A) = f(y_t|\psi_t, \theta_A, R_A) \sim \mathcal{N}(\theta_A\psi_t, R_A), \quad \Theta_A = (\theta_A, R_A). \tag{4.4}$$

Adaptive LQG controller uses the current point estimates $\hat{\theta}_{A;t-1}$ of the estimated parameter pdf $f(\theta_A|d(t-1))$ and minimizes the quadratic loss J_t over the h-th receding-horizon

$$J_t = \sum_{\tau=t}^{t+h} \begin{bmatrix} y_\tau \\ \psi_\tau \end{bmatrix}' Q \begin{bmatrix} y_\tau \\ \psi_\tau \end{bmatrix}, \tag{4.5}$$

determined by positive semi-definite matrix Q.

The LQG controller for the ARX model is formally equivalent to the state space model shown in Section 4.1, and also the linear control law is obtained. The exact solution of the Riccati equation (4.3) for ARX model is of course different from the state space model. For its safe numerical evaluation the factorized equivalent is used, see [?].

4.1.2 Tuning Parameters of the LQG Controller

All parameters of a controller that are constant through the simulation process can be tuned by the adopted methodology. In the case of the LQG controller, the tuning parameters are represented by the kernel Q of its quadratic criterion (4.5).

The number of independent elements of matrix Q is often high and its full optimization is computationally intensive. Moreover, the necessary positive semi-definiteness of the matrix Q represents a significant constraint imposed on its entries.

It is also worthwhile knowing the physical meaning of particular tuning parameters, which is rather difficult in the situation of the whole matrix Q.

Thus, the quadratic criterion constructed from smaller and simpler pieces is suitable to decrease the problem of dimensionality. It is wise to parameterize Q by tuning parameters with a good intuitive physical meaning. It allows us to get control over particular properties of the controller. The following quadratic criterion is considered

$$J_t = \sum_{\tau=t}^{t+h} (q_1 l_{1;\tau}^2 + q_{2;\tau} l_{2;\tau}^2 + \ldots + q_{\mathring{q}} l_{\mathring{q};\tau}^2), \tag{4.6}$$

where the scalar weights $q_\bullet \geq 0$, called penalization coefficients, are taken as the tuning parameters. The linear vector function l_t depends on quantities y_t and ψ_t and measures the signal deviations from the desired values. The particular elements of the function l_t are weighted by the tuning parameters q.

Generally, the overall criterion (4.6) depends on quantities y_t and ψ_t in the same way as the full quadratic criterion (4.5) does. It makes sense, however, to fix the linear functions and let the designer of the LQG controller find the weights $q_\bullet$ only.

Typical forms of the quadratic criterion, given by specific function l_t, and their correspondence with the constraints imposed on particular signals follow.

The regulation problem is the simplest variant. The controller drives the system in order to keep the output close to the desired set point y^{ref} and the input close to its reference value u^{ref}. Penalizations are then chosen to penalize the output error and to penalize difference of the input from its reference value

$$l_t = \left[y_{1;t} - y_1^{\text{ref}}, y_{2;t} - y_2^{\text{ref}}, \ldots, u_{1;t} - u_1^{\text{ref}}, u_{2;t} - u_2^{\text{ref}}, \ldots \right]. \tag{4.7}$$

Penalization weights belonging to the model output $q_1, \ldots, q_{\mathring{y}}$ and input $q_{\mathring{y}+1}, \ldots, q_{\mathring{y}+\mathring{u}}$ have to be set to represent optimal trade-off between regulation error and actuator effort. The discussed penalizations suit well for tuning of the controller that respects constraints on the range of the system input while minimizing regulation error.

The joint servo-regulation problem is obtained by allowing time dependent y^{ref}. In this case, limits on input increments are put more often. Then, the appropriate penalization of input increments is

$$l_t = \left[y_{1;t} - y_{1;t}^{\text{ref}}, y_{2;t} - y_{2;t}^{\text{ref}}, \ldots, u_{1;t} - u_{1;t-1}, u_{2;t} - u_{2;t-1}, \ldots \right]. \tag{4.8}$$

This discrete-time analogy of the first derivative can be extended to the penalization of the discrete-time analogy of higher order derivatives.

To cope effectively with the constraints of mutually dependent signals in case of MIMO system, the corresponding l_t vector function entry has to be included. For instance, closeness of two input signals is controlled by the following entry

$$u_{i;t} - u_{j;t}, \quad i \neq j. \tag{4.9}$$

This kind of penalization is called non-diagonal because of its matrix representation as a quadratic form.

Note Though the function l_t might resemble the constraint vector c_t, see Section 3.4.2, forming the function Z_c (3.11) or the argument of the quadratic criterion (3.8) forming the loss function Z_o, there is a basic difference. The function l_t is parameterized by the tuning parameters q and it determines the properties of the LQG controller through its quadratic criterion J_t (4.6). Thus the criterion J_t is not any global measure of the controller quality. It is merely a tool changed by the controller tuning process in order to satisfy user's requirements represented by absolute—not parameterized—loss function Z_o and constraint function Z_c.

4.2 Initial Approximate of Tuning Parameters

In the remaining part of this chapter, two methods for estimation of reasonable initial tuning parameter values are introduced. The first method yields an approximation of the input penalization by using the estimated system model and the second method approximates both the input and output penalizations with respect to the desired closed loop variances and also gives an upper bound of the output penalization.

4.2.1 Approximate Lower Bound of Tuning Parameters for LQG Controller

First estimation uses a pilot solution of the Riccati equation (4.3) with all matrices marked by zero in superscript. Riccati matrix S_t^0 computed recursively by the Riccati equation with Q_u^0 set to zero can be used as an approximate lower bound of matrix Q_u.

Considering factor

$$Q_u + B'S_{i+1}B$$

from (4.3), the control law solution does not significantly change if Q_u is of smaller order than the term $B'S_{i+1}B$. This term is, however, dependent on Q_u and therefore it cannot be used to determine the bound on Q_u directly. Using matrix S_i^0, evaluated from the Riccati equation with matrix $Q_u^0 = 0$ on the same horizon and the same matrix Q_y, gives us a useful estimate $S_t^0 \leq S_t$ in the sense of positive semidefinitness.

This holds because criterion J_t in (4.5) is smaller if the term in its defining summation is smaller. Thus, for J_t^0 calculated for $Q_u^0 = 0$ holds $J_t^0 \leq J_t$ for any initial state φ_t. The criterion J_t can be also written as

$$J_t = \varphi_t' S_0 \varphi_t \text{ and } J_t^0 = \varphi_t' S_0^0 \varphi_t,$$

therefore $S_0^0 \leq S_0$ in the sense of positive semidefinitness.

