Czech Technical University in Prague

Faculty of Electrical Engineering

Doctoral Thesis

Peter Matisko

Prague, 2013

"*There is nothing more practical than a good theory*."

Kurt Lewin

Czech Technical University in Prague

Faculty of Electrical Engineering

Department of Control Engineering

Estimation of the stochastic properties of controlled systems

Doctoral thesis statement for obtaining the academic title of “Doctor”, abbreviated to “Ph.D.”

Ph.D. Programme: Electrical Engineering and Information Technology
Branch of study: Control Engineering and Robotics

Supervisor: Prof. Ing. Vladimír Havlena, CSc.

Prague, May 2013

Abstract

The doctoral thesis covers a part of the stochastic properties identification for linear dynamic systems. Knowledge of the noise sources and uncertainties is essential for the state estimation performance. The covariances of the process and measurement noise represent tuning parameters for the Kalman filter and the state estimation quality depends directly on them. The thesis deals with estimation of the noise covariances from the measured data. A Bayesian approach together with Monte Carlo methods are employed for this task. The thesis describes optimality tests that can be used to evaluate the quality of the state estimates obtained by a Kalman filter. A new approach was introduced to detect the color property of the process noise. If the process noise is colored, the shaping filter can be found in the time or frequency domain. It can be added to the Kalman filter which can be then tuned optimally. The limitations of the noise covariance estimation are evaluated by the Cramér–Rao bounds. The convergence of the proposed algorithms and the previously published ones were compared.

Key words: Stochastic systems, Kalman filter, Noise covariance estimation, Optimality tests, Cramér–Rao bounds, Colored noise detection, Shaping filter identification, Bayesian approach, Monte Carlo.

Acknowledgement

At the very first place, I would like to thank Prof. Vladimír Havlena for his care and supervising during past years. Special thank goes to my wife Gabriela, my parents and my colleagues and friends Samuel Prívara, Jiří Řehoř and Ondřej Šantin for the support of my work.

This thesis was financed by the following grants SGS: SGS10/197/OHK3/2T/13 and SGS10/283/OHK3/3T/13; GACR: P102/08/0442 and P103/11/1353; Honeywell donation program; UCEEB: CZ.1.05/2.1.00/03.0091.

TABLE OF CONTENT

1. Introduction

In the second half of the 20th century, a new state–space theory about dynamic systems was developed. The pioneering papers of the new theory were written by Rudolf E. Kalman, (Kalman 1960, 1961). The new theory considers not only input–output behavior as previously used transfer functions, but also the trajectories of the internal states. The state–space (SS) approach allows incorporating the first principles to the mathematical models. If not all of the states are measurable, they can be estimated by state observers. This allows using, for example, a state feedback even if some of the states are hidden.

In the 60‘s and 70's, the new theory was being developed remarkably quickly and it brought a new potential for solving various problems, especially in the field of control and regulation, (Kailath, 1979; Anderson and Moore, 1979; Kailath, *et al*. 2000; Gibbs, 2011). High computational power offers employing sophisticated control algorithms that can significantly improve the control performance. During past decades, the Model Predictive Control (MPC) algorithms has begun to be widely used for control of large processes such as petro chemical industry and power plants. If the model of a real system is known, it allows predicting the state and output trajectories which can be further used for the control optimization. To obtain the state space models, many identification methods have been developed, e.g. Subspace identification methods, Grey Box Modeling and others, (Ljung 1987; Katayama 2005; Řehoř, 2010, 2011). However, only minor attention was paid to identification of the stochastic part of the dynamic systems. Omitting uncertainties and noise properties can lead to significant limitation of the algorithms based purely on modeling of the deterministic part. As an example, consider a state estimator – Kalman filter. The quality of the state estimates depends on the knowledge of the deterministic part of the real system, but also on the knowledge of the stochastic properties and noise entering the process.

Throughout the thesis, linear dynamic systems affected by a Gaussian noise will be considered as the models of real processes. The corresponding state estimator for a linear system is given by the system matrices and the noise properties represented by covariance matrices.

The problem of finding a model of the deterministic part is solved by various identification methods. Examination and identification of the stochastic properties of real processes is studied in this thesis. If the real process is well identified, the deterministic as well as the stochastic part, then the predictions of such a model are accurate even for longer horizons. This leads to better estimation and prediction of the hidden system states and outputs. Accurate estimates/predictions can be used by controllers leading to better performance of the real process, less energy consumption, less pollution and increased safety which are the main goals of present technology.

1.1 State of the art

Methods for identification of linear dynamic systems are being developed for over one century. At the beginning, input–output relation represented by transfer functions was used. In the 60's, the state space methods became popular and quickly improved. However, only minor attention was paid to the identification of the stochastic part of a system and noise properties.
In the 60's, the pioneering papers were written about the estimation of noise covariances, (Mehra 1970, 1972; Carew and Belanger 1973; Neethling and Young 1974; Belanger 1974). It was observed that the autocorrelation of the output prediction errors is linearly depended on the covariance matrices describing the entering Gaussian noise. Then, for several decades, this topic was quite overlooked by the control science. Some research was done within the fields of speech, acoustics, signal processing and statistics, but the knowledge was not sufficiently applied to solve the problems of the system and control theory.

The latest methods concerning the estimation of noise covariances were published in years 2005–2009, (Odelson, *et al*. 2005; Akesson, *et al*. 2008; Rajamani and Rawlings, 2009; Duník, *et al*. 2009). The main contribution of the recent papers was algorithms that offer a solution for finding the noise covariance matrices in the explicit form. The original paper was written by Odelson *et al*. and the further publications offer several extensions and modification of the original approach. The last mentioned reference, (Duník, *et al*. 2009), offers a comparison and discussion over the methods for the estimation of noise covariances.

Another approach is described in Pour *et al*. (2009) which describes a method for estimation of the covariance matrix using the innovation form of a linear system. This paper also proves the consistency of the estimation process. However, the initial Kalman gain is assumed to be a priori known for the given system, which simplifies the problem.

1.2 Goals of the thesis and main contributions

The goal of the thesis is developing new approaches and algorithms for analysis and identification of the stochastic part of dynamic systems. The main goal is to develop a Bayesian algorithm for estimation of the noise covariance matrices. For the analysis purposes, the models will be considered as discrete, linear and affected by Gaussian noise.

The main results of the thesis are separated into the individual chapters. The first main contribution of the thesis is covered in Chapter 3. It solves a question, whether the filter performance is optimal or not, i.e. whether the quality of the state estimates can be improved. The question is answered by examining the sequence of output prediction errors, which is the only measurable value. If the output prediction errors form a white sequence, than it holds, that the filter performance is optimal. Chapter 3, therefore, solves a problem if the given sequence is white or colored. Several different methods are described and compared to the widely cited method published by Mehra in 1970. The optimality tests are then used together with the noise covariance estimating algorithms as a form of adaptive Kalman filter.

The second part of the thesis contains a detailed description of the Bayesian approach used to estimate the covariance matrices of the entering noise from the given output data. Bayesian theory is used together with Monte Carlo numerical methods. Several modifications are discussed for this method, and an extensive comparison to the previously published algorithms is given.

The third part of the thesis provides an overview about the colored noise. In the typical application of the Kalman filter, it is assumed that the entering noise is white. However, this is not necessarily true, and neglecting the color property may lead to the poor state estimates. Chapter 5 discusses how to detect whether the entering noise is colored or white using the time and the frequency domain. It also discusses whether it is possible to distinguish between colored process noise and colored measurement noise. The chapter further offers an overview

of possible solutions. It also contains several numerical examples and highlights a significant potential of further research on this field.

The last part derives the Cramér–Rao bound for the estimation of the noise covariances. This bound represent the limitation of the estimation algorithms and can provide overview about possible convergence rates for any new approaches solving the estimation of the noise covariances. A numerical example demonstrates the performance of the Bayesian algorithm and the recent methods and compares the results to the Cramér–Rao bound.

Summary of the thesis goals

1) Analyze stochastic properties of linear dynamic systems.

2) Summarize, develop and compare algorithms for performance evaluation of a Kalman filter.

3) Develop an approach for estimation of the noise covariance matrices.

4) Analyze linear systems affected by colored noise. Develop a method for detection of a colored noise. Develop an approach that deals with the colored noise and demonstrate a potential of using a noise shaping filter.

5) Analyze the limits for quality of the noise covariance estimation algorithms using Cramér-Rao bounds.

The results of the thesis were presented at most impact IFAC conferences: 16th IFAC System Identification Symposium 2012, 18th IFAC World Congress 2011 and IFAC Workshops on Adaptation and Learning in Control and Signal Processing 2010. The main results were published in *International Journal of Adaptive Control and Signal Processing*. Almost all parts of the following text were reviewed by international reviewers and were accepted for publishing in international proceedings and journals. The paper Matisko and Havlena (2012) has already one SCI citation.

2. State and its estimation

This chapter provides a brief introduction to the linear system theory and filtering. The definitions given further will be used throughout the thesis. Further information about the linear system theory can be found in Kailath (1979), Anderson and Moore (1979) or Antsaklis and Michel (1997).

2.1 Linear stochastic systems

A linear stochastic system of order n, with p– and r–dimensional stochastic inputs, m deterministic inputs and r outputs (the dimension of $\mathbf{y}(t)$ is the same as the dimension of $\mathbf{e}(t)$), can be defined as a state–space model of the form

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{G}\mathbf{v}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{e}(t) + \mathbf{D}\mathbf{u}(t), \end{aligned} \tag{2.1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is a state vector, $\mathbf{y}(t) \in \mathbb{R}^r$ is an output vector and $\mathbf{u}(t) \in \mathbb{R}^m$ is a deterministic input. Further, $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a system matrix, $\mathbf{C} \in \mathbb{R}^{r \times n}$ is an output matrix, $\mathbf{G} \in \mathbb{R}^{n \times p}$ is the noise input matrix and $\mathbf{B} \in \mathbb{R}^{n \times m}, \mathbf{D} \in \mathbb{R}^{r \times m}$ are the deterministic input matrices. In all the examples, system (2.1) is considered to be stable with zero deterministic input $\mathbf{u}(t)$. For the ease of analysis of the stochastic properties, the deterministic input $\mathbf{u}(t)$ and matrix $\mathbf{B}$ will be omitted. The stochastic inputs are $\mathbf{v}(t) \in \mathbb{R}^p$, $\mathbf{e}(t) \in \mathbb{R}^r$ with properties

$$\begin{bmatrix} \mathbf{v}(t) \\ \mathbf{e}(t) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{S}^T & \mathbf{R} \end{bmatrix} \right), \tag{2.2}$$

where $\mathbf{Q} \in \mathbb{R}^{p\times p}$ is the covariance matrix of the process noise, $\mathbf{R} \in \mathbb{R}^{r\times r}$ is the covariance matrix of the measurement noise and $\mathbf{S} \in \mathbb{R}^{p\times r}$ is the cross–covariance matrix. Symbol $\mathcal{N}(\boldsymbol{\mu};\mathbf{R})$ denotes the normal distribution with the mean $\boldsymbol{\mu}$ and the positive definite covariance matrix $\mathbf{R}$.

The model of the deterministic part (matrices **A**, **B**, **C**, **D**) of a real process can be obtained by state–space identification methods. A number of publications deals with this topic, (Ljung, 1987; Katayama, 2005). The model accuracy and knowledge of the noise properties influence the KF performance. The definition of the optimal Kalman filter is given by the lowest trace of the state prediction error. However, this criterion cannot be evaluated directly because the states are not measurable. For this reason, the filter performance is evaluated by analyzing the output prediction errors. The methods for analyzing an output sequence is described in Chapter 3. This thesis concentrates on estimating the noise properties to assure the KF performance be close to the optimum. Therefore, we assume the model of the deterministic part to be known. It is clear that every model is only an approximation of the real process. For this reason, it is not possible to find the "true" system parameters or the noise characteristics. The goal is to find such mathematical description of the real process that can sufficiently model and predict the real values of states or outputs. The term *optimal Kalman filter* is thus understood as the filter with the best possible quality of the state estimation that can be verified by examining the output prediction error and its whiteness property.

Matrix **G** is not obtained by the identification methods, but can be set according to the prior information about the noise structure. If there is no prior knowledge about the noise, **G** can be considered to be a unit matrix of order *n*. However, if the number of the noise sources is less than the system order, estimation of the process noise properties is less complicated and requires a smaller amount of data.

A more general stochastic model considers the cross–correlation between the process and measurement noise.

It is known from the system theory that the stochastic system can be alternatively defined in an *innovation form*, (Ljung, 1987; Kailath, *et al*. 2000)

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{K}\boldsymbol{\varepsilon}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \boldsymbol{\varepsilon}(t), \end{aligned} \tag{2.3}$$

where $\mathbf{K} \in \mathbb{R}^{n \times r}$ is a Kalman gain and $\boldsymbol{\varepsilon}(t) \sim \mathcal{N}\left(0, \mathbf{R}_{\varepsilon}\right), \mathbf{R}_{\varepsilon} \in \mathbb{R}^{r \times r}$ is a white innovation sequence. For the analysis purposes, the initial conditions might be neglected leading to matrices $\mathbf{K}, \mathbf{R}_{\varepsilon}$ being time invariant. It holds that if $\mathbf{K}$ in (2.3) is the optimal Kalman gain obtained by solving algebraic Riccati equation for system (2.1) and $\mathbf{R}_{\varepsilon}$ is the covariance matrix of innovations, than the output spectral density of system (2.1) and (2.3) is the same. It also means, that given system (2.1) with two noise source with covariances $\mathbf{Q}$ and $\mathbf{R}$, it is possible to find a system in the form (2.3) with parameters $\mathbf{K}$ and $\mathbf{R}_{\varepsilon}$ such that the output spectral density is the same, (Havlena, Štecha, 1999; Kailath, *et al*., 2000).

Some comments can be added on both definitions. System (2.1) has two independent sources of uncertainty, the process noise and the measurement noise. In a non–scalar case, the variables $\mathbf{Q}$, $\mathbf{R}$ are matrices and contain redundant information. Noise covariance matrices represent properties of the noise sources including their structure. Matrix $\mathbf{Q}$ have the same dimensions as the number of states (if $\mathbf{G}$ is unit) and matrix $\mathbf{R}$ has a dimension equal to the number of outputs. However, from the observed output data, only the number of noise sources equal to the number of outputs can be recognized. Therefore, the model (2.3) is minimal in the sense of the number of noise sources and it generates the same output spectral density as the model (2.1), provided that $\mathbf{K}$ is the optimal Kalman gain for the system (2.1). A disadvantage of the innovation form comes from the fact that we lose the physical background of the noise sources and their structure. Another complication arises from the direct estimation of the Kalman gain $\mathbf{K}$ because it is required the matrix $\mathbf{A} - \mathbf{KC}$ be stable. A method that estimates the innovation covariance matrix is presented in Pour *et al*. (2009), where also the consistency of its estimation is proven. However, the gain $\mathbf{K}$ is considered to be a priori known. This assumption simplifies the problem. Another possibility for finding the gain $\mathbf{K}$ is using the Subspace Identification methods that can identify the system model together with the Kalman filter, (Katayama, 2005). However, this approach can lead to problems with stability of $\mathbf{A} - \mathbf{KC}$. For these reasons, this thesis concentrates on finding the noise covariance matrices that parameterize the stabilizing Kalman gain.

2.2 State estimation and the Kalman filter

The Kalman filter (KF) is an optimal state estimator for a linear dynamic system, (Kalman, 1960; Kalman and Bucy, 1961; Anderson and Moore, 1979). It is optimal in the sense of minimal state prediction error, mathematically defined as a minimal trace of the state prediction error covariance matrix. If the entering noise is normal, the filter is optimal in the *mean square* (MS) sense. Otherwise, the filter is optimal in the *linear mean square* (LMS) sense.

The Kalman filter can be alternatively derived using Bayesian principles and probability distributions. The filter updates the conditional probability distribution function (cpdf) $p(\mathbf{x}(t) \mid \mathcal{Y}^{\tau})$ of the state of a linear dynamic system conditioned by the given data $\mathcal{D}^{\tau} = \left\{\mathbf{u}(0), \mathbf{y}(0), \mathbf{u}(1)\mathbf{y}(1), \ldots, \mathbf{u}(\tau), \mathbf{y}(\tau)\right\}$, or alternatively $\mathcal{Y}^{\tau} = \left\{\mathbf{y}(0), \mathbf{y}(1), \ldots, \mathbf{y}(\tau)\right\}$, up to the time τ. The cpdf of the state can be expressed recursively due to the Markov property of the dynamic system (2.1), (Peterka, 1981). The Markov property of the state can be expressed as

$$p\left(\mathbf{x}(t+1), \mathbf{y}(t) \middle| \mathbf{x}(t), \mathcal{D}^{t-1}\right) = p\left(\mathbf{x}(t+1), \mathbf{y}(t) \middle| \mathbf{x}(t)\right), \tag{2.4}$$

which means that the state $\mathbf{x}(t)$ contains all the information from the previous data $\mathcal{D}^{t-1}$ up to the time $t-1$.

The estimates at time t obtained from the measurements up to time τ use the index $\left(t \mid \tau\right)$. For a linear time invariant system and a normal prior, the cpdf $p(\mathbf{x}(t) \mid \mathcal{Y}^{t-1})$ is the normal distribution $\mathcal{N}\left(\hat{\mathbf{x}}(t \mid t-1); \mathbf{P}(t \mid t-1)\right)$. The parameters of this distribution can be recursively calculated by the Kalman filter defined as

$$\hat{\mathbf{x}}(t+1 \mid t) = \mathbf{A}\hat{\mathbf{x}}(t \mid t-1) + \left(\mathbf{AP}(t \mid t-1)\mathbf{C}^{T} + \mathbf{S}\right)\left(\mathbf{CP}(t \mid t-1)\mathbf{C}^{T} + \mathbf{R}\right)^{-1} \boldsymbol{\varepsilon}(t \mid t-1), \tag{2.5}$$

$$\begin{aligned} \mathbf{P}\left(t+1 \mid t\right) = {} & \mathbf{AP}\left(t \mid t-1\right)\mathbf{A}^{T} + \mathbf{Q} \\ & - \left(\mathbf{AP}\left(t \mid t-1\right)\mathbf{C}^{T} + \mathbf{S}\right)\left(\mathbf{CP}\left(t \mid t-1\right)\mathbf{C}^{T} + \mathbf{R}\right)^{-1}\left(\mathbf{CP}\left(t \mid t-1\right)\mathbf{A}^{T} + \mathbf{S}^{T}\right), \end{aligned} \tag{2.6}$$

where the following term is called the Kalman gain

$$\mathbf{K}(t \mid t-1) = \Big(\mathbf{P}\big(t \mid t-1\big)\mathbf{C}^T + \mathbf{S}\Big)\Big(\mathbf{C}\mathbf{P}\big(t \mid t-1\big)\mathbf{C}^T + \mathbf{R}\Big)^{-1}$$

and $\mathbf{P}(t \mid \tau) \in \mathbb{R}^{n \times n}$ is a covariance matrix of the state prediction error, i.e.

$$\mathbf{P}(t \mid \tau) \triangleq \mathcal{E}\left\{\Big(\hat{\mathbf{x}}(t \mid \tau) - \hat{\mathbf{x}}(t \mid \tau)\Big)\big(\ldots\big)^T\right\},$$

and

$$\boldsymbol{\varepsilon}\big(t \mid t-1\big) \triangleq \mathbf{y}(t) - \mathbf{C}\hat{\mathbf{x}}(t \mid t-1) \tag{2.7}$$

is the output prediction error. In the optimal case, this is a white sequence called an *innovation sequence*.

If the cross–covariance matrix $\mathbf{S}$ is zero, the Kalman filter can be defined in two steps as follows, (Štecha and Havlena, 1999; Simon, 2006).
In the *data update step*, the state estimate $\hat{\mathbf{x}}(t \mid t-1)$ is updated using measured data at time *t*

$$\begin{aligned}\hat{\mathbf{x}}(t \mid t) &= \hat{\mathbf{x}}(t \mid t-1) + \mathbf{P}\big(t \mid t-1\big)\mathbf{C}^T\Big(\mathbf{C}\mathbf{P}\big(t \mid t-1\big)\mathbf{C}^T + \mathbf{R}\Big)^{-1}\boldsymbol{\varepsilon}\big(t \mid t-1\big),\\ \mathbf{P}\big(t \mid t\big) &= \mathbf{P}\big(t \mid t-1\big) - \mathbf{P}\big(t \mid t-1\big)\mathbf{C}^T\Big(\mathbf{C}\mathbf{P}\big(t \mid t-1\big)\mathbf{C}^T + \mathbf{R}\Big)^{-1}\mathbf{C}\mathbf{P}\big(t \mid t-1\big).\end{aligned} \tag{2.8}$$

In the *time update step*, the state prediction $\hat{\mathbf{x}}(t+1 \mid t)$ is calculated for the next time instant $t+1$

$$\begin{aligned}\hat{\mathbf{x}}(t+1 \mid t) &= \mathbf{A}\hat{\mathbf{x}}(t \mid t),\\ \mathbf{P}\big(t+1 \mid t\big) &= \mathbf{A}\mathbf{P}\big(t \mid t\big)\mathbf{A}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T.\end{aligned} \tag{2.9}$$

Taking this equality into account, the probability distribution function of the output $\mathbf{y}(t)$ conditioned by data $\mathcal{Y}^{t-1}$ and the covariance matrices $\mathbf{Q}$, $\mathbf{R}$ is given by

$$\begin{aligned}p\Big(\mathbf{y}(t) \mid \mathcal{Y}^{t-1}\Big) &= \int p\Big(\mathbf{y}(t) \mid \mathbf{x}(t)\Big)\,p\Big(\mathbf{x}(t) \mid \mathcal{Y}^{t-1}\Big)\,d\mathbf{x}(t) =\\ &= \mathcal{N}\Big(\hat{\mathbf{y}}\big(t \mid t-1\big);\, \mathbf{P}_{yy}\big(t \mid t-1\big)\Big),\end{aligned} \tag{2.10}$$

where the output prediction and its covariance matrix $\mathbf{P}_{yy}\big(t+1 \mid t\big) \in \mathbb{R}^{r \times r}$ are calculated by

$$\begin{aligned}\hat{\mathbf{y}}\left(t \mid t-1\right) &= \mathbf{C}\hat{\mathbf{x}}\left(t \mid t-1\right),\\ \mathbf{P}_{yy}\left(t \mid t-1\right) &= \mathbf{C}\mathbf{P}\left(t \mid t-1\right)\mathbf{C}^T + \mathbf{R}.\end{aligned} \tag{2.11}$$

Further, consider a linear system in the innovation form (2.3), (Anderson and Moore, 1979; Kailath, at al. 2000), having as many noise sources as the number of outputs. Obviously, the process noise $\mathbf{K}\boldsymbol{\varepsilon}(t)$ and the measurement noise $\boldsymbol{\varepsilon}(t)$ sources are correlated. The entering noise can be described by the covariance matrices (2.2) as

$$\begin{bmatrix}\mathbf{v}(t)\\ \mathbf{e}(t)\end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\mathbf{K}\mathbf{R}_{\varepsilon}\mathbf{K}^T & \mathbf{K}\mathbf{R}_{\varepsilon}\\ \mathbf{R}_{\varepsilon}\mathbf{K}^T & \mathbf{R}_{\varepsilon}\end{bmatrix}\right). \tag{2.12}$$

The relationship between the forms (2.1) and (2.3) can be derived as follows. Consider the system (2.1) to be completely known. Then, the innovation sequence for $\mathbf{y}(t)$ can be recursively calculated using a Kalman filter as follows

$$\begin{aligned}\hat{\mathbf{x}}(t+1 \mid t) &= \mathbf{A}\hat{\mathbf{x}}(t \mid t-1) + \mathbf{K}\boldsymbol{\varepsilon}(t \mid t-1),\\ \boldsymbol{\varepsilon}(t \mid t-1) &= \mathbf{y}(t) - \mathbf{C}\hat{\mathbf{x}}(t \mid t-1),\end{aligned} \tag{2.13}$$

where $\mathbf{K}$ is the optimal Kalman gain obtained by solving the algebraic Riccati equation. The first equation can be modified considering the definition of innovations $\boldsymbol{\varepsilon}(t \mid t-1) \triangleq \mathbf{y}(t) - \mathbf{C}\hat{\mathbf{x}}(t \mid t-1)$ resulting in the form

$$\begin{aligned}\hat{\mathbf{x}}(t+1 \mid t) &= \left(\mathbf{A} - \mathbf{K}\mathbf{C}\right)\hat{\mathbf{x}}(t) + \mathbf{K}\mathbf{y}(t),\\ \boldsymbol{\varepsilon}(t) &= -\mathbf{C}\hat{\mathbf{x}}(t \mid t-1) + \mathbf{y}(t),\end{aligned} \tag{2.14}$$

which is the standard form of a state–space description for a linear system. The system (2.14) is entered by a colored signal $\mathbf{y}(t)$. The output of this system is an innovation sequence, which is white. Therefore, the system (2.14) is called a whitening filter. It holds that the system (2.3) is causally invertible for the process $\mathbf{y}(t)$, (proof in Kailath, *et al*. 2000, Chap 9.). For all triplets of covariances $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{S}$ in (2.2) for system (2.1), it is possible to find $\mathbf{K}$, $\mathbf{R}_{\varepsilon}$ such that the matrix $\mathbf{A} - \mathbf{K}\mathbf{C}$ is stable, and the spectral density of the outputs of the systems (2.1) and (2.3) is equal. The gain $\mathbf{K}$ in the system (2.3) is then equal to the optimal Kalman gain obtained by solving the Riccati equation. For this reason, the variable $\mathbf{K}$ is used in both cases.

The Kalman filter is optimal only if the following conditions hold

- the real system is linear, time invariant and its state space model is exactly known,
- the process and the measurement noise are white processes,
- the process and the measurement noise covariance matrices are known.

In this thesis, the stochastic properties are of interest. Identification methods are a well studied field that are beyond the objective of this thesis. Therefore, throughout the thesis, the system matrices are assumed to be known after the identification of a real process. Several approaches for the noise covariance matrix estimation will be introduced and compared to the earlier published methods. In the following chapter, we will concentrate on the Kalman filter performance and optimality.

3. Optimality tests for the Kalman filter[1]

3.1 Introduction to the optimality analysis of the Kalman filter

It was mentioned in the previous chapter that the Kalman filter is an optimal state estimator under certain conditions. The Kalman filter is a state observer and intuitively, we expect to obtain "the best" state estimates. A criterion that is used to derive the Kalman filter considers the state prediction error (difference between the actual state value and the state estimate) which should be minimal. However, as the real process states are hidden, it is not possible to evaluate the state estimation quality directly. The only available data is the measured output and the output prediction error. It is known from the estimation theory that if the Kalman filter works optimally, the output prediction error sequence is white. Therefore, to evaluate the quality of state estimates, the whiteness property of the output prediction error sequence is used.

Inspecting the whiteness property of a sequence has been well studied by mathematicians and the results have been applied in the field of acoustics and signal processing. However, these methods are not widely used in the system and control field, and inspection of the Kalman filter quality is often done ad hoc without any analytical tools.

This chapter provides derivation of several whiteness tests and compares them by an application in the state estimation problems. The properties of proposed methods are discussed, and their performance is extensively tested on simulations.

[1] This chapter is a part of the paper originally published in IFAC–PapersOnLine, www.ifac–papersonline.net, and is reproduced here with permission from IFAC, (Matisko and Havlena 2012).

3.2 Autocorrelation function

An autocorrelation function is the cross–correlation of the sequence with itself, (Åström, 1970). For an infinite sequence of real numbers generated by a stationary ergodic process $y(t)$ with zero mean, it is defined as

$$Y_k = \mathcal{E}\Big\{y(t)y(t+k)\Big\}, \tag{3.1}$$

where $\mathcal{E}\Big\{\ \Big\}$ is the expected value operator. From this point further, the main interest is paid to white noise. It is known, that the autocorrelation function of the white noise sequence is a multiple of Kronecker delta function, (Papoulis, 1991). That means $Y_k = 0, \text{for } k \neq 0$.