4.2.2 Tuning Parameters Approximation Using Fully Probabilistic Design

In this section, a method called the Fully Probabilistic Design is used to obtain initial approximations of the optimal tuning parameter values from the properties of this design method.

Fully probabilistic design [?] solves the dynamic decision task from Section 2.3.3 that searches for a stochastic controller given by pdf $f(u_t|\varphi_{t-1})$ for a given system model $f(y_t|\psi_t)$ such that the Kullback-Leibler divergence of the density of closed loop behavior of proposed controller to some ideal density is minimized. The ideal closed loop behavior is given by pdfs of ideal controller ${}^If(u_t|\varphi_{t-1})$ and ideal system ${}^If(y_t|\psi_t)$.

$$\begin{aligned}
\text{minimize} \quad & \mathcal{D}_{\mathrm{KL}}(f(d(T)|d(0)) \| {}^If(d(T)|d(0))) \\
\text{over} \quad & f(u_t|\varphi_{t-1}) \\
\text{where} \quad & f(d(T)|d(0)) = \prod_{\tau=1}^{T} f(y_\tau|\psi_\tau) f(u_\tau|\varphi_{\tau-1}) \\
\text{and} \quad & {}^If(d(T)|d(0)) = \prod_{\tau=1}^{T} {}^If(y_\tau|\psi_\tau)\, {}^If(u_\tau|\varphi_{\tau-1}),
\end{aligned}$$

where $\mathcal{D}_{\mathrm{KL}}(\cdot\|\cdot)$ is the Kullback-Leibler divergence (3.20). This rather general formulation has a solution for the Gaussian ARX model and static ideal distributions.

$$\begin{aligned}
f(y_t|\psi_t) &\sim \mathcal{N}(\theta_A \psi_t | R_A) \\
{}^If(y_t|\psi_t) &\sim \mathcal{N}(0, Q_y^{-1}) \\
{}^If(u_t|\varphi_{t-1}) &\sim \mathcal{N}(0, Q_u^{-1})
\end{aligned} \tag{4.10}$$

The resulting controller pdf has the form

$$f(u_t|\varphi_{t-1}) \sim \mathcal{N}(K\varphi_{t-1}, R_u)$$

where matrix K is the same as the linear control law (4.2) produced by LQG controller with quadratic criterion

$$J_t = \sum_{\tau=1}^{T} u_\tau' Q_u u_\tau + y_\tau' Q_y y_\tau$$

and added white noise of certain variance R_u. For more detail see [**?**].

From this result, it is possible to approximate the matrices Q_u and Q_y from desired respective variances Q_u^{-1} and Q_y^{-1}, which are for one step horizon closest to the ideal.

The second advice obtained from the fully probabilistic design is that it makes only sense choosing the desired output signal variance Q_y^{-1} not less than the process noise covariance R of the model $f(y_t|\psi_t)$. Thus $Q_y^{-1} \geq R$ in positive semi-definiteness. If no special requirement for the output variance are given, the choice $Q_y^{-1} = R$ is the reasonable one.

Chapter 5

Experiments

This chapter presents several simulated experiments where all the steps of the controller design performed by the Jobcontrol toolbox are demonstrated emphasizing the controller tuning task – the main topic of this work.

The first part, Section 5.1, shows the properties of designed controller for a simple system with various design settings. Then Section 5.2 presents a controller design for a complex model of binary distillation column.

5.1 Simple MIMO System

The first set of experiments is demonstrated on a simple MIMO system of two inputs and two outputs. Purpose of these experiments is to show basic properties of the proposed algorithm and to verify its reliability for more complex tasks such as the distillation column control presented in the next section. The discrete MIMO system is constructed from a set of SISO subsystems interconnected as shown if Figure 5.1.

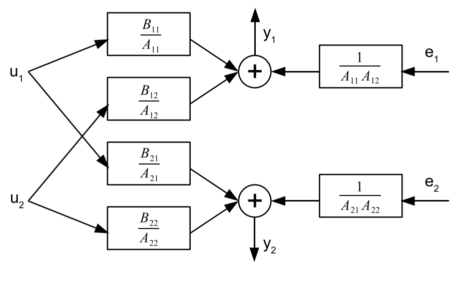

Figure 5.1: MIMO ARX system constructed from four SISO subsystems.

Reasonable discrete-time SISO system blocks are obtained by discretization of continuous-time SISO blocks. The SISO blocks

$$S(s) = \begin{bmatrix} \frac{1}{s+0.8} & \frac{0.5}{s+1} \\ \frac{0.5}{s+1.2} & \frac{1}{s+1.4} \end{bmatrix}$$

are discretized with sampling period 0.1 resulting to a discrete-time blocks $S(z)$ where $\frac{B_{ij}}{A_{ij}} = S_{ij}(z)$

$$S(s) = \begin{bmatrix} \frac{0.0961}{z-0.9231} & \frac{0.04758}{z-0.9048} \\ \frac{0.04712}{z-0.8869} & \frac{0.09332}{z-0.8694} \end{bmatrix}.$$

Noise signal is filtered in order to create an ARX model. Thus the filter has unit numerator and its denominator is the product of the denominators of respective blocks for given output. The noise power is $1 \cdot 10^{-5}$.

The Jobcontrol toolbox has not access to the exact definition of this system. It uses data generated from it and identifies its own model of the controlled system. Only in the verification of the designed controller the original system is used to prove the capabilities of the controller.

There were 10000 data samples generated for the Jobcontrol identification. The driving input signal was white noise distributed by $\mathcal{N}(0, 1)$.

5.1.1 Identifiacation

The system is identified in form of ARX model

$$y_t = \theta\psi_t + Re_t, \quad e_t \sim \mathcal{N}(0, 1)$$

where the model parameters Θ are divided into the coefficient parameters and covariance $\Theta = (\theta, R)$ as described in Section 2.6. For the identification an ARX model of maximal order of two was used. Model structure estimation selected only some model parameters to be estimated, while the rest is considered unimportant and set equal to zero. The estimated model parameters are obtained as a posterior Gauss-inverse-Wishart distribution (2.22). For illustration the maximum likelihood estimate parameters form the following model

$$\begin{aligned} y_{1,t} &= 1.82y_{1,t-1} - 0.832y_{1,t-2} + 0.0961u_{1,t-1} + 0.0475u_{2,t-1} - \\ &- 0.0867u_{1,t-2} - 0.0437u_{2,t-2} + 1.07 \cdot 10^{-5}e_{1,t} \\ y_{1,t} &= 1.75y_{1,t-1} - 0.768y_{1,t-2} + 0.0471u_{1,t-1} + 0.0933u_{2,t-1} - \\ &- 0.0407u_{1,t-2} - 0.082u_{2,t-2} + 1.05 \cdot 10^{-5}e_{1,t}. \end{aligned}$$

To show the parameter uncertainty the diagonals of parameter covariance follows

$$\begin{aligned} \mathrm{diag}(\mathrm{cov}\Theta_1) &= [1.0 \cdot 10^{-3},\ 8.40 \cdot 10^{-5},\ 8.62 \cdot 10^{-5}, \\ &\quad 7.2 \cdot 10^{-3},\ 4.2 \cdot 10^{-4},\ 2.9 \cdot 10^{-3}] \\ \mathrm{diag}(\mathrm{cov}\Theta_2) &= [1.8 \cdot 10^{-3},\ 8.40 \cdot 10^{-5},\ 8.62 \cdot 10^{-5}, \\ &\quad 7.7 \cdot 10^{-3},\ 1.0 \cdot 10^{-4},\ 1.3 \cdot 10^{-2}]. \end{aligned}$$

The order of parameter elements in Θ corresponds to the order of parameters in the point estimate model above.

5.1.2 Servo-Control Task

The designed controller has to follow reference trajectory with steps in both output channels in different times

$$y^{\text{ref}}_{1,t} = 0 \qquad \text{for } t < 100 \tag{5.1}$$
$$y^{\text{ref}}_{1,t} = 1 \qquad \text{for } t \geq 100 \tag{5.2}$$
$$y^{\text{ref}}_{2,t} = 0 \qquad \text{for } t < 200 \tag{5.3}$$
$$y^{\text{ref}}_{2,t} = 1 \qquad \text{for } t \geq 200 \tag{5.4}$$

while keeping the input signals in specified range

$$u_{1,t} \in \langle -5, 5 \rangle \tag{5.5}$$
$$u_{2,t} \in \langle -5, 5 \rangle . \tag{5.6}$$

As this task falls into servo control, Section 3.4.3, the constraint function is chosen to measure maximal constraint violation according to (3.12).

The tuning parameters were selected to penalize the input signals magnitude. The penalization of outputs was fixed one.

$$J_t = \sum_{\tau=t}^{t+h} y^2_{1;\tau} + y^2_{2;\tau} + q_1 u^2_{1;\tau} + q_2 u^2_{2;\tau}.$$

The LQG horizon length h was 100 steps.

The initial estimate of the tuning parameters was obtained using the lower bound approximation described in Section 4.2.1. This initial estimate at $[0.0541, 0.0541]$ was refined to value $[0.0229, 0.0218]$ by single dimensional search of zero crossing of the constraint function as described in Section 3.6.3. Finally the multidimensional constrained optimization found the optimum at $[0.0218, 0.0171]$.

The simulated run of the tuned controller with the estimated model of the controlled system is shown in Figure 5.2. To evaluate the obtained controller the verification with the original system is shown in Figure 5.3. The predicted and verified behavior are very similar, thus the controller design was successful.

Controller for Different Input Signal Ranges

The values of tuning parameters for controller design under the same conditions as in the previous section but for different input signals ranges are shown in the following table.

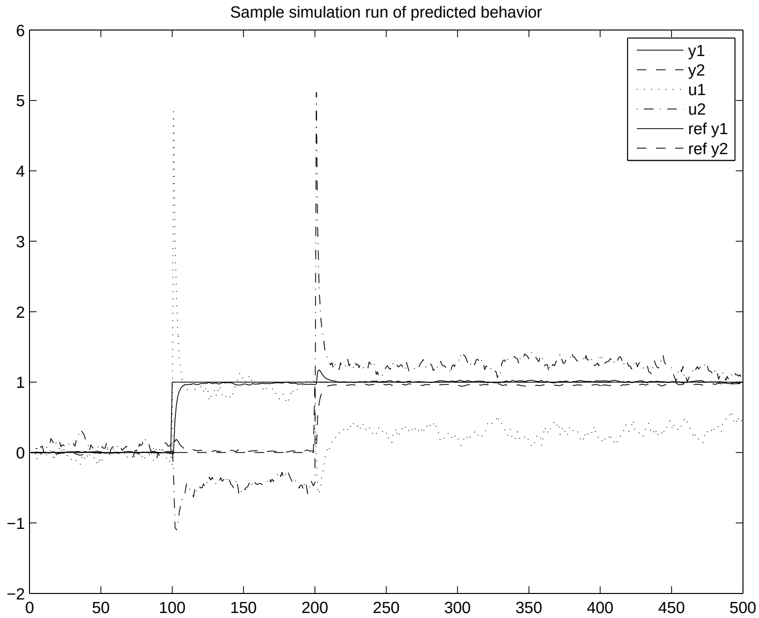

Figure 5.2: Prediction of controller designed for constraints of $\langle -5, 5\rangle$ for both inputs

Allowed range for inputs	Resulting tuning parameter	Z_o
$u_1 \in \langle -5, 5\rangle, \quad u_2 \in \langle -5, 5\rangle$	$[0.0218, 0.0171]$	0.0057
$u_1 \in \langle -3, 3\rangle, \quad u_2 \in \langle -3, 3\rangle$	$[0.0735, 0.0524]$	0.0107
$u_1 \in \langle -1, 1\rangle, \quad u_2 \in \langle -1, 1\rangle$	$[0.565, 0.4006]$	0.0641
$u_1 \in \langle -5, 5\rangle, \quad u_2 \in \langle -3, 3\rangle$	$[0.0209, 0.0513]$	0.0096

The constraints were satisfied for all ranges. As the elements of tuning parameter vector penalize the respective elements of the input signal, their reciprocal dependency on the input intervals size can be seen.

Controller with Constrained Input Increments

This section modifies the controller designed in the beginning of Section 5.1.2 so that the constraints on the input magnitudes are replaced by constraints on input increments. The tuning parameter vector was extended by elements penalizing the increments. The quadratic criterion of LQG controller is now in the form

$$J_t = \sum_{\tau=t}^{t+h} y_{1;\tau}^2 + y_{2;\tau}^2 + q_1 u_{1;\tau}^2 + q_2 u_{2;\tau}^2 + q_3 \Delta u_{1;\tau}^2 + q_4 \Delta u_{2;\tau}^2$$

where $\Delta u_t = u_t - u_{t-1}$.