The definition (3.1) requires an infinite set of data and knowledge of the probability distribution of $y(t)$. However, the given data represents only one realization. To use the single realization for the estimation of autocorrelation function, the ergodicity of the generating process must be assumed. The two formulas ((3.2) and (3.3)) are used to estimate the autocorrelation function, (Mehra, 1970; Pukkila and Krishnaiah, 1988a; Fuller, 1996; Odelson, *et al.* 2005). However, none of these formulas returns samples that are independent and identically distributed (i.i.d.) for every $k \geq 1$. This property is necessary for hypotheses testing, otherwise the tests loose reliability. In this chapter, the two models will be considered, and a new formula will be derived, that generates i.i.d. samples for every $k \geq 1$, (Matisko and Havlena, 2012a). Assume having a set of data with length N. The unbiased estimate of the autocorrelation function is given by

$$\hat{Y}_k^{N-k} \triangleq \frac{1}{N-k}\sum_{i=1}^{N-k} y(i)y(i+k). \tag{3.2}$$

In Mehra (1970) and Fuller (1996), it is stated that the following definition gives less mean square error and should be preferred to (3.2)

$$\hat{Y}_k^{N} \triangleq \frac{1}{N}\sum_{i=1}^{N-k} y(i)y(i+k). \tag{3.3}$$

The normalized autocorrelation functions are defined as

$$\hat{\Gamma}_k^N \triangleq \frac{\hat{Y}_k^N}{\hat{Y}_0^N},\ \hat{\Gamma}_k^{N-k} \triangleq \frac{\hat{Y}_k^{N-k}}{\hat{Y}_0^{N-k}}. \tag{3.4}$$

Suppose having a sequence of N numbers generated from the normal distribution $\mathcal{N}(0,R)$, where R is an arbitrary covariance. Parameter M is the maximum desired lag of the autocorrelation function. The goal is to examine statistics of the function $\hat{\Gamma}_k$, for $k=1\ldots M$, that are asymptotically normal. Now, for each k, the statistics among all sequences are calculated. It is expected that for $k\geq 1$ it holds that $\mathcal{E}\left\{\hat{\Gamma}_k\right\}=0$ and $\operatorname{cov}\left\{\hat{\Gamma}_k\right\}=1/N$, (Odelson *et al.* 2005). We will show, that the normalized estimated autocorrelation functions (3.3), (3.4) do not satisfy the condition $\operatorname{cov}\left\{\hat{\Gamma}_k\right\}=1/N$. For the independent normal random sequences with $\mathbf{x}$, $\mathbf{y}$ zero mean it holds[2]

$$\begin{aligned}\mathcal{E}\left\{\mathbf{x}^T\mathbf{y}\right\} &= \mathcal{E}\left\{\mathbf{x}\right\}\mathcal{E}\left\{\mathbf{y}\right\},\\ \operatorname{cov}\left\{\mathbf{x}^T\mathbf{y}\right\} &= \operatorname{cov}\left\{\mathbf{x}\right\}\operatorname{cov}\left\{\mathbf{y}\right\}.\end{aligned}$$

Further, if $y\sim\mathcal{N}\left(0,R\right)$, then $\operatorname{cov}\left\{\sum_{i=1}^{N} y_i\right\}=NR$. Therefore, $\hat{Y}_k^{N-k}$ and $\hat{Y}_k^N$ can be expressed as random values generated approximately from the normal distribution

$$\hat{Y}_k^{N-k} \sim \frac{1}{N-k}\mathcal{N}\left(0,(N-k)R^2\right)=\mathcal{N}\left(0,\frac{1}{N-k}R^2\right), \tag{3.5}$$

$$\hat{Y}_k^{N} \sim \frac{1}{N}\mathcal{N}\left(0,(N-k)R^2\right)=\mathcal{N}\left(0,\frac{(N-k)}{N^2}R^2\right). \tag{3.6}$$

It can be seen, that in both cases, the variance varies with k and therefore the values of the estimated autocorrelation functions (3.4) are not i.i.d. The goal is to find a constant multiplying the sum in (3.2), that would assure the variance of $\hat{Y}_k^*$ to be independent of k. It must hold

[2] If the sequences **x**, **y** have a non-zero mean, than it holds
$\operatorname{cov}\{\mathbf{x}^T\mathbf{y}\}=\operatorname{cov}\{\mathbf{x}\}\operatorname{cov}\{\mathbf{y}\}+\operatorname{cov}\{\mathbf{x}\}\mathcal{E}\{\mathbf{y}\}^2+\operatorname{cov}\{\mathbf{x}\}\mathcal{E}\{\mathbf{y}\}^2$.

$$\hat{Y}_k^* \sim \frac{1}{D}\mathcal{N}\Big(0,(N-k)R^2\Big) = \mathcal{N}\left(0,\frac{1}{N}R^2\right). \tag{3.7}$$

It is obvious that $D = \sqrt{N(N-k)}$. The desired formula for the autocorrelation estimation and its normalized form is given by

$$\hat{Y}_k^* \triangleq \frac{1}{\sqrt{N(N-k)}}\sum_{i=1}^{N-k} y(i)y(i+k), \quad \hat{\Gamma}_k^* \triangleq \frac{\hat{Y}_k^*}{\hat{Y}_0^*}. \tag{3.8}$$

If it holds that $N \gg M$, then k samples may be omitted from the summation. The estimated autocorrelation function would be of the form

$$\hat{Y}_k^{N,M} \triangleq \frac{1}{(N-M)}\sum_{i=k+1}^{N-M+k} y(i)y(i+k), \tag{3.9}$$

where the summation sums $N-M$ numbers for each k. A disadvantage of this method is, that it does not use all of the available data.

An extension of the autocorrelation function to the multivariate variables can be expressed using the following definitions

$$\mathbf{Y}_k = \mathcal{E}\Big\{\mathbf{y}(t)\mathbf{y}^T(t+k)\Big\}, \tag{3.10}$$

$$\hat{\mathbf{Y}}_k^* \triangleq \frac{1}{\sqrt{N(N-k)}}\sum_{i=1}^{N-k}\mathbf{y}(i)\mathbf{y}^T(i+k) \tag{3.11}$$

that lead to the final normalized form, (Mehra, 1970; Lütkepohl, 2005)

$$\left[\hat{\Gamma}_k^*\right]_{ij} = \frac{\left[\hat{\mathbf{Y}}_k^*\right]_{i,j}}{\sqrt{\left[\hat{\mathbf{Y}}_0^*\right]_{i,i}\left[\hat{\mathbf{Y}}_0^*\right]_{j,j}}}, \tag{3.12}$$

where $[\cdot]_{i,j}$ represents the element of the matrix on the position i, j.

3.3 Optimality tests

It is known from the control theory, that if all optimality conditions given in Chapter 1 hold, the innovation sequence is white. Therefore, testing the optimal performance of the filter reduces to testing the whiteness property of the innovation sequence. The following tests are built on Wald (1945), Mehra (1970), Pukkila and Krishnaiah (1988a, 1988b), Rencher (2002), Seber (2008), Eshel (2012). The original text of this section can be found in paper Matisko and Havlena (2012a). The tests will be compared in Section 3.4.

3.3.1 Optimality Test 1

Test.1 It was shown in the previous section that the values of normalized autocorrelation function $\hat{\Gamma}_k^*$ are asymptotically i.i.d from the normal distribution with zero mean and variance 1/*N*. Hypotheses can be formulated as follows

- H_0: the values $\hat{\Gamma}_k^*, k \geq 1$ were generated from the normal distribution with zero mean and variance 1/*N*,
- H_1: the values $\hat{\Gamma}_k^*, k \geq 1$ were not generated from the normal distribution with zero mean and variance 1/*N*.

Based on the desired confidence level *I*, a quantile q_p of the normal distribution is taken. If the sequence was generated from a normal distribution with a variance 1/*N*, then *I* of the values must lie within the interval $\pm q_{(1-I)/2} / \sqrt{N}$. If, for example, the confidence level is $I = 0.975$, then 97.5% of the values $\hat{\Gamma}_k^*$, $1 \leq k \leq M$ have to lie within the interval $\pm q_{(1-I)/2} / \sqrt{N} = \pm 2.24 / \sqrt{N}$. If the number of values $\hat{\Gamma}_k^*, k \geq 1$ lying inside the confidence interval is higher than $I \cdot M$, than the null hypothesis is accepted and the sequence $\hat{\Gamma}_k^*$ is considered to be generated from the normal distribution with variance 1/*N*. This leads to the conclusion, that the given sequence $y(t)$ is uncorrelated white.

This optimality test was proposed by Mehra (1970) and is widely cited. However, according to the simulations, this test has significantly lower performance than the following ones. Therefore, it will not be further considered for testing and comparisons.

3.3.2 Optimality Test 2

Test.2 Hypotheses can be formulated as follows

H_0: the values $\hat{\mathbf{Y}}_k^*, k \geq 1$ were generated from the normal distribution with zero mean and variance 1/*N*,

H_1: the values $\hat{\mathbf{Y}}_k^*, k \geq 1$ were not generated from the normal distribution with zero mean and variance 1/*N*.

It is known, that a sum of squared independent values generated from the normal distribution $\mathcal{N}\left(0;1\right)$ is distributed according to the χ^2, (Rencher, 2002; Seber, 2008). Consider *M* values of the function $\hat{\mathbf{\Gamma}}_k^*$ and a sum of its squares for each diagonal member *j*

$$\Psi_j(M) = \sum_{i=1}^{M} \left(\left[\hat{\Gamma}_i^* \right]_{jj} \right)^2. \tag{3.13}$$

If $\hat{\mathbf{Y}}_k$ is an uncorrelated white sequence, values $\left[\hat{\Gamma}_k^*\right]_{jj}$ are normally distributed with zero mean and variance 1/*N*, (Mehra 1970, prove in reference [15] therein). Therefore, the function $N \cdot \Psi_j(M)$ represents random values from the $\chi^2(M)$ distribution with *M* degrees of freedom. Hypothesis H_0 is accepted, if for each *j* it holds

$$N\Psi_j\left(M\right) < q_I^{\chi^2(M)}, \tag{3.14}$$

where $q_I^{\chi^2(M)}$ is the quantile of χ^2 distribution with *M* degrees of freedom and the confidence level *I*. Furthermore, the values $N\Psi_j\left(M\right)$ can be used as a measure to characterize the distance between the actual setting and the optimum, which is zero. Alternatively, value $q_I^{\chi^2(M)}$

can be considered as a 'reachable' optimum for the given amount of data. Finally, if the hypothesis H_0 is accepted, it can be concluded that the given sequence $\hat{\mathbf{Y}}_k$ is an uncorrelated white sequence.

3.3.3 Optimality Test 3

Test.3 This test examines whether the given sequence comes from a non–zero order *l*–dimensional autoregressive (AR) process or not, (Lütkepohl, 2005). If the sequence is white, it can be stated that it was generated by AR model of order zero. The hypotheses are formulated as follows

H_0: the data generator is an AR process of order zero,
H_1: the data generator is an AR process of order one or higher with non-zero coefficients.

The multivariate AR model of order m is generally defined as

$$\sum_{i=0}^{m} \mathbf{A}_i \mathbf{x}(t-i) = \mathbf{e}(t), \tag{3.15}$$

where $\mathbf{A}_i \in \mathbb{R}^{l \times l}$ are such coefficients that the AR system is stable. In most practical cases the zero coefficient is unit, i.e. $\mathbf{A}_0 = \mathbf{I}$. System (3.15) with $\mathbf{A}_0 = \mathbf{I}$ is stable and stationary iff all the roots of

$$\det(\mathbf{I} - \mathbf{A}_1 z^{-1} - \ldots \mathbf{A}_m z^{-m}) = 0 \tag{3.16}$$

lie inside the the unit circle, i.e. $|z| < 1$, where $z \in \mathbb{C}$, (Lütkepohl, 2005; Hendry and Juselius, 2000).

Several criteria can be defined for finding m in a general form as

$$\psi(m) = N \log\left|\hat{\mathbf{S}}^2(m)\right| + m\, g(N), \tag{3.17}$$

where $g(N)$ is a nonnegative penalty term and $\left|\hat{\mathbf{S}}^2(m)\right|$ is a determinant of the residual covariance matrix dependent on the selected AR order m. The use of the penalization term has the following reason. Consider a discrete linear system of order one excited by white noise that has generated an output sequence with a length of 100 samples. Theoretically this sequence can be generated by a system of order 100. However, such model covers not only the real system dynamics, but also the dynamic of the entering noise. Therefore, the mean–square error cannot be the only criterion, because it would be possible to find a system generating the given sequence with zero error. It is necessary to find a compromise between the modeling error and the system order. Criterion in the form (3.17) penalizes the system order.

The residual covariance matrix can be expressed in the form

$$\hat{\mathbf{S}}^2(m) = \hat{\mathbf{Y}}_0^* - \hat{\mathbf{A}}_1\hat{\mathbf{Y}}_1^* - \ldots - \hat{\mathbf{A}}_m\hat{\mathbf{Y}}_m^*, \tag{3.18}$$

where $\hat{\mathbf{A}}(m) = \left[\hat{\mathbf{A}}_1, \hat{\mathbf{A}}_2, \ldots, \hat{\mathbf{A}}_m\right]^T$ are parameters obtained by solving the multivariate Yule–Walker equations (Pukkila and Krishnaiah, 1988b)

$$\underbrace{\begin{bmatrix} \hat{\mathbf{Y}}_0^* & \hat{\mathbf{Y}}_1^* & \cdots & \hat{\mathbf{Y}}_{m-1}^* \\ \hat{\mathbf{Y}}_1^* & \hat{\mathbf{Y}}_0^* & & \hat{\mathbf{Y}}_{m-2}^* \\ \vdots & & \ddots & \vdots \\ \hat{\mathbf{Y}}_{m-1}^* & \hat{\mathbf{Y}}_{m-2}^* & \cdots & \hat{\mathbf{Y}}_0^* \end{bmatrix}}_{\boldsymbol{\Psi}(m)} \underbrace{\begin{bmatrix} \hat{\mathbf{A}}_1 \\ \hat{\mathbf{A}}_2 \\ \vdots \\ \hat{\mathbf{A}}_m \end{bmatrix}}_{\hat{\mathbf{A}}(m)} = \begin{bmatrix} \hat{\mathbf{Y}}_1^* \\ \hat{\mathbf{Y}}_2^* \\ \vdots \\ \hat{\mathbf{Y}}_m^* \end{bmatrix}. \tag{3.19}$$

Matrix $\hat{\mathbf{S}}^2(m)$ can be shortly expressed as

$$\hat{\mathbf{S}}^2(m) = \hat{\mathbf{Y}}_0^* - \hat{\mathbf{A}}^T(m)\boldsymbol{\Psi}(m)\hat{\mathbf{A}}(m) \tag{3.20}$$

The penalty term depends on the selected criterion, e.g. Akaike information criterion (AIC), Bayesian information criterion (BIC), Efficient determination criterion (EDC), Hannan–Quin criterion (HQ) (see Pukkila and Krishnaiah, 1988a, 1988b for more details). In the following text, the penalty function is BIC and is of the form $g(N) = l^2 \log N,$ where *N* is the length of the given data and *l* is its dimension.

Determining the order of the AR process can be expressed by combining (3.17), (3.20) and the penalty term to the form

$$m^* = \arg \min_{0 \le m \le m_{\max}} \left(N \log \left| \hat{\mathbf{Y}}_0^* - \hat{\mathbf{A}}^T(m) \boldsymbol{\Psi}(m) \hat{\mathbf{A}}(m) \right| + m\, l^2 \log N \right). \tag{3.21}$$

The hypothesis H_0 is accepted if $m^* = 0$, and that imply that the given sequence $\hat{\mathbf{Y}}_k$ is uncorrelated white. Alternatively, a modified criterion can be defined as

$$T(m^*) = \min_{1 \le m \le m_{\max}} \left(0,\ N \log \left| \mathbf{I} - \hat{\mathbf{Y}}_0^{-1} \hat{\mathbf{A}}^T(m) \boldsymbol{\Psi}(m) \hat{\mathbf{A}}(m) \right| + m\, l^2 \log N \right). \tag{3.22}$$

The hypothesis H_0 is accepted if $T(m^*) > 0$. This test is more strict than (3.21), i.e. for H_0 to be accepted, the tested sequence have to be closer to white noise than in criterion (3.21).

Performance of the Kalman filter

If we have several Kalman filters at hand, and the best one is to be selected, the statistics of the optimality tests can be used. Usually, the system is not perfectly identified, therefore all the identified noise covariances can be non–optimal. In such case, the best settings should be chosen.

Considering *Test.2*, a qualitative measure can be defined as

$$M_{Test.2} = \frac{N}{l} \sum_{j=1}^{l} \Psi_j \left(M \right), \tag{3.23}$$

or as

$$M_{Test.2} = \max_{\forall j \le l} N \Psi_j \left(M \right), \tag{3.24}$$

where $\Psi_j \left(M \right)$ is given by (3.13). The first measure calculates an average value of statistic $N \Psi_j \left(M \right)$, while the second formula returns the worst case. There can be a difference when comparing these two results. If there is a strong correlation in one dimension of the tested signal, then the statistic (3.23) will be much lower than (3.24).

For *Test.3*, the qualitative measure can be defined as

$$M_{Test.3} = \max_{0 \le m \le m_{\max}} \operatorname{abs} \left\{ N \log \left| \mathbf{I} - \hat{\mathbf{Y}}_0^{-1} \hat{\mathbf{A}}^T \boldsymbol{\Psi} \hat{\mathbf{A}} \right| + m\, l^2 \log N \right\}. \tag{3.25}$$

For both measures it holds that a lower number means better performance. If the best Kalman filter is to be chosen, the qualitative statististics (3.23), (3.24) or (3.25) are calculated for the output prediction error of each individual Kalman filter. The best filter has the lowest value of the qualitative measure.

3.3.4 Optimality test 4 (Sequential test)

In this section, a sequential optimality test (*Test.4*) will be described, built on Wald (1945). It allows to control the errors of both kinds. The test works online in parallel with the system and as soon as there is enough data, it returns a decision about the filter optimality. The idea is analogous as in *Test.3*. The order of the data generating system is of interest. If it is of the order zero, the tested sequence can be concluded to be white.

The measured samples are recursively used for the data generator identification which is assumed to be an AR model of order *m*. However, there is a difficulty with determining the zero order system. If the system is of the order zero, and the model to be identified is of the higher order, it leads to the situation, when all the identified parameters are close to zero. Therefore, it is difficult to distinguish the model of order zero and the model of higher order with all parameters close to zero. For this reason, alternatively, a penalty function can be added to the criterion to increase the reliability of detecting order zero. The same approach was applied for the *Test.3*.

The sequential test compares the pdf $p_m(\mathbf{y}(t) \mid \mathcal{D}^{t-1})$ to the pdf $p_0(\mathbf{y}(t) \mid \mathcal{D}^{t-1})$. The first pdf represents a likelihood function that the sequence $\mathbf{y}(t)$ was generated by an AR system of order *m*. The parameters of the unknown AR system are identified recursively within the algorithm. The second pdf, $p_0(\mathbf{y}(t) \mid \mathcal{D}^{t-1})$, represents a likelihood function that the given sequence was generated by an AR model of order 0. A sequence generated by an AR system (with non-zero parameters) of order *m* is colored, while the sequence generated by an AR system of order 0 is white.

The algorithm can be summarized as follows

Algorithm

1) **Initialization**

Consider a model of order m. Calculate the probability density function of the current output $p_m(\mathbf{y}(t) \mid \mathcal{D}^{t-1})$, conditioned by the data $\mathcal{D}^{t-1}$ up to the time $t-1$. For a system with a single output, use the following approach

Calculation of the probability distribution function $p_m(y(t) \mid \mathcal{D}^{t-1}, \hat{\theta})$

a) Start with the initial estimate of the AR model parameters $\hat{\boldsymbol{\theta}}(0) = \hat{\boldsymbol{\theta}}_0 \in \mathbb{R}^m$ and its covariance matrix $\mathbf{P}(0) = \mathbf{P}_0 \in \mathbb{R}^{m \times m}$, e.g. $\hat{\boldsymbol{\theta}}(0) = \mathbf{0}$ and $\mathbf{P}(0) = \gamma \mathbf{I}, \gamma \gg 1$.

b) Update the estimated parameter vector and its covariance matrix using the regressor $\mathbf{z}(t) = \left[y(t-1), ..., y(t-m)\right]^T$ and a new data sample $y(t)$

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \frac{\mathbf{P}(t-1)\mathbf{z}(t)}{1+\zeta(t)}\Big(y(t) - \mathbf{z}^T(t)\hat{\boldsymbol{\theta}}(t-1)\Big),$$
$$\mathbf{P}(t) = \mathbf{P}(t-1) - \frac{\mathbf{P}(t-1)\mathbf{z}(t)\mathbf{z}^T(t)\mathbf{P}(t-1)}{1+\zeta(t)},$$

where $\zeta(t) = \mathbf{z}^T(t)\mathbf{P}(t-1)\mathbf{z}(t)$. Further, the residual variance ($s^2(0) = 0$) is recursively calculated by

$$(t-m)s^2(t) = (t-m-1)s^2(t-1) + \frac{\Big(y(t) - \mathbf{z}^T(t)\hat{\boldsymbol{\theta}}(t-1)\Big)^2}{1+\zeta(t)}.$$

c) Use $\mathbf{z}(t)$, $\hat{\theta}(t)$ and $\mathbf{P}(t)$ to calculate estimate $\hat{y}(t) = \mathbf{z}^T(t)\hat{\boldsymbol{\theta}}(t-1)$ and covariance matrix $P_y = s^2(t-1)\Big(\mathbf{z}^T(t)\mathbf{P}(t-1)\mathbf{z}(t) + 1\Big)$. The probability density function of the output of the AR model is defined as $p(y_m(t) \mid \mathcal{D}^{t-1}, \hat{\theta}) = \mathrm{N}\Big(\hat{y}(t), P_y\Big)$.

End of cpdf calculation $p_m(y(t) \mid \mathcal{D}^{t-1})$

2) **Calculate cpdf for a model of order 0**

Consider a model of the order 0. The probability density function of the current output is defined as

$$p_0(\mathbf{y}(t) \mid \mathcal{D}^{t-1}, \mathbf{S}_0^2) = \mathcal{N}\left(0, \mathbf{S}_0^2(t-1)\right), \tag{3.26}$$

where $\mathbf{S}_0^2(t) = \frac{1}{t}\sum_{i=0}^{t} \mathbf{y}(i)\mathbf{y}^T(i)$.

3) **Compare the models of order *m* and 0**

The criterion of the sequential test using a logarithm of the ratio of the joint probability densities p_m, p_0 can be defined as

$$\mathrm{T}(t) = \mathrm{T}(t-1) + \log\left(\frac{p_0(\mathbf{y}(t) \mid \mathcal{D}^{t-1}, \mathrm{S}_0^2)}{p_m(\mathbf{y}(t) \mid \mathcal{D}^{t-1}, \hat{\theta})}\right). \tag{3.27}$$

This is a recursive modification of the original sequential test proposed by Wald (1945) that uses a ratio of the joint probability densities in the form

$$\mathrm{T}_M(t) = \log\frac{p_0\left(\mathbf{y}(0), \mathbf{y}(1), \ldots, \mathbf{y}(t)\right)}{p_m\left(\mathbf{y}(0), \mathbf{y}(1), \ldots, \mathbf{y}(t)\right)} \tag{3.28}$$

which is a form of the General Likelihood Test.

4) **Calculate the probability thresholds and compare the criterion value to them**

Further, constants a and b are to be chosen such that they represent the desired probability of the first kind and second kind error. The decision about optimality is now as follows

$$\mathrm{T}(t) > \log\frac{1-b}{a} \text{ – the AR model is of the order 0,} \tag{3.29}$$

$$\mathrm{T}(t) < \log\frac{b}{1-a} \text{ – the AR model is of the order } m \tag{3.30}$$

$$\log\frac{1-b}{a} < \mathrm{T}(t) < \log\frac{b}{1-a} \text{ – not enough data for a decision} \tag{3.31}$$

5) **Decide the optimality**

If the criterion value $\mathrm{T}(t)$ does not satisfy the inequality (3.29) or (3.30), that indicates that there is not enough data for a decision respecting the desired probabilities a and b. The lower the probabilities are, the more data is necessary for the decision. If the order zero is confirmed, that indicates that the given sequence is white and the Kalman filter is set optimally. If the higher order is confirmed, the tested sequence is not white. The criterion value $\mathrm{T}(t)$ can be set to zero and the algorithm is restarted for the new decision process. Alternativelly, the Kalman filter can be re–tuned, if non–optimality is detected.

6) Continue with step 2.

End of algorithm

3.4 Numerical simulations

Example 1

In this section, a linear system with a Kalman filter will be considered for numerical simulations. The sequence of innovations will be used as input data for the optimality tests. Consider a stable linear stochastic system of order *n*, $\mathbf{x}(t) \in \mathbb{R}^n$ with *p* stochastic inputs, $\mathbf{v}(t) \in \mathbb{R}^p$ an *r* outputs, $\mathbf{y}(t) \in \mathbb{R}^r$ given by (2.1).

For the first simulation, consider a MISO system having slow, fast and negative poles given by

$$\mathbf{A} = \begin{bmatrix} 0.5 & 1 & 0.2 \\ 0 & -0.5 & 0.2 \\ 0 & 0 & 0.97 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \mathbf{c}^T = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}. \tag{3.32}$$

The covariances used for the input data generator and the tuning matrices are

$$\mathbf{Q}_s = \begin{bmatrix} 10 & -0.5 \\ -0.5 & 1 \end{bmatrix}, R_s = 1; \ \mathbf{Q}_1 = \begin{bmatrix} 6 & 0 \\ 0 & 4 \end{bmatrix}. \tag{3.33}$$

The simulation results are summarized in Tab. 1. The noise generator covariances $\mathbf{Q}_s, R_s$ remain the same and the actual KF setting $\mathbf{Q}, R$ is written in the first column of the table. The first block of Tab. 1 shows the simulation results with optimally set KF. Other blocks show results with two non–optimal settings. High reliability of both tests can be seen with relatively small sets of data.

Tab. 1 – Simulation results for the optimality tests. The values in the table indicate % of correct decisions within repeated simulations. The numbers in the *Test.4* column are % of correct and wrong decisions. The difference between them is % of undecided tests. At *Test.4* simulation, *N* means maximal number of available data and parameter $a = 0.05$, $b = 0.01$.

	N	Test.2 %	Test.3 %	Test.4 %
$\mathbf{Q} = \mathbf{Q}_s$ $R = R_s$	200	94	94	39 ; 1
	500	96	100	97 ; 0
	1000	96	99	100 ; 0
$\mathbf{Q} = 4\mathbf{I}$ $R = R_s$	200	33	55	8 ; 27
	500	53	85	47 ; 33
	1000	86	98	51 ; 44
$\mathbf{Q} = \mathbf{Q}_1$ $R = 2$	200	21	36	94 ; 5
	500	37	68	98 ; 2
	1000	64	94	95 ; 5

It can be seen, that *Test.4* returns many wrong decisions in the situation, when the output prediction error sequence is only slightly colored. The performance of this test can be influenced by tuning the parameters *a*, *b*. If the parameter *a* is lowered, the number of wrong decisions will decrease, but the number of undecided tests increases, because more data is needed for a more precise test.

The following histograms, Figure 1 and Figure 2, show what number of the samples the sequential algorithm needs to decide about the optimality. For both example, the system (3.32) was used. The first example, Figure 1, considered an optimally set KF. It can be seen, that most of the times the algorithm decided within 150 and 600 samples and the number of wrong decisions is very low. For the second example, Figure 2, the Kalman Filter was tuned

using the covariance matrices $\mathbf{Q} = \mathbf{Q}_1, R = 2$, which is not optimal. It can be seen, that the majority of the tests required 110 – 150 samples to correctly decide about the optimality. The number of wrong decisions is 4.8% leaving 0% undecided.

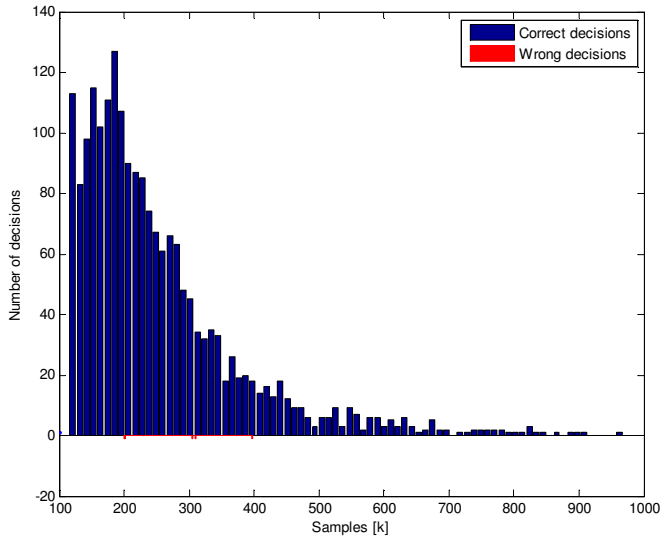

Figure 1 – The sequential test was repeated 2,000 times processing the output prediction error of the system (3.34) that was set optimally. The histogram (axis *y*) shows the number of the tests that decided the optimality after *k* samples (axis *x*). Blue – correct decisions, red – wrong decision. Undecided 8%.