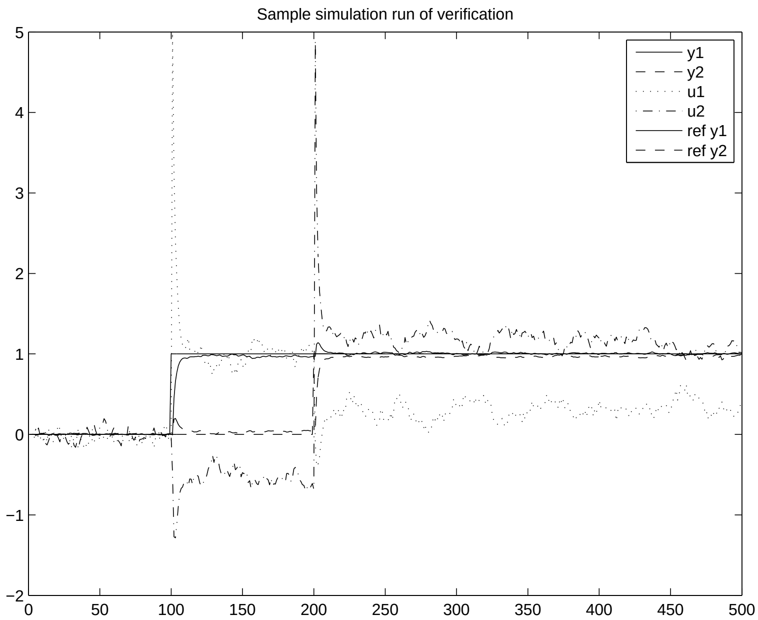

Figure 5.3: Verification of controller designed for constraints of $\langle -5, 5\rangle$ for both inputs

The tuning procedure used the increment penalizations q_3 and q_4 to achieve the desired behavior, while the magnitudes penalizations q_1 and q_2 are very small. The verification simulation run of controller designed for interval $\langle -1, 1\rangle$ for both inputs is shown in Figure 5.4. This experiment was repeated for different input increments constraints with results shown in the following table.

Allowed range for increments	Resulting tuning parameter	Z_o
$\Delta u_1 \in \langle -5, 5\rangle, \quad \Delta u_2 \in \langle -5, 5\rangle$	$[0.0001, 0.0003, 0.0110, 0.0128]$	0.0049
$\Delta u_1 \in \langle -3, 3\rangle, \quad \Delta u_2 \in \langle -3, 3\rangle$	$[0.0003, 0.0004, 0.0462, 0.0519]$	0.0059
$\Delta u_1 \in \langle -1, 1\rangle, \quad \Delta u_2 \in \langle -1, 1\rangle$	$[0.0007, 0.0009, 0.6983, 0.6985]$	0.0094
$\Delta u_1 \in \langle -5, 5\rangle, \quad \Delta u_2 \in \langle -1, 1\rangle$	$[0.0013, 0.0018, 0.0151, 0.6555]$	0.0073

The use of increment penalization removes the problem of constant regulation error even for rather restrictive constraints in comparison to the magnitude penalization in the previous experiment.

5.1.3 Regulation Task

Regulation task uses the same settings as the servo-control task in Section 5.1.2, up to zero setpoint. The controller now compensates only the system disturbance.

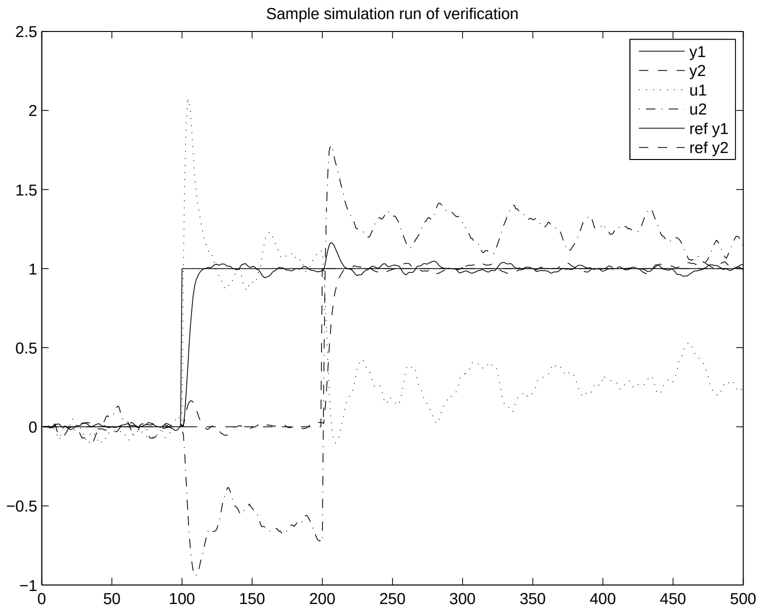

Figure 5.4: Verification of controller designed for increments bounded to $\langle -1, 1\rangle$ for both inputs

Controller for Different System Disturbance

Settings for this experiments include fixed constraints on inputs $\Delta u_1 \in \langle -5, 5\rangle, \quad \Delta u_2 \in \langle -5, 5\rangle$. The tuning result for various covariance of disturbance is show in the following table.

Noise covariance of both e_1 and e_2	Resulting tuning parameter	Z_o
$\text{cov}e_1 = \text{cov}e_2 = 1$	$[3.0977, 8.2269]$	110
$\text{cov}e_1 = \text{cov}e_2 = 1 \cdot 10^{-1}$	$[0.3093, 0.2272]$	15.6
$\text{cov}e_1 = \text{cov}e_2 = 1 \cdot 10^{-2}$	$[0.0722, 0.0272]$	0.413
$\text{cov}e_1 = \text{cov}e_2 = 1 \cdot 10^{-3}$	$[1 \cdot 10^{-9}, 1 \cdot 10^{-9}]$	0.0043
$\text{cov}e_1 = \text{cov}e_2 = 1 \cdot 10^{-4}$	$[1 \cdot 10^{-10}, 1 \cdot 10^{-10}]$	0.00043
$\text{cov}e_1 = 1 \cdot 10^{-1}, \quad \text{cov}e_2 = 1 \cdot 10^{-4}$	$[0.3477, 0.1236]$	5.77

As expected, with decreasing disturbance covariance it is sufficient to use decreasing the penalization to keep constraints satisfied.

Sample verification run for $\text{cov}e_1 = \text{cov}e_2 = 1 \cdot 10^{-3}$ is show in Figure 5.5.