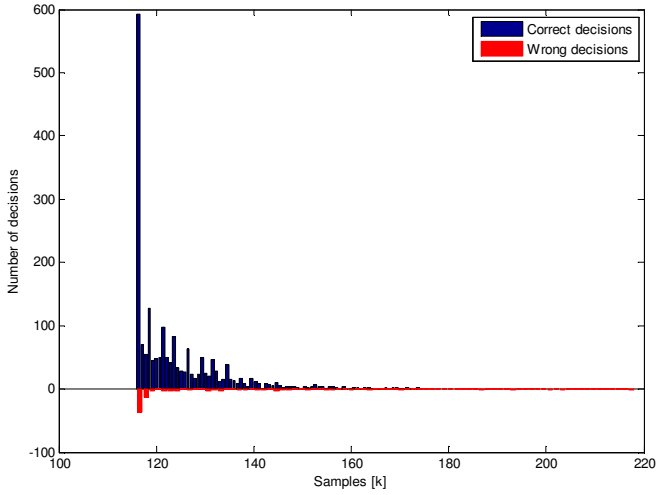

Figure 2 – The sequential test was repeated 2,000 times processing the output prediction error of the system (3.35) that was set non-optimally, the tuning matrices for the KF were $\mathbf{Q} = \mathbf{Q}_1, R = 2$, (3.33) The histogram (axis *y*) shows the number of the tests that decided the optimality after *k* samples (axis *x*). Blue – correct decisions, red – wrong decision. Undecided 0%.

Example 2

Next, the sequential test will be used together with the algorithm for the noise covariance estimation described in Section 4.4, (Matisko and Havlena, 2013). They will work together as an adaptive filter. The process of adaptation of the Kalman gain will be shown in the figure. The example will use a system given by

$$\mathbf{A} = \begin{bmatrix} 0.85 & 1 & 0 & 0 \\ 0 & -0.6 & 1 & 0 \\ -0.2 & 0 & 0.8 & -1 \\ 0 & 0 & 0 & 0.9 \end{bmatrix}, \mathbf{G} = \mathbf{I}_4, \mathbf{c}^T = \begin{bmatrix} 1 & 0.8 & -1 & 0 \end{bmatrix}$$

and the true covariances $\mathbf{Q}_s = \mathrm{diag}(10,3,4,2)$, $R = 1$. The parameters of the sequential tests are $a = 0.01$, $b = 0.001$, the forgetting factor $\varphi = 0.98$ and the order of AR model is $n = 7$ that is higher than the system order n. (Maximal order of the innovation sequence is $2n$, because the whole system consists of the process and the estimator that are both of order n. However, practically it is sufficient to estimate the AR model of order lower than $2n$.) Alternatively, several AR models with different orders can be identified simultaneously and the most probable model is chosen for the sequential criterion.

To evaluate the quality of the KF settings, the Frobenius norm $\left\|\mathbf{K} - \mathbf{K}_{opt}\right\|_F$ will be considered, where $\mathbf{K}$ is a Kalman gain calculated from the identified covariances $\boldsymbol{Q}$, R and $\mathbf{K}_{opt}$ is the optimal Kalman gain. Both gains are time varying. The optimal Kalman gain is actually not known; therefore the method of validation can only be used with simulation data. The quality of the current setting can be, alternatively, back–tested using historical data. If there is enough computational time, several parallel *Q*, *R*–estimating algorithms can be run. Then the optimality tests can validate the quality of the estimates. The process of adaptation in the example can be seen in Figure 3. The prior setting was chosen to be far from the optimum $\mathbf{Q} = 0.01\mathbf{I}, R = 100$.

It can be seen from the resulting tables that the chosen tests have a high level of reliability, even for relatively small sets of data. Also, a very good consistency of *Test.3* can be highlighted.

The sequential test, together with the covariance estimation algorithm was used as an adaptive filter with satisfactory results. This approach can be also used for time varying noise. The change of the noise covariance is indicated by the optimality test and further the adaptation is done by using the estimated noise covariances. The estimated covariances can be back–tested before they are used. The values of the criteria can be alternatively used as a performance measure.

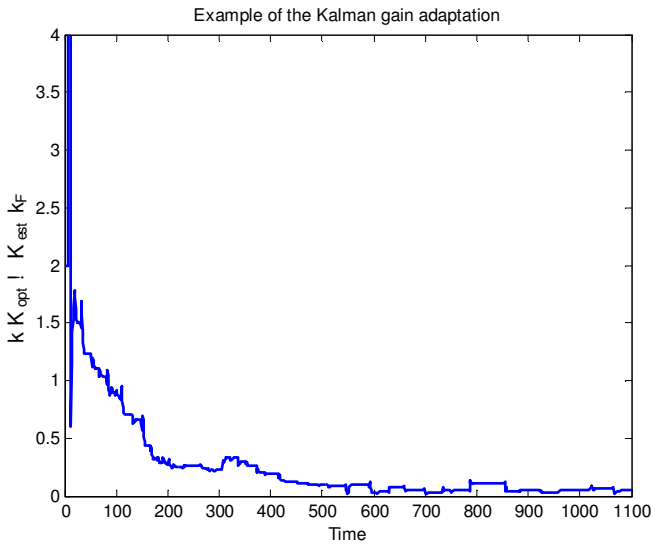

Figure 3 – An example of the adaptation process. The difference between the optimal and the estimated Kalman gain is decreasing.

Example 2

The following simulation considers a larger MIMO system of order six obtained by discretizing a continuous–time system. It has fast, slow and also complex modes to cover a large family of systems. It has three stochastic inputs and three outputs and is given by matrices

$$\mathbf{A} = \begin{bmatrix} 0.98 & 0 & 0.09 & 0 & 0.02 & 0.07 \\ 0 & 0.82 & 0 & 0 & 0 & 0 \\ -0.19 & 0 & 0.9 & 0.04 & 0.43 & -0.09 \\ 0 & 0 & 0 & 0.68 & 0.65 & -0.07 \\ 0 & 0 & 0 & -0.14 & 0.68 & 0.01 \\ 0.08 & 0 & 0 & 0 & 0 & 0.61 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}^T,$$
$$\mathbf{G} = \begin{bmatrix} 1.03 & 0 & -0.14 & 0.04 & -0.01 & 0.83 \\ 0.05 & 9.06 & 0.95 & 0.8 & -0.16 & 0 \\ 0.05 & 0.91 & 0.05 & -0.04 & 0 & 0.79 \end{bmatrix}^T. \tag{3.36}$$

The true noise covariances and the matrices for the KF setting are

$$\mathbf{Q}_s = \begin{bmatrix} 5 & 0.2 & 0 \\ 0.2 & 10 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}, \mathbf{R}_s = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 6 & -0.1 \\ 0 & -0.1 & 0.1 \end{bmatrix}$$

The results of the simulations are listed in Tab. 2. We have not considered Test.4, because its implementation is significantly more difficult for a MIMO system than for a SISO system. The parameter identification of a MIMO AR system requires significantly more data than a SISO system. This examples demonstrates that the optimality tests can be used even for larger MIMO systems.

Tab. 2 – Simulation results for the optimality tests. The values in the table indicate % of correct decisions. First block represents optimal setting, the others non–optimal.

	N	Test.2 %	Test.3 %
$\mathbf{Q} = \mathbf{Q}_s$ $R = R_s$	200	88	92
	500	87	96
	1000	86	97
$\mathbf{Q} = 2\mathbf{I}$ $\mathbf{R} = \mathbf{R}_s$	200	25	94
	500	38	100
	1000	42	100
$\mathbf{Q} = \mathbf{Q}_s$ $\mathbf{R} = \mathbf{I}$	200	12	99
	500	21	100
	1000	30	100

4. Estimation of the noise covariances

4.1 Introduction to the noise covariance estimation

In early 70's, the pioneering papers of Mehra (1970, 1974) on estimation of covariance matrices were published. Several principles were described, including the maximum likelihood approach, correlation methods and covariance matching techniques. In the first half of the 70's methods of Carew and Belanger (1973) and Belanger (1974) were published. The former paper deals with a method for direct estimation of a Kalman gain. The Belanger's paper describes a method for time varying systems. It uses the observation that the autocorrelation of innovations is linearly dependent on the noise covariances. For the following thirty years, this topic was quite overlooked.

It was observed that the autocorrelation of innovation sequence is linearly dependent on the noise covariances, (Belanger, 1974). This property was used Odelson *et al*. (2005) who offered a new method for the covariances estimation called ALS (Autocorrelation least squares). Further modifications of this method can be found in Akesson *et al*. (2008) and Rajamani and Rawlings (2009). The mentioned methods were tested on various systems in Matisko (2009). ALS is a point estimate method and it does not provide any additional information about the solution accuracy. The objective is to find some estimation quality measure, such as the probability distribution function of the estimates.

4.2 Noise covariance estimation using Autoregressive Least Squares method

In this chapter, the overview of th ALS method will be given, because it will be used in the numerical examples and comparisons in the later sections. More extensive testing of the esti-

mating algorithms was done in Matisko (2009). For other discussions and comparisons, see Duník *et al*. (2009).

Consider a linear stochastic system given by

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{G}\mathbf{v}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{e}(t). \end{aligned} \tag{4.1}$$

A state prediction error is defined as $\tilde{\mathbf{x}}(t \mid t-1) \triangleq \mathbf{x}(t) - \hat{\mathbf{x}}(t \mid t-1)$. It is possible to write a trajectory of $\tilde{\mathbf{x}}(t \mid t-1)$ using a state space model. The output of the system is the L–innovation sequence. 'L' emphasizes that the KF is not tuned optimally because the true noise covariances are unknown. The system is of the form

$$\tilde{\mathbf{x}}(t+1 \mid t) = \underbrace{(\mathbf{A} - \mathbf{L}\mathbf{C})}_{\bar{\mathbf{A}}} \tilde{\mathbf{x}}(t \mid t-1) + \underbrace{[\mathbf{G} \quad -\mathbf{L}]}_{\bar{\mathbf{G}}} \underbrace{\begin{bmatrix} \mathbf{v}(t) \\ \mathbf{e}(t) \end{bmatrix}}_{\bar{\mathbf{w}}(k)}, \tag{4.2}$$

$$\begin{aligned} \tilde{\mathbf{x}}(t+1) &= \bar{\mathbf{A}}\tilde{\mathbf{x}}(t) + \bar{\mathbf{G}}\bar{\mathbf{w}}(k), \\ \tilde{\mathbf{y}}(t) &= \mathbf{C}\tilde{\mathbf{x}}(t) + \mathbf{e}(t), \end{aligned} \tag{4.3}$$

where the covariance matrices of the stochastic inputs are

$$\begin{aligned} \mathcal{E}\left\{\bar{\mathbf{w}}(t)\bar{\mathbf{w}}^T(t)\right\} &= \bar{\mathbf{Q}}_w = \begin{bmatrix} \mathbf{Q} & 0 \\ 0 & \mathbf{R} \end{bmatrix}, \\ \mathcal{E}\left\{\bar{\mathbf{w}}(t)\mathbf{v}^T(t)\right\} &= \begin{bmatrix} 0 \\ \mathbf{R} \end{bmatrix}. \end{aligned} \tag{4.4}$$

The steady–state Kalman gain $\mathbf{L} \in \mathbb{R}^{n \times r}$ is calculated using

$$\mathbf{L} = \mathbf{A}\mathbf{P}\mathbf{C}^T \left(\mathbf{C}\mathbf{P}\mathbf{C}^T + \mathbf{R}_0\right)^{-1}, \tag{4.5}$$

where $\mathbf{P}$ is the solution of the algebraic Riccati equation using the prior estimates of noise covariances $\mathbf{Q}_0, \mathbf{R}_0$. It is assumed that the pair $(\mathbf{A}, \mathbf{C})$ is detectable and matrix $\bar{\mathbf{A}} = \mathbf{A} - \mathbf{L}\mathbf{C}$ is stable, so the effect of the initial conditions can be neglected, i.e. $E\{\boldsymbol{\varepsilon}(0)\} \to 0$ and $\operatorname{cov}\{\tilde{\mathbf{x}}(0)\} \to \mathbf{P}_x$. The steady–state covariance of the state prediction error can be calculated using the Lyapunov equation of the form

$$\mathbf{P}_{\tilde{x}} = \bar{\mathbf{A}}\mathbf{P}_{\tilde{x}}\bar{\mathbf{A}}^T + \bar{\mathbf{G}}\bar{\mathbf{Q}}_w\bar{\mathbf{G}}^T \tag{4.6}$$

which can be solved by using vectorization operator

$$\begin{aligned} \mathbf{P}_x &= \bar{\mathbf{A}}\mathbf{P}_x\bar{\mathbf{A}}^T + \bar{\mathbf{G}}\bar{\mathbf{Q}}_w\bar{\mathbf{G}}^T, \\ (\mathbf{P}_x)_{\text{vec}} &= (\bar{\mathbf{A}} \otimes \bar{\mathbf{A}})(\mathbf{P}_x)_{\text{vec}} + (\bar{\mathbf{G}}\bar{\mathbf{Q}}_w\bar{\mathbf{G}}^T)_{\text{vec}}, \\ (\mathbf{P}_x)_{\text{vec}} &= (\mathbf{I}_{n^2} - \bar{\mathbf{A}} \otimes \bar{\mathbf{A}})^{-1}(\bar{\mathbf{G}} \otimes \bar{\mathbf{G}})(\bar{\mathbf{Q}}_w)_{\text{vec}}. \end{aligned} \tag{4.7}$$

The autocorrelation function of the L–innovation sequence can be calculated by

$$\begin{aligned} \tilde{\mathbf{Y}}_0 &\equiv \{\tilde{\mathbf{y}}(t)\tilde{\mathbf{y}}^T(t)\} = \mathbf{C}\mathbf{P}_x\mathbf{C}^T + \mathbf{R}, \\ \tilde{\mathbf{Y}}_k &\equiv \mathcal{E}\{\tilde{\mathbf{y}}(t+k)\tilde{\mathbf{y}}^T(t)\} = \mathbf{C}\bar{\mathbf{A}}^k\mathbf{P}_x\mathbf{C}^T - \mathbf{C}\bar{\mathbf{A}}^{k-1}\mathbf{L}\mathbf{R}, \ 1 \le k \le M. \end{aligned} \tag{4.8}$$

The user–defined parameter M is the maximum number of used lags. The autocorrelation matrix is defined as

$$\mathcal{R}(M) \triangleq \begin{bmatrix} \tilde{\mathbf{Y}}_0 & \cdots & \tilde{\mathbf{Y}}_{M-1} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{Y}}_{M-1}^T & \cdots & \tilde{\mathbf{Y}}_0 \end{bmatrix}, \tag{4.9}$$

If $\mathbf{L}$ in (4.8) is optimal, the off–diagonal members of the autocorrelation matrix are zero. However, the optimal gain is not known and is to be found, therefore all members of the autocorrelation matrix have to be considered for its calculation. Further, we define several matrices to express the autocorrelation matrix (4.9)

$$\mathcal{O} \triangleq \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\bar{\mathbf{A}} \\ \vdots \\ \mathbf{C}\bar{\mathbf{A}}^{M-1} \end{bmatrix}, \quad \Xi = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \mathbf{C} & 0 & 0 & 0 \\ \vdots & \ddots & & \vdots \\ \mathbf{C}\bar{\mathbf{A}}^{M-2} & \cdots & \mathbf{C} & 0 \end{bmatrix}, \quad \Psi = \Xi\left(\bigoplus_{j=1}^{M}(-\mathbf{L})\right), \tag{4.10}$$

where $\mathcal{O}$ is an observability matrix. Using matrices (4.10) and the definitions (4.4), the autocorrelation matrix (4.9) can be expressed as

$$\mathcal{R}(M) \triangleq \mathcal{O}\mathbf{P}_x\mathcal{O}^T + \Xi\left(\bigoplus_{i=1}^{M}\bar{\mathbf{G}}\bar{\mathbf{Q}}_w\bar{\mathbf{G}}^T\right)\Xi^T + \Psi\left(\bigoplus_{i=1}^{M}\mathbf{R}\right) + \left(\bigoplus_{i=1}^{M}\mathbf{R}\right)\Psi^T + \bigoplus_{i=1}^{M}\mathbf{R}. \tag{4.11}$$

From the above definition it can be seen, that the autocorrelation matrix is linearly dependent on the noise covariance matrices **Q**, **R**. To find the unknown covariance matrices, the sample autocorrelation function of the innovation will be used as a data to be fitted by (4.11). The sample autocorrelation matrix is defined as

$$\hat{\mathcal{R}}(M) \triangleq \begin{bmatrix} \hat{\mathbf{Y}}_0^{N-k} & \cdots & \hat{\mathbf{Y}}_{M-1}^{N-k} \\ \vdots & \ddots & \vdots \\ \hat{\mathbf{Y}}_{M-1}^{N-k} & \cdots & \hat{\mathbf{Y}}_0^{N-k} \end{bmatrix}, \tag{4.12}$$

where the elements of matrix (4.12) are calculated using measured outputs by

$$\hat{\mathbf{Y}}_k^{N-k} \triangleq \frac{1}{N-k} \sum_{i=1}^{N-k} \tilde{\mathbf{y}}(i) \tilde{\mathbf{y}}^T(i+k). \tag{4.13}$$

The goal is to solve the minimization problem of the form

$$\begin{bmatrix} \left(\hat{\mathbf{Q}}\right)_{\text{vec}} \\ \left(\hat{\mathbf{R}}\right)_{\text{vec}} \end{bmatrix} = \arg\min_{\mathbf{Q},\mathbf{R}} \left\| \mathcal{A} \begin{bmatrix} \left(\mathbf{Q}\right)_{\text{vec}} \\ \left(\mathbf{R}\right)_{\text{vec}} \end{bmatrix} - \left(\hat{\mathcal{R}}(M)\right)_{\text{vec}} \right\|_2^2, \tag{4.14}$$

where matrix $\mathcal{A}$ can be expressed by the following formulas using the solution of the Lyapunov equation (4.7)

$$\begin{aligned} \mathcal{B} &= (\mathcal{C} \otimes \mathcal{C}) \left(\mathbf{I}_{n^2} - \bar{\mathbf{A}} \otimes \bar{\mathbf{A}}\right)^{-1} + (\Xi \otimes \Xi) \mathcal{P}_{n,M}, \\ \mathcal{A} &= \begin{bmatrix} \mathcal{B}(\mathbf{G} \otimes \mathbf{G}) & \mathcal{B}(\mathbf{L} \otimes \mathbf{L}) + \left(\Psi \oplus \Psi + \mathbf{I}_{r^2 M^2}\right) \mathcal{P}_{r,M} \end{bmatrix}, \end{aligned} \tag{4.15}$$

The covariances obtained by solving (4.14) are symmetric due to the structure of the least–squares problem, (Odelson *et al.*, 2006).

The permutation matrix[3] $\mathcal{P}_{n,N}$ consists of zeros and ones and has the following properties

$$\text{a)} \quad \text{vec}\left\{ \bigoplus_{i=1}^{N} \mathbf{X} \right\} = \mathcal{P}_{j,N} \, \text{vec}\{\mathbf{X}\}$$

for matrix $\mathbf{X} \in \mathbb{R}^{j \times j}$, or

[3] The Matlab function for $\mathcal{P}_{n,N}$ calculation can be found in Appendix A.

$$b)\ \ \mathrm{vec}\left\{\bigoplus_{i=1}^{N}\mathbf{X}\right\}=\boldsymbol{\mathcal{P}}_{j,l,N}\ \mathrm{vec}\{\mathbf{X}\} \tag{4.16}$$

for matrix $\mathbf{X}\in\mathbb{R}^{j\times l}$. The size of the permutation matrix is a) $(jN)^2\times j^2$ and b) $ljN^2\times lj$. Further information about deriving the above formulas can be found in the original paper of Odelson *et al*. (2005). A newer paper, (Rajamani and Rawlings, 2009), offers a modification for this approach using operator vecMin instead of vec, (briefly described in subsection 4.3). That leads to less equations to be solved in criterion (4.14) and therefore significantly less memory consumption. Another modification is given by Akesson *et al*. (2008) that considers cross–correlation between the process and measurement noise. The mentioned papers also discuss the problem of positive definiteness of the obtained matrices and offer a method to recover the number of stochastic inputs.

Some comments need to be added on this approach. It is obvious that the input data for the algorithm are of the form of the second statistical moment. The main problem of using the estimated autocorrelation function is its slow convergence that is of order 1/*N*, (Odelson *et al*. 2005). That means that each member of matrix (4.12) is blurred by noise. This complicates the estimation of the noise covariances. Theoretically, it is true that the approach can use arbitrary many equations. However, the information of the autocorrelation function decreases with increasing lag. As the autocorrelation function is calculated for higher lag, the newly obtained information decreases. Therefore, adding new equations (using higher lags) without increasing the amount of data does not significantly improve the results.

Another problematic part of the approach is memory consumption. Matrices (4.10) and (4.15) are large even for small systems and the „out of memory“ exception can appear during computation. This problem is mostly solved by Rajamani and Rawlings (2009) that uses only one column of matrix (4.12). This is possible due to the fact, that the other columns of the matrix (4.12) contain the same information as the first column.

4.3 A single column Autoregressive least squares method (scALS)

A very brief summary of the single column ALS (scALS) method will be given in this sub–section. For complete derivation, see the original paper by Rajamani and Rawlings (2009).

The full algorithm can be downloaded from the web site under GPL license. The scALS method uses the following matrices, analogously to the ALS method

$$\mathcal{O} \triangleq \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\bar{\mathbf{A}} \\ \vdots \\ \mathbf{C}\bar{\mathbf{A}}^{M-1} \end{bmatrix}, \quad \Xi = \begin{bmatrix} \mathbf{I}_r \\ -\mathbf{C}\bar{\mathbf{A}}\mathbf{L} \\ \vdots \\ -\mathbf{C}\bar{\mathbf{A}}^{M-2}\mathbf{L} \end{bmatrix}, \tag{4.17}$$

where the matrix Ξ is different from (4.10) in the previous section. The single column sample autocorrelation function is given by

$$\hat{\mathcal{R}}_{sc}(M) \triangleq \frac{1}{N-M+1} \begin{bmatrix} \boldsymbol{\varepsilon}(1) & \boldsymbol{\varepsilon}(2) & \cdots & \boldsymbol{\varepsilon}(N-M+1) \\ \boldsymbol{\varepsilon}(2) & \boldsymbol{\varepsilon}(3) & & \boldsymbol{\varepsilon}(N-M+2) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\varepsilon}(M) & \boldsymbol{\varepsilon}(M+1) & \cdots & \boldsymbol{\varepsilon}(N) \end{bmatrix} \begin{bmatrix} \boldsymbol{\varepsilon}(1)^T \\ \boldsymbol{\varepsilon}(2)^T \\ \vdots \\ \boldsymbol{\varepsilon}(N-M+1)^T \end{bmatrix}, \tag{4.18}$$

where N is the data length and M is the maximum lag of the autocorrelation function. Further, the criterion is using the $\mathrm{vecMin}\{\}$ operator. Matrix $\mathcal{A}$ is given by

$$\begin{aligned} \mathcal{A} &= \begin{bmatrix} \mathcal{A}_1 & \mathcal{A}_2 \end{bmatrix}, \\ \mathcal{A}_1 &= \left(\mathbf{C} \otimes \mathcal{O}\right)\left(\mathbf{I}_{n^2} - \bar{\mathbf{A}} \otimes \bar{\mathbf{A}}\right)^{-1} \mathcal{D}_n, \\ \mathcal{A}_2 &= \left(\left(\mathbf{C} \otimes \mathcal{O}\right)\left(\mathbf{I}_{n^2} - \bar{\mathbf{A}} \otimes \bar{\mathbf{A}}\right)^{-1}\left(\mathbf{L} \otimes \mathbf{L}\right) + \left(\mathbf{I}_r \otimes \Xi\right)\right) \mathcal{D}_r, \end{aligned} \tag{4.19}$$

where $\mathcal{D}_n \in \mathbb{R}^{n^2 \times n(n+1)/2}$ is a duplication matrix[4] (Magnus and Neudecker, 1999) containing ones and zeros such that for a symmetric matrix **X,** it holds

$$\mathrm{vec}\{\mathbf{X}\} = \mathcal{D}_n \, \mathrm{vecMin}\{\mathbf{X}\}. \tag{4.20}$$

Now, matrices (4.18) and (4.19) are put into the criterion (4.14). The obtained matrices are in the minimal–vectorized form.

The paper Rajamani and Rawlings (2009) offers an approach to enforce the positive definiteness by inserting a barrier function to the criterion (4.14); ALS+SDP refers to the Semidefinite Programming used with the ALS approach. This approach not only enforces the positive

[4] The Matlab function for $\mathcal{D}_n$ calculation can be found in Appendix A.

definiteness, but also significantly improves the quality of ALS estimates. ALS+SDP returns estimates having lower variance among repeated estimates than the original ALS.

4.4 Bayesian method for the noise covariance estimation[5]

In this section, the Bayesian principles are employed to gain insight into the estimation of the noise covariance matrices. The goal is to express the posterior probability distribution function of the covariance matrices conditioned by measured data using the Bayes formula. The posterior probability distribution accumulates information from data by recursive multiplying of the likelihood function with the prior pdf

$$\begin{aligned} p\left(\mathbf{Q},\mathbf{R} \mid \mathcal{Y}^{t}\right) &= p\left(\mathbf{Q},\mathbf{R} \mid \mathbf{y}(t),\mathcal{Y}^{t-1}\right) = \frac{p\left(\mathbf{Q},\mathbf{R},\mathbf{y}(t) \mid \mathcal{Y}^{t-1}\right)}{p\left(\mathbf{y}(t) \mid \mathcal{Y}^{t-1}\right)} \\ &= \frac{p\left(\mathbf{y}(t) \mid \mathcal{Y}^{t-1},\mathbf{Q},\mathbf{R}\right)}{p\left(\mathbf{y}(t) \mid \mathcal{Y}^{t-1}\right)} p\left(\mathbf{Q},\mathbf{R} \mid \mathcal{Y}^{t-1}\right), \end{aligned} \tag{4.21}$$

where $p\left(\mathbf{y}(t) \mid \mathbf{Q},\mathbf{R},\mathcal{Y}^{t-1}\right)$ represents the likelihood function of $\mathbf{Q}$, $\mathbf{R}$ for the given data calculated by the Kalman filter described in Section 2.2. The prior probability distribution $p\left(\mathbf{Q},\mathbf{R} \mid \mathcal{Y}^{t-1}\right)$ for was calculated in the previous step using measurements up to the time $t-1$. There is no conjugate prior for $\mathbf{Q}$, $\mathbf{R}$, because the likelihood function has a nonlinear dependency on $\mathbf{Q}$, $\mathbf{R}$ due to the Riccati equation. The posterior probability cannot be expressed in a closed form, therefore several numerical implementations will be introduced further. The denumerator in (4.21) is a normalization factor. In the following, it will be shown that it is not necessary to calculate the cpdf $p\left(\mathbf{y}(t) \mid \mathcal{Y}^{t-1}\right)$. Instead of (4.21), the following formula will be used

[5] The text of Sections 4.4 to 4.9 was originally published in the International Journal of Adaptive Control and Signal Processing, reference Matisko and Havlena (2013). The text of the paper is reproduced here with explicit permission from John Wiley & Sons. and any third parties are excluded from this permission. This text might be further use only with additional permission from John Wiley & Sons.

$$p\left(\mathbf{Q},\mathbf{R} \mid \mathcal{Y}^t\right) \propto p\left(\mathbf{y}(t) \mid \mathcal{Y}^{t-1},\mathbf{Q},\mathbf{R}\right) p\left(\mathbf{Q},\mathbf{R} \mid \mathcal{Y}^{t-1}\right), \tag{4.22}$$

where $\propto$ means equality up to a normalization constant. The normalization will be proceeded numerically.

4.4.1 Estimation of the noise covariances on a grid of parameters

Consider a stable SISO system (4.1) of order n with two uncorrelated stochastic inputs $v(t),\ e(t) \in \mathbb{R}$ and a single output, $y(t) \in \mathbb{R}$. The cpdf $p\left(Q,R \mid \mathcal{Y}^t\right)$ will be calculated on a grid of scalar parameters Q, R. The algorithm uses parallel Kalman filters, one for each pair of the noise covariances covered by the grid. For each data sample $y(t)$, the conditional posterior probability distribution of the covariances is calculated using (4.22). The posterior cpdf calculated at time instant t becomes the prior probability at time $t+1$. After the desired amount of data has been processed, the maximum (or mean) of the posterior cpdf $p\left(Q,R \mid \mathcal{Y}^t\right)$ is found. The algorithm is summarized as follows.