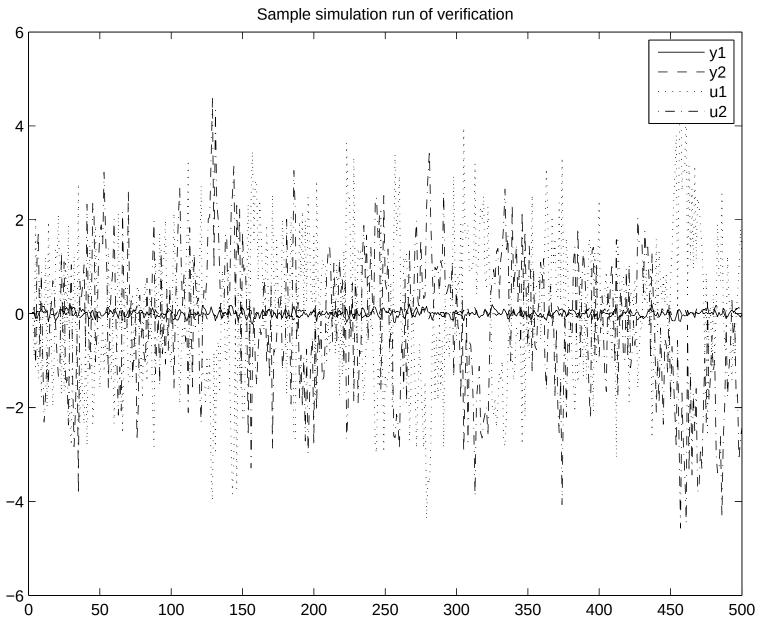

Figure 5.5: Verification of controller designed for noise compensation

5.1.4 Controller for Different Uncertainty of System Parameters

This experiment shows the effect of parameter estimate uncertainty on the tuned closed loop. It uses the noise compensation for the same settings as in the previous example with fixed disturbance covariance at $\text{cov}e_1 = \text{cov}e_2 = 1 \cdot 10^{-1}$. The parameter covariance $\text{cov}\Theta_1$ and $\text{cov}\Theta_2$, described in Section 5.1.1, are multiplied by various factors to simulate less accurate identification.

Parameter covariance multiplier	Resulting tuning parameter	Z_o	$\max(Z_c)$
1	$[0.3093, 0.2272]$	15.6	$-1.7 \cdot 10^{-7}$
5	$[0.592, 0.284]$	18.8	$-3.6 \cdot 10^{-7}$
10	$[0.961, 4.37]$	22.4	-0.00451
15	$[2.756, 9.27]$	30.9	-0.095
20	$[4.71, 59.4]$	48.4	-0.099

The last column in the table shows the maximal constraint violation of Z_c. It can be seen that at smaller parameter covariances, the optimal controller reaches the constraints, which is indicated by $\max(Z_c)$ be almost zero. For bigger parameter uncertainties, the value of $\max(Z_c)$ is negative. This shows the input constraint interval is not used entirely. This is caused by the fact that controller for highly uncertain system has to be conservative, otherwise the controller-model mismatch causes worse control behavior for controllers using bigger input range, see Section 3.6.3 and Figure 3.6.

5.1.5 Experiment Summary

The experiments above presented properties of controller design consisting of the identification, controller tuning for both servo-control and noise compensation tasks, and finally the verification of the resulting controller. All controllers designed in the performed experiments satisfied the given constraints. The dependency of the tuning parameters on conditions was shown, namely several cases of constraint intervals for magnitudes and increments, covariances of system disturbance and covariances of the uncertain model parameters were shown in tables.

The satisfactory results of these simple experiments encouraged us to do more complex tasks such as the experiment described in the following text.

5.2 Binary Distillation Column

This section presents an experiment with a model of binary distillation column taken from the IFAC benchmark problems [3] example No. 1. This model has been selected to represent a complex close to real system.

5.2.1 Problem Description

The task is to control a linear continuous system of 11-th order with three inputs and three outputs with an external disturbance. The description of the system includes constraints imposed on the input variables and defines the disturbance to be compensated. As only the allowed ranges for input signal magnitudes are bounded, the constraint vector c_t, Section 3.4.2, is identical to the input $c_t = u_t$. The particular ranges are:

$$\begin{aligned} u_1 &\in C_1 = \langle -2.5, 2.5 \rangle \\ u_2 &\in C_2 = \langle -2.5, 2.5 \rangle \\ u_3 &\in C_3 = \langle -0.3, 0.3 \rangle \quad . \end{aligned}$$

Whereas the control task is to compensate the disturbance, the constraint function Z_{cP} (3.13) was selected to represent this type of constraints with minimal probability of constraint satisfaction being set to $\alpha_{\min} = 0.9$. The matrix W of the loss function Z_o (3.8) was chosen as recommended in Section 3.4.1 as the reciprocal values to the conditional variances of respective output signals in d_t.

A Simulink model was used to model the reality. The scheme in Figure 5.6 contains blocks representing the plant and the disturbance. Matrices A, B, C of the block "System model" and matrices A, B of the block "Noise

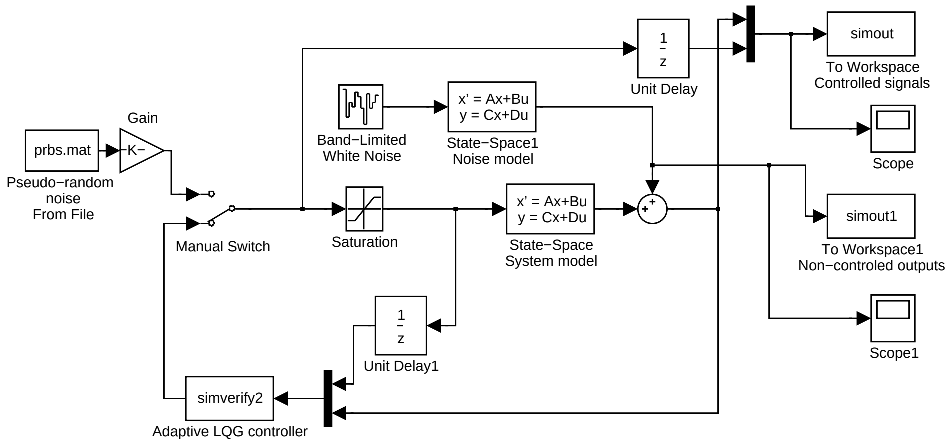

Figure 5.6: Simulink model of closed loop of binary distillation column.

model" are described in [3]. Matrices D for both blocks are zero with appropriate dimensions.

The data used further in the design were obtained as a response to the pseudo-random binary signal (PRBS) system input and a disturbance in the output.

5.2.2 Processing and Results

The control design process passes through the following procedures as described in Section 3.7.2:

1. Data Preprocessing. In this case, the data are only scaled and reasonable sampling is chosen. The scaling is done automatically, the sampling period was chosen $T_s = 10s$ on basis of the step response test. The time span for measured data was 100000 seconds. The preprocessing took about a half of second on the AMD Athlon 1800+ processor.

2. Structure Estimation. Data are processed by function determining the most probable structure of the linear regression model. Prior information about the gain was used (see [7]) for improved structure determination. The maximum structure was allowed up to the delay 15 of any variable. The best structure estimated using prior information consisting of the static gains of the system is shown in Table 5.1. The calculation took about 4 minutes.