Algorithm

---

1) **Initialization**

Create two sets of values $Q > 0$ and $R > 0$ with sizes n_Q, n_R

$$\mathcal{Q} = \left\{Q_1,\ldots,Q_{n_Q}\right\},$$
$$\mathcal{R} = \left\{R_1,\ldots,R_{n_R}\right\}.$$

Create a set $\mathcal{S}$ of pairs $[Q_i,R_j]$ as a Cartesian combination of the sets $\mathcal{S} = \mathcal{Q} \times \mathcal{R}$ having $n_{\mathcal{S}} = n_Q \cdot n_R$ members. The pairs $[Q_i,R_j]$ represent the samples of process and measurement noise covariances used by individual Kalman filters running in parallel. Next, the prior pdf $p\left(Q_i,R_j \mid \mathcal{Y}^{-1}\right)$ is set to some starting distribution. The initial values of the state estimate are

$$\begin{aligned}\hat{\mathbf{x}}_{i,j}\left(0\mid -1\right) &= 0,\\ \mathbf{P}_{i,j}\left(0\mid -1\right) &= \rho\mathbf{I}_{n\times n},\ \rho \gg 1.\end{aligned} \tag{4.23}$$

The initial estimates and $p\left(Q_i, R_j \mid \mathcal{Y}^{-1}\right)$, can be modified based on the available prior information. The measurement accuracy is usually known and it can be considered for forming the set $\mathcal{R}$. Underlying first principles usually provide information about the process noise structure and intervals for its values. Logarithmic scale is reasonable for the sets $\mathcal{Q},\mathcal{R}$ to assure a higher grid density for smaller covariances leading to equal relative approximation errors.

For each $t = 0 \ldots T_{\max}$

2) **Processing the output data** $y(t)$

For each i, j such that $Q_i, R_j \in \mathcal{S}$

3) **Estimation of the state for each of the parallel Kalman filters**

Pick a pair $[Q_i, R_j]$ from the set $\mathcal{S}$. Given $\hat{\mathbf{x}}_{i,j}\left(t-1\mid t-1\right)$ use (2.9) to calculate the state prediction $\hat{\mathbf{x}}_{i,j}\left(t\mid t-1\right)$ and its covariance matrix $\mathbf{P}_{i,j}\left(t\mid t-1\right)$. Further, calculate the output prediction and its covariance using (2.11) for each individual KF. This step is skipped if $t = 0$.

4) **Calculation of the likelihood function of the Q, R for the current output** $y(t)$

Based on the values from step 2), the likelihood function $p\left(y(t)\mid \mathcal{Y}^{t-1}, Q_i, R_j\right)$ is calculated using (2.10).

5) **Calculation of the posterior probability of the pair** $[Q_i, R_j]$

The prior probability of the pair $[Q_i, R_j]$ is updated by the likelihood function using (4.22) resulting to the non–normalized posterior cpdf $p'\left(Q_i, R_j \mid \mathcal{Y}^t\right)$.

6) **Calculation of the posterior state estimate for each Kalman filter**

Use (2.8) with the data $y(t)$ to calculate $\hat{\mathbf{x}}_{i,j}\left(t \mid t\right)$ and $\mathbf{P}_{i,j}\left(t \mid t\right)$. Continue to step 3) until all the members of the set $\mathcal{S}$ have been completed.

Next i, j

---

7) N**ormalization**

Normalization of the posterior cpdf is done by

$$p\left(Q_i, R_j \mid \mathcal{Y}^t\right) = \frac{p'\left(Q_i, R_j \mid \mathcal{Y}^t\right)}{\sum_{R_i, Q_j \in \mathcal{S}} p'\left(Q_i, R_j \mid \mathcal{Y}^t\right)}. \tag{4.24}$$

8) **Increasing of the time index**

Finish, if the desired amount of data $T_{\max}$ has been processed. Otherwise, increase the time index $t := t + 1$ and return to step 2) .

Next t

---

9) **Estimation of the noise covariances**

The covariance estimates are obtained by

$$[\hat{Q}, \hat{R}]_{\mathrm{MAP}} = \arg\max_{R_i, Q_j \in \mathcal{S}} \left(p\left(Q_i, R_j \mid \mathcal{Y}^t\right)\right), \tag{4.25}$$

or alternatively

$$[\hat{Q}, \hat{R}]_{\mathrm{MS}} = \sum_{R_i, Q_j \in \mathcal{S}} [Q_i, R_j] \cdot p\left(Q_i, R_j \mid \mathcal{Y}^t\right). \tag{4.26}$$

End of algorithm

The maximum a posteriori probability (MAP) estimates and the mean square (MS) estimates can differ, especially if the shape of the posterior cpdf is complicated or multimodal. Considering the mean value as the final result might make the algorithm more robust to inaccuracy in cpdf approximation. Both estimates can be considered for the KF tuning. The state estimation quality of Kalman filters with different settings can be compared by optimality tests described in Chapter 3.

Finding the covariance estimates on the grid can be inaccurate especially if the intervals of the sets $\mathcal{Q}$, $\mathcal{R}$ are large. A possible solution is to update the grid around the cpdf maximum and repeat the algorithm with the new grid. Several methods for an adaptive grid are available, (Šimandl, *et al*. 2006). However, this solution might not be satisfactory due to a complicated shape of the posterior cpdf (4.21). Another approach uses *importance sampling,* which generates samples from a distribution given by a set of samples, (Candy, 2009). Importance sampling belongs to the Monte Carlo (MC) family. This approach also allows extension of the estimation to the multidimensional case.

4.4.2 Posterior probability function, numerical examples

In this section, we will shortly demonstrate basic properties of the posterior probability distribution function of Q, R conditioned by data $\mathcal{Y}^t$. Consider a scalar system with $a = 0.5$, $c = 1$, $g = 1$ and the noise characterized by covariances $Q_r = 2$, $R_r = 10$. In Figure 4, the likelihood function of the noise covariances is shown using a single data $y(t)$. The shape of the likelihood function depends on the system matrices, (mainly **A**) and the noise covariances. Updating the likelihood function (4.22) leads to its sharpening around the maximum. After 2,000 output data is used, the posterior pdf has a well detectable maximum, Figure 5. The following graphs demonstrate that the measured output data contains sufficient information about the unknown noise covariances.

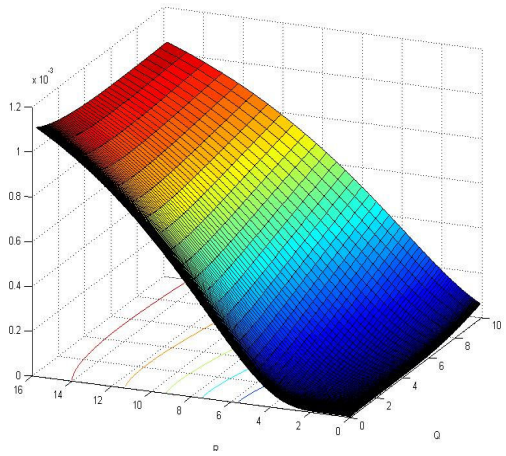

Figure 4 – Likelihood function of the covariances Q, R at a single time step.

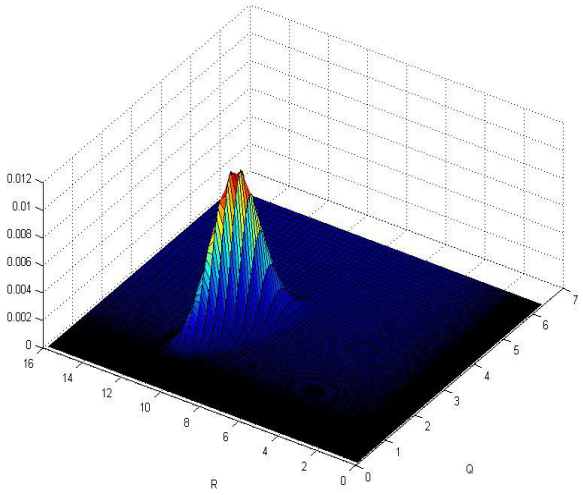

Figure 5 – Posterior pdf of the covariances Q, R using 2,000 output data.

4.5 Monte Carlo approach

Monte Carlo is a group of numerical methods useable for solving various problems, which are hard–to–solve analytically. The previous section described calculation of the conditional posterior probability distribution function of the covariances Q, R on a fixed grid of parameters. For a better localization of the cpdf maximum, it is desirable to have more points in the place of high posterior probability (4.24) and omit those with a low probability. Therefore, after each run of the algorithm in Section 4.4.1 called an iteration (labeled k) the set $\mathcal{S}$ is updated. The new points for the set $\mathcal{S}$ are generated from the posterior cpdf (4.24) employing the importance sampling method. Then, a prior pdf is assigned to the new points in $\mathcal{S}$ and the posterior cpdf (4.24) is calculated again.

The covariance matrices must be symmetric and positive definite, but they can be parameterized in several ways. The appropriate parameterization is selected according to the prior knowledge about the stochastic properties. For example, it can be a multiplier of a unit matrix $\mathbf{Q} \in \left\{\sigma^2 \mathbf{I}_p\right\}$, diagonal elements of the matrix $\mathbf{Q} \in \left\{\mathrm{diag}(\sigma_1^2,\ldots,\sigma_p^2)\right\}$ or elements of the Cholesky factor. The parameters will be labeled as a vector $\boldsymbol{\theta}$; i.e. both noise covariance matrices $\mathbf{Q}(\boldsymbol{\theta})$ and $\mathbf{R}(\boldsymbol{\theta})$ are functions of the parameter vector which is to be found.

The Monte Carlo algorithm can be summarized as follows.

Algorithm

1) **Initialization**

Create a set $\mathcal{S}$ containing n_S samples of parameter vectors $\boldsymbol{\theta}_i^{(0)}$. Iteration $k = 0$. The initial setting of the parallel state estimators is set by (4.23).

For each $k = 0 \ldots K$

For each $t = 0 \ldots T_{\max}$

2) **Processing the output data** $\mathbf{y}(t)$

For each i such that $\boldsymbol{\theta}_i \in \mathcal{S}^{(k)}$

3) **Estimation of the state for each of the parallel Kalman filters**

Pick $\boldsymbol{\theta}_i$ from the set $\mathcal{S}^{(k)}$. Given $\hat{\mathbf{x}}_i\left(t-1 \mid t-1\right)$ use

$$\begin{aligned}\hat{\mathbf{x}}(t+1 \mid t) &= \mathbf{A}\hat{\mathbf{x}}(t \mid t),\\ \mathbf{P}\left(t+1 \mid t\right) &= \mathbf{A}\mathbf{P}\left(t \mid t\right)\mathbf{A}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T\end{aligned}$$

to calculate the state prediction $\hat{\mathbf{x}}_i\left(t \mid t-1\right)$ and its covariance matrix $\mathbf{P}_i\left(t \mid t-1\right)$. Further, calculate the output prediction and its covariance using

$$\begin{aligned}\hat{\mathbf{y}}\left(t+1 \mid t\right) &= \mathbf{C}\hat{\mathbf{x}}\left(t+1 \mid t\right),\\ \mathbf{P}_{yy}\left(t+1 \mid t\right) &= \mathbf{C}\mathbf{P}\left(t+1 \mid t\right)\mathbf{C}^T + \mathbf{R}\end{aligned}$$

for each individual KF. This step is skipped if $t = 0$.

4) **Calculation of the likelihood function of** $\boldsymbol{\theta}_i$ **for the current output** $\mathbf{y}(t)$

Based on the values from step 3), the likelihood function $p\left(\mathbf{y}(t) \mid \mathcal{Y}^{t-1}, \boldsymbol{\theta}_i\right)$ is calculated using

$$p\left(\mathbf{y}(t) \mid \mathcal{Y}^{t-1}, \mathbf{Q}, \mathbf{R}\right) = \int p\left(\mathbf{y}(t) \mid \mathbf{x}(t), \mathbf{Q}, \mathbf{R}\right) p\left(\mathbf{x}(t) \mid \mathcal{Y}^{t-1}, \mathbf{Q}, \mathbf{R}\right) d\mathbf{x}(t) = \\ = \mathcal{N}\left(\hat{\mathbf{y}}\left(t \mid t-1\right); \mathbf{P}_{yy}\left(t \mid t-1\right)\right),$$

5) **Calculation of the posterior probability of** $\boldsymbol{\theta}_i$

The prior probability of the pair $\boldsymbol{\theta}_i$ is updated by the likelihood function using

$$p\left(\mathbf{Q}, \mathbf{R} \mid \mathcal{Y}^t\right) \propto p\left(\mathbf{y}(t) \mid \mathcal{Y}^{t-1}, \mathbf{Q}, \mathbf{R}\right) p\left(\mathbf{Q}, \mathbf{R} \mid \mathcal{Y}^{t-1}\right),$$

resulting to the non–normalized posterior cpdf $p'\left(\boldsymbol{\theta}_i \mid \mathcal{Y}^t\right)$.

6) **Calculation of the posterior state estimate for each Kalman filter**

Use the Kalman filter equations

$$\hat{\mathbf{x}}(t \mid t) = \hat{\mathbf{x}}(t \mid t-1) + \mathbf{P}\left(t \mid t-1\right)\mathbf{C}^T\left(\mathbf{C}\mathbf{P}\left(t \mid t-1\right)\mathbf{C}^T + \mathbf{R}\right)^{-1}\boldsymbol{\varepsilon}\left(t \mid t-1\right),$$
$$\mathbf{P}\left(t \mid t\right) = \mathbf{P}\left(t \mid t-1\right) - \mathbf{P}\left(t \mid t-1\right)\mathbf{C}^T\left(\mathbf{C}\mathbf{P}\left(t \mid t-1\right)\mathbf{C}^T + \mathbf{R}\right)^{-1}\mathbf{C}\mathbf{P}\left(t \mid t-1\right),$$

with the data $\mathbf{y}(t)$ to calculate $\hat{\mathbf{x}}_i\left(t \mid t\right)$ and $\mathbf{P}_i\left(t \mid t\right)$. Continue to step 3) until all the members of the set $\mathcal{S}^{(k)}$ have been completed.

Next *i*

---

7) N**ormalization**

Normalization of the posterior cpdf is done by

$$p\left(\boldsymbol{\theta}_i \mid \mathcal{Y}^t\right) = \frac{p'\left(\boldsymbol{\theta}_i \mid \mathcal{Y}^t\right)}{\sum_{\boldsymbol{\theta}_i \in \mathcal{S}} p'\left(\boldsymbol{\theta}_i \mid \mathcal{Y}^t\right)}. \tag{4.27}$$

8) **Increasing of the time index**

Finish, if the desired amount of data $T_{\max}$ has been processed. Otherwise, increase the time index $t := t + 1$ and return to step 2) .