3. Parameter Estimation. Data are used by the parameter estimation function. Parameters of the linear regression model, having the structure determined in the previous step, are identified. The resulting MIMO model

MISO model for channel y_1

delays	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
y_1		*	*	*	*	*	*	*	*	*	*					
y_2	*	*	*	*	*	*	*	*	*	*	*					
y_3	*	*	*	*	*	*	*	*	*		*					
u_1	*		*		*		*	*		*	*	*				
u_2	*		*			*			*							
u_3	*	*	*	*	*			*	*							

MISO model for channel y_2

delays	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
y_1		*	*	*	*	*	*									
y_2		*	*	*	*	*	*									
y_3	*	*	*		*		*									
u_1	*	*	*	*												
u_2	*	*	*	*	*	*										
u_3	*	*	*	*												

MISO model for channel y_3

delays	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
y_1		*	*	*	*											
y_2		*														
y_3		*	*	*												
u_1	*	*	*	*												
u_2	*	*	*													
u_3	*	*	*	*												

Table 5.1: Estimated structure of three MISO factors of the binary distillation column. Nonzero entries are marked by $*$.

is identified in the form of a set of MISO ARX models (2.22). The particular parameters are not shown here. The calculation took about 3 seconds.

4. Forgetting Factor Estimation. After parameter identification, the data are processed again with the aim to determine appropriate forgetting factor. Estimated value is 0.3842. As an alternative model is used the one obtained in the previous step. The calculation took about 7 minutes.

5. Optimization Procedure. The optimization procedure, Section 3.6, performs simulations using the identified model as a plant, evaluates the results represented by fulfillment of the requirements and designs a new controller tuning parameter settings represented by the penalizations.

The tuning parameters of tuned adaptive LQG controller are defined by the following quadratic criterion (4.6)

$$J_t = \sum_{\tau=t}^{t+h} y_{1;\tau}^2 + y_{2;\tau}^2 + y_{3;\tau}^2 + q_1 u_{1;\tau}^2 + q_2 u_{2;\tau}^2 + q_3 u_{3;\tau}^2$$

with horizon $h = 200$ for generating the first input sample. The penalizations for outputs was set to one, because of the signal scaling. The iterations spread in time (IST) method [**?**] is used in the rest of simulation.

The results of the optimization procedure are the recommended penalization and prediction of the closed loop behavior. As the starting point of the optimization algorithm was chosen the approximate lower bound of penalization according to Section 4.2.1. The sample path method was used to find the optimal tuning parameter with starting approximation q_0. The optimization took 64 simulation runs. The optimum found was

$$q = [0.0093,\ 0.0232,\ 0.0284].$$

The constraints of the first and third input range were active. A sample of simulation run using the optimal controller is shown in Figure 5.7 and the histogram of the signal values are in Figure 5.9. Variances and constraint satisfaction probabilities, estimated as relative frequencies, of signals of the optimized control loop are shown in the Tables 5.1(a) and 5.1(b) in columns called "predicted." The calculation took about 8 minutes.

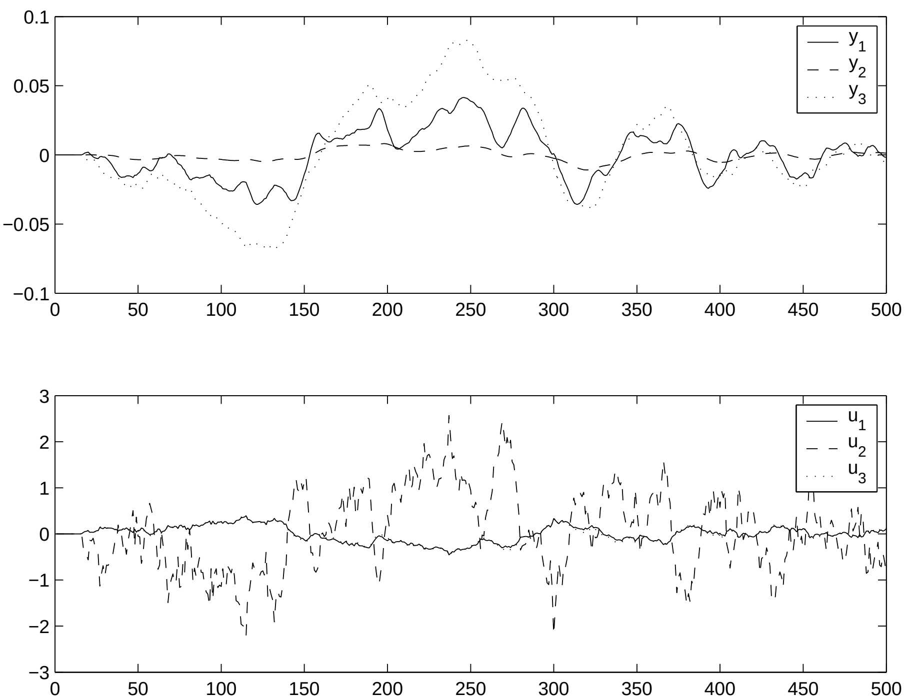

Figure 5.7: Sample simulation run in tuning phase. The upper plot shows the output signals and the lower one shows the input signals.

6. Verification. The verification step confronts the derived setting with the true plant—in this example with the Simulink model. Sample of simulation run using the optimal controller is shown in Figure 5.8 and the

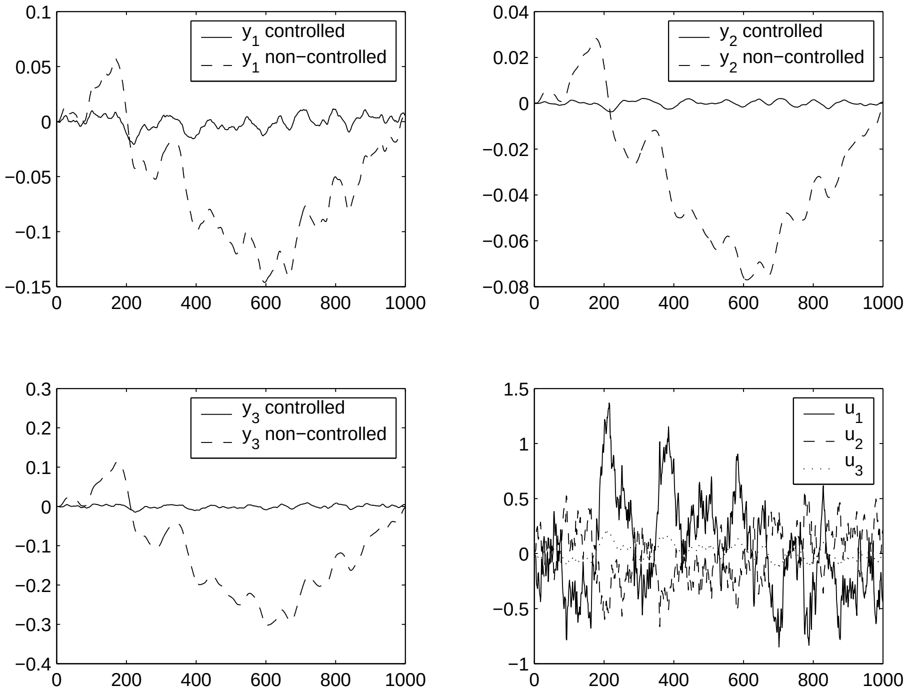

Figure 5.8: Sample Simulink verification run. First three plots shows the particular controlled and non-controlled output signals and the fourth one shows the controlling input signal.

histograms of the signal values are in Figure 5.10. Now the differences between the behavior predicted by the optimized model and the verification from the original Simulink model can be compared. The variances of signals are shown in Table 5.1(a) and the relative frequencies of input signal constraint satisfaction are shown in Table 5.1(b).

5.2.3 Distillation Column Summary

The LQG controller design by the Jobcontrol toolbox performs three main tasks—system identification, controller tuning, and verification.

The controller tuning part found the controller for which the constraint imposed on the second input signal u_2 range is satisfied more than necessary, while the optimized penalization q_2 is not zero. This behavior is caused by a small influence of this penalization on the overall closed loop performance, and by the fact that the optimization method stops when gradient of the objective is too small. The starting point q_0 of the optimization method was chosen as the highest reasonable value for which the controller can still effectively control the system as described in Section 4.2.2.

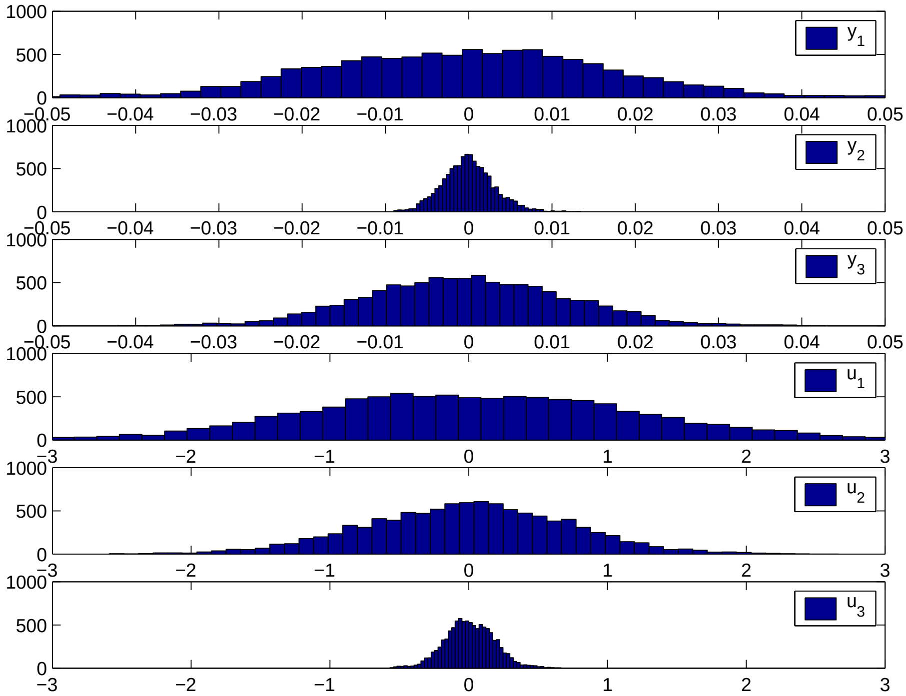

Figure 5.9: Histograms of signal values for the simulation run in tuning phase.

The verification shows that the signals obtained from the control process in the Simulink use smaller ranges than the predicted one, Table 5.2.2, therefore, the constraints are satisfied perfectly. This deviation from the expected behavior is caused by the model identification inaccuracy, though, the resulta are considered to be satisfactory.

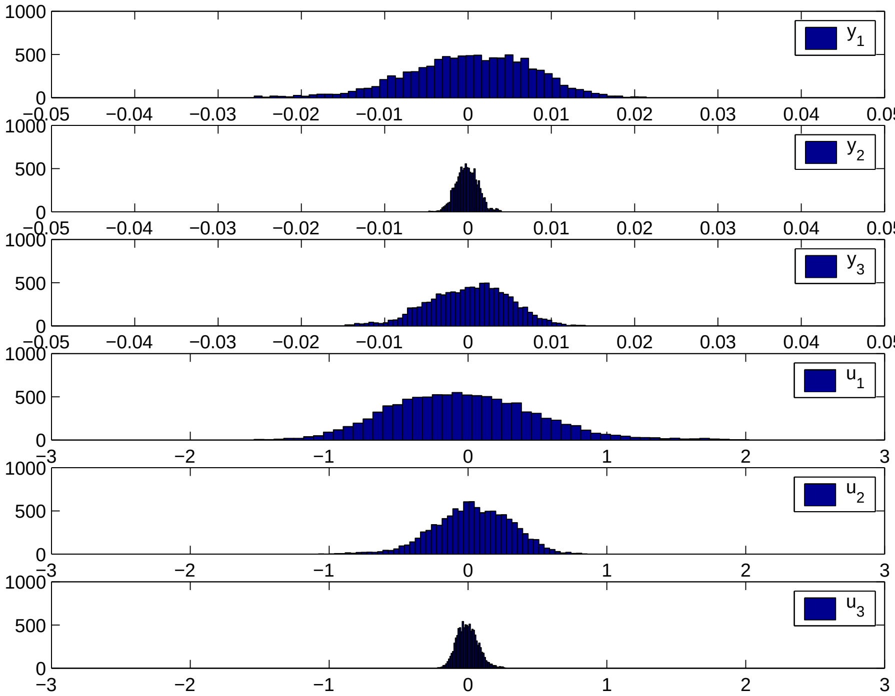

Figure 5.10: Histograms of signal values for the Simulink verification run.

(a) Signal variances

	predicted	verified
y_1	0.0181	0.0075
y_2	0.0033	0.0013
y_3	0.0129	0.0048
u_1	1.2246	0.5214
u_2	0.7081	0.2745
u_3	0.1845	0.0761

(b) Relative frequencies constraint satisfaction

	predicted	verified
u_1	0.94	1.00
u_2	1.00	1.00
u_3	0.88	1.00

Table 5.2: Comparison of the optimization result prediction with the Simulink model verification of the experiment.

Chapter 6

Conclusions

The controller tuning is a necessary part of any controller design. Though, the tuning is still not fully explored for the model based controllers. The presented work contributes to this topic by developing a methodology for tuning of the adaptive LQG controller with constraints imposed on its external variables and under uncertain knowledge of the system.

The thesis, finalizing author's previous work [?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?], is focused on the adaptive LQG controller tuning. Multiple constraints and multiple input and output variables are considered. The method employs the Bayesian approach to deal with the uncertainty contained in imprecise knowledge of the controlled system. The obtained controller is calculated with respect to this uncertainty and takes into account whole range of uncertain parameters. The other methods used are the Monte Carlo for controller quality evaluation and numerical optimization for the tuning itself.

The proposed methods and algorithms were implemented as the software toolbox Designer and successfully tested on several complex experimental models. The important contribution of the thesis is a step towards freeing the control engineers from manual controller tuning and to support usability of the model based controller in the practice.

6.1 Summary of Contributions of this Thesis

This section summarizes the most important assets of this work in a list of points. The main point is given in the following sentence:
The automated design algorithm of multivariate adaptive controller for stochastic and incompletely known systems with constraints has been developed.

Some of the most interesting parts of the presented results follow:

Controller tuning The task of controller tuning is to transform the user specified requirements into the values of the tuning parameters. The desired closed loop behavior, considered in this work, is to minimize the output error while fulfilling the constraints imposed on the action quantities. To express the desired closed loop behavior, the optimality conditions are translated into the form of so called quality functions. The constraints are handled differently in the conditions of the noise compensation and the reference setpoint tracking tasks.

Respecting incomplete knowledge of controlled system
The system knowledge is incomplete. The Bayesian estimation delivers the parameters not as known numbers but represented by a pdf. The tuning is done for the whole class of possible models thus it takes into account the uncertainty. The description of the uncertainty of the controlled system allows to tune the adaptive controller, which is being adapted while the uncertain model is simulated.

Generalization to multivariate controller and multiple constraints
The important contribution of this work is extending the tuning to the multiple input multiple output (MIMO) controllers, where multiple constraints on particular quantities are considered simultaneously.

Use of on-line stopping rules Computational feasibility of Monte Carlo evaluation was improved by employing on-line stopping rules for speeding simulations. The general approach in [**?**] was extended to stopping rules for Markov chains. The rules are based on measuring contribution of new data samples to the precision of calculated quantity.

Complete controller design was developed The presented work of controller tuning complements the results in the field of system identification and uncertainty handling achieved in the Department of Adaptive Systems in Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic and thus it allows the development of the complete controller design.

The model used for tuning is the black box model, where the only connection to the real system is through identification data and prior information. This approach is advantageous for complex systems where the physical structure of controlled system is hidden or it is too complex to be used with a model based controller.

It has to be noted that the capabilities of the tuning algorithm are dependent on a good model and realistic constraints that it is possible to satisfy. The proposed methodology optimizes the designed controller. Thus it does not guarantee the desired properties, rather it finds the best controller possible from the class of considered controllers.

6.1.1 Methods

The techniques and methods used and developed for the controller tuning task are summarized below:

- Majority of methods used is derived from the general Bayesian decision making theory.
- Monte Carlo method was used for sampling complex distributions.
- The constrained multivariate stochastic optimization task was solved using the sample path method.
- A reasonable initial approximation of the searched tuning parameter values for LQG controller was given.

6.1.2 Algorithms, Software, and Experiments

The methods and algorithms presented in this thesis are accompanied by the respective software implementation forming the toolbox Designer. The Designer is included in the Jobcontrol toolbox, which consistently unifies all the steps needed for the controller design starting from data preprocessing, identification, controller tuning and others. The presented experiments were performed using the Jobcontrol toolbox and they show the applicability of the proposed methods.

The Designer toolbox is developed for general use of controller design. Currently the LQG controller is only supported, but the construction of the toolbox is general enough to include other types of controllers.

6.2 Possible Future Work

The quality of the resulting designed controller is very dependent on the quality of model estimate. The use of the Gaussian ARX model is planned to be extended to the mixture of Gaussian models [**?**], which is capable to better approximate a much wider class of systems. The controllers used for the mixture models are based on the fully probabilistic design. The tuning of this kind of controller forms a natural continuation of this work.

Bibliography

[1] IFAC workshop on digital control. past, present and future of PID control. Barcelona, Spain, April 5–7, 2000.

[2] G. A. Montague D. P. Searson, M. J. Willis. Evolutionary design of process controllers. pages 1456–1461, Seansea, UK, 1998.

[3] E. J. Davison. *Benchmark Problems for Control System Design.* IFAC report of IFAC Theory Committee, 1990.

[4] W. Grega. Integrated environment for real-time control and simulation. *Computers in Industry*, 31:3–14, 1996.

[5] A. F. Lynch. Internal model-based controller design using measured costs and gradients. *International Journal of Control*, 69(2):257–270, 1998.

[6] D. J. N. Limebeer M. Green. *Linear robust control.* Prentice-Hall, Inc., 1995.

[7] N. Khailova L. Pavelkova M. Kárný, P. Nedoma. Use of prior information in structure estimation. *IEE Proceedings – Control Theory and Applications.*, 2003. This issue.

[8] J.M. Maciejowski. *Predictive Control with Constraints.* Prentice Hall, England, 2002.

[9] The Matlab Inc. *Optimization Toolbox for Matlab*, volume MA 01760-2098. Natick, 2001.

[10] M. J. Pont. Control systems design using design real–time patterns. pages 1078–1083, Seansea, UK, 1998.

[11] S. M. Robinson. Analysis of sample-path optimization. *Mathematics of Operations Research*, 21(3):513–528, 1996.

[12] J. Rojíček. *Controller design under constraints and incomplete knowledge of the system.* P.O.Box 18, 182 08 Prague, Czech Republic, 1998.

[13] I.-J. Wang and J. C. Spall. A constrained simultaneous perturbation stochastic approximation algorithm based on penalty functions. In *Proceedings of the American Control Conference*, pages 393–399, 1999.

[14] K. Zhou, J. C. Doyle, and K. Glover. *Linear Controller Design - Limits of Performance.* Prentice Hall, New Jersey, 1991.

[15] K. Zhou, J. C. Doyle, and K. Glover. *Robust Optimal Control.* Prentice Hall, New Jersey, 1996.

[16] Y. Zhu, E. T. Serrano, and S. Weiland. Towards a multivariable autotuner. In IFAC, editor, *IFAC World Congress, Preprints.* IFAC, Prague, 2005.