Next t

9) **Estimation of the noise covariance matrices**

Find the parameter estimates using

$$\hat{\boldsymbol{\theta}}_{MAP}^{(k)} = \arg\max_{\boldsymbol{\theta}_i \in \mathcal{S}} \left(p\left(\boldsymbol{\theta}_i^{(k)} \mid \mathcal{Y}^{T_{\max}} \right) \right), \tag{4.28}$$

or alternatively

$$\hat{\boldsymbol{\theta}}_{MS}^{(k)} = \sum_{\boldsymbol{\theta}_i \in \mathcal{S}} \boldsymbol{\theta}_i^{(k)} p\left(\boldsymbol{\theta}_i^{(k)} \mid \mathcal{Y}^{T_{\max}} \right). \tag{4.29}$$

Calculate the noise covariance matrices from the estimated parameters $\mathbf{Q}\left(\hat{\boldsymbol{\theta}}_{MS}^{(k)}\right)$, $\mathbf{R}\left(\hat{\boldsymbol{\theta}}_{MAP}^{(k)}\right)$.

10) **Generating an initial set of the parameter vectors θ for the next iteration**

Generate n_S new parameter vectors $\boldsymbol{\theta}_i^{(k+1)}$ from the distribution $p\left(\boldsymbol{\theta}^{(k)} \mid \mathcal{Y}^{T_{\max}} \right)$ using the importance sampling method forming a new set $\mathcal{S}^{(k+1)}$ for the next iteration

$$\boldsymbol{\theta}_i^{(k+1)} \sim p\left(\boldsymbol{\theta}^{(k)} \mid \mathcal{Y}^{T_{\max}} \right). \tag{4.30}$$

Perturb the new samples $\boldsymbol{\theta}^{(k+1)}$ with a noise $\Delta\boldsymbol{\theta}$ to prevent degenerating of the cpdf

$$\boldsymbol{\theta}_i^{(k+1)} := \boldsymbol{\theta}_i^{(k+1)} + \Delta\boldsymbol{\theta}_i. \tag{4.31}$$

In a special case, when the system has two scalar stochastic inputs, $\boldsymbol{\theta}_i^{(k+1)} = [Q_i^{(k+1)}, R_i^{(k+1)}]$, the log–normal distribution can be used for the perturbation. It can be realized by

$$
\begin{aligned}
Q_i^{(k+1)} &:= Q_i^{(k+1)} \exp(u_i), \quad u_i \sim \mathcal{N}\left(0;\, \alpha Q_i^{(k+1)}\right), \\
R_i^{(k+1)} &:= R_i^{(k+1)} \exp(w_i), \quad w_i \sim \mathcal{N}\left(0;\, \alpha R_i^{(k+1)}\right),
\end{aligned}
\tag{4.32}
$$

where α is an empirically set constant.

11) **Checking the termination conditions**

Conditions for terminating the algorithm can be the following

a) use the optimality tests and compare the qualitative characteristics of whiteness property of the innovation sequence. Finish, if the innovation sequence passes the whiteness test.

b) difference between the last and the current estimate $\left\|\hat{\boldsymbol{\theta}}^{(k-1)} - \hat{\boldsymbol{\theta}}^{(k)}\right\|$ is satisfactory small,

c) the maximum number of iterations *K* has been reached, i.e. $k = K$,

Otherwise, continue a new iteration $k := k + 1$ with Step 2) using the set $\mathcal{S}^{(k+1)}$ containing the new samples (4.31).

Next *k*

End of algorithm

4.5.1 Numerical simulations

In this section, several SISO systems (with two independent scalar noise sources and known matrices **A**, **C**, **G**) will be considered for simulations. Estimation process is repeated 50 times for several data generators. The Bayesian algorithm 4.4.1 uses a number of data given in the tables. The Monte Carlo resampling is repeated in 15 iterations, which is sufficiently large according to the selected systems. More iterations can improve the results and the localization of probability maximum, however the time consumption increases significantly.

Resulting statistics (mean μ and standard deviation σ in % of the true *Q*, *R* values) of the obtained estimates are shown in the following tables. The following settings will be used

for simulations. Initial grids for Q and R are logarithmic within interval (0.01; 15), each containing 20 points; set $\mathcal{S}$ contains 400 pairs. The perturbation parameter is empirically set to $\alpha = 0.012$. For each tested system, several combinations of the noise generator covariances will be applied marked as Q_r, R_r. Further, the two different data lengths will be used for a better demonstration of the algorithm consistency. For larger systems, more data are necessary to obtain satisfactory results. Therefore, we increased the number of data for systems of order 3 and higher. The estimated pair $[\hat{Q}, \hat{R}]$ is found by (4.25). This method might be preferable especially if the cpdf has a complicated shape which is the case even for small systems. Alternatively, both settings obtained by (4.25) and (4.26) can be compared using the optimality tests.

System 1: $\mathbf{A} = 0.5, \mathbf{C} = 1, \mathbf{G} = 1$. The resulting statistics are in Tab. 3.

Tab. 3 – System 1. Statistics of the repeated estimates.

$[Q_r; R_r]$	μ_Q		σ_Q [%]		μ_R		σ_R [%]	
data length	500	1,000	500	1,000	500	1,000	500	1,000
[10; 1]	10.14	10.50	11.2	7.7	0.91	0.77	84.2	59.7
[1; 10]	1.13	1.14	76.2	51.3	9.80	9.73	12.9	7.5
[0.1; 10]	0.30	0.19	383.0	272.0	9.70	10.02	7.4	5.7
[10; 0.1]	9.82	9.90	7.8	5.9	0.25	0.19	588.0	288.0
[1; 1]	0.99	1.00	19.2	15.5	1.00	1.01	19.2	13.0

System 2: $\mathbf{A} = \begin{bmatrix} 0.8 & 1 \\ 0 & -0.9 \end{bmatrix}$, $\mathbf{G} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\mathbf{C} = \begin{bmatrix} 1 & 1 \end{bmatrix}$. The resulting statistics are in Tab. 4.

Tab. 4 – System 2. Statistics of the repeated estimates.

$[Q_r; R_r]$	μ_Q		σ_Q [%]		μ_R		σ_R [%]	
data length	500	1,000	500	1,000	500	1,000	500	1,000
[10; 1]	9.80	9.97	9.9	5.9	1.36	1.21	163.3	110.9
[1; 10]	0.99	1.00	16.0	12.0	9.92	9.98	8.9	6.4
[0.1; 10]	0.10	0.11	50.9	42.6	10.06	9.94	6.7	5.3
[10; 0.1]	9.51	9.73	6.6	5.2	0.99	0.71	1214.8	1042.9
[1; 1]	0.98	1.01	14.1	6.1	0.99	0.99	31.9	16.6

System 3: $\mathbf{A} = \begin{bmatrix} 0.8 & 1 & 0 \\ 0 & -0.9 & 1 \\ 0 & 0 & 0.9 \end{bmatrix}$, $\mathbf{G} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, $\mathbf{C} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$. The resulting statistics are in Tab. 5.

Tab. 5 – System 3. Statistics of the repeated estimates.

$[Q_r; R_r]$	μ_Q		σ_Q [%]		μ_R		σ_R [%]	
data length	1,000	2,000	1,000	2,000	1,000	2,000	1,000	2,000
[10; 1]	9.82	9.93	5.1	3.0	1.47	1.23	163.7	120.0
[1; 10]	1.00	1.01	9.6	6.4	10.08	9.99	8.5	6.2
[0.1; 10]	0.1	0.1	18.5	14.0	10.00	9.99	5.6	3.2
[10; 0.1]	9.82	9.83	4.2	3.9	0.75	0.59	1295.9	777.1
[1; 1]	1.00	0.99	6.9	5.1	0.96	1.01	28.1	21.6

System 4: $\mathbf{A} = \begin{bmatrix} 0.9 & 0 & 1 & 1 \\ 0 & -0.8 & 0 & -1 \\ 0 & -0.1 & 0.85 & 1 \\ -0.2 & 0 & 0 & -0.9 \end{bmatrix}$, $\mathbf{G} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$, $\mathbf{C} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$. The resulting statistics are in Tab. 6.

Tab. 6 – System 4. Statistics of the repeated estimates.

$[Q_r; R_r]$	μ_Q		σ_Q [%]		μ_R		σ_R [%]	
data length	1,000	2,000	1,000	2,000	1,000	2,000	1,000	2,000
[10; 1]	9.95	9.93	4.6	3.5	0.91	1.08	70.8	55.5
[1; 10]	1.00	1.00	5.6	3.8	9.80	9.97	6.9	4.4
[0.1; 10]	0.10	0.10	7.9	6.2	9.98	9.97	5.5	2.8
[10; 0.1]	9.87	10.01	4.6	3.9	0.31	0.33	471.8	399.0
[1; 1]	1.01	1.01	6.0	4.2	0.97	0.97	13.3	9.3

The above tables offer results summary of extensive algorithm testing using linear time invariant systems of orders 1 to 4. It can be seen, that the algorithm is consistent, i.e. the standard deviation decreases with more data.

It has to be noted, that a higher standard deviation of the repeated estimates does not necessarily mean high standard deviation of the Kalman gain, which is of main interest. The quality of estimation depends on the sensitivity of the Kalman filter to the noise covariances. If the sensitivity is low, the Kalman gain accuracy can be satisfactory even for less accurate covariance estimates. For a brief demonstration of the varying sensitivity of the Kalman gain l on

the covariances, consider a simple scalar SISO system given by $a = 0.95, g = 1, c = 1, Q, R$. The sensitivity of *l* on the covariances is given by

$$\frac{\partial l}{\partial Q} = \frac{\partial l}{\partial P}\frac{\partial P}{\partial Q} + \frac{\partial l}{\partial Q},$$
$$\frac{\partial l}{\partial R} = \frac{\partial l}{\partial P}\frac{\partial P}{\partial R} + \frac{\partial l}{\partial R}, \tag{4.33}$$

where $\frac{\partial P}{\partial Q}, \frac{\partial P}{\partial R}, \frac{\partial l}{\partial Q}, \frac{\partial l}{\partial R}$ are obtained by differentiation of the solution of the algebraic Riccati equation and the formula for the Kalman gain

$$P = \frac{1}{2}\left(-R + a^2 R + Q + \sqrt{(R - a^2 R - Q)^2 + 4QR}\right), \; l = \frac{aP}{P + R}.$$

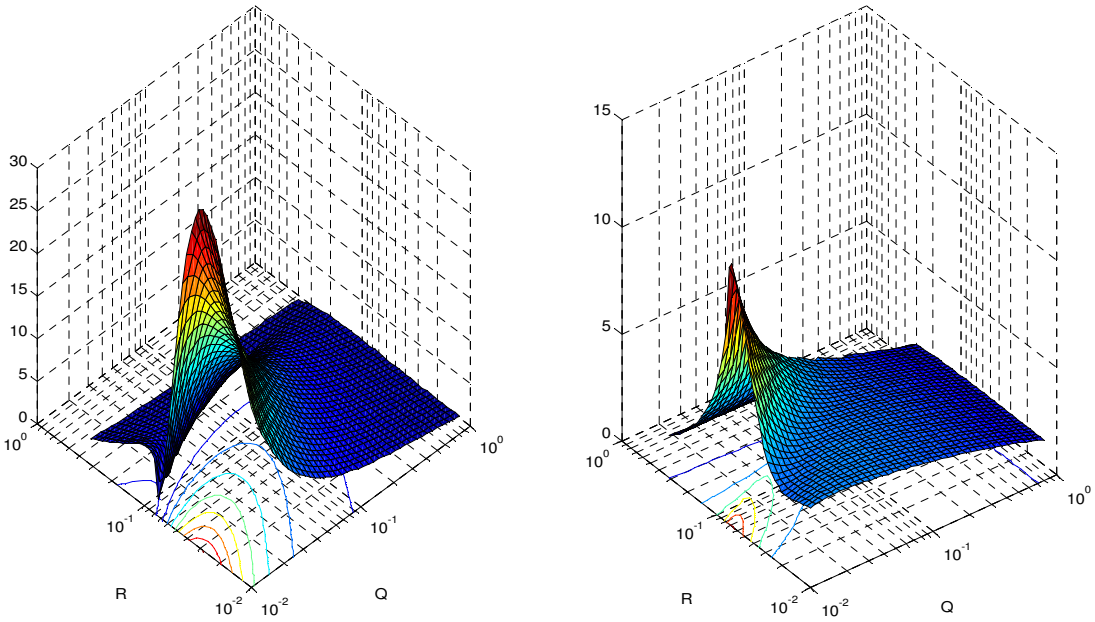

Figure 6 – Sensitivity of the Kalman gain $\frac{\partial l}{\partial Q}$ (left fig.) and $\frac{\partial l}{\partial R}$ (right fig.).

In Figure 6, it can be seen that the sensitivity decreases with increasing covariances. The sensitivity can be calculated for the parameters of the covariance matrices in advance, because it does not depend on the true noise covariance matrices. The sensitivity can be used to form the initial distribution of $\boldsymbol{\theta}_i \in \mathcal{S}$.

4.6 Recursive estimation algorithm

The algorithms described previously use all the available data for the estimation of the parameters of noise covariance matrices. After all the data was used to update the posterior conditional probability density of the parameters, a new set $\mathcal{S}$ of the parameters $\boldsymbol{\theta}$ were generated by the importance sampling method. Then, the Bayesian algorithm started again with the next iteration *k* calculating the posterior cpdf of the parameters on the newly generated set $\mathcal{S}$.

In this section, a recursive algorithm will be proposed using the incoming data for updating the parameter estimates at each time step. The set $\mathcal{S}$ is also updated at each time step, therefore the time instants *t* merge with the iterations *k*.

The recursive algorithm can alternatively run in parallel with the Kalman filter using the new data for estimating the parameters of the covariance matrices **Q** and **R**. Such configuration can be a form of adaptive Kalman filter which can deal with disturbances with time–varying covariance matrices. The recursive algorithm would be as follows.

Algorithm

1) **Initialization**

Create a set $\mathcal{S}$ containing n_S samples of the parameter vectors $\boldsymbol{\theta}_i^{(0)}$.

2) **Calculation of initial parameter estimates**

Run the algorithm described in Section 4.4.1 using the desired amount of data. Consider a set of parameter vectors instead of pairs *Q*, *R*. No update of the set $\mathcal{S}$ is done during this period. An initial shape of the cpdf $p\left(\boldsymbol{\theta}^{(0)} \mid \mathcal{Y}^{T_{\text{init}}}\right)$ is formed. The length of the start period T_{init} is empirical depending on the system order and the available data.

Do while new data $\mathbf{y}(t)$ is available

3) **Performing the Bayesian approach**

Perform the steps 2) to 7) of the algorithm in Section 4.4.1 using the current output value $\mathbf{y}(t)$ and the set $\mathcal{S}^{(t)}$ of the parameter vectors $\boldsymbol{\theta}_i^{(k)}$.

4) **Generating new parameter vectors $\boldsymbol{\theta}^{(k+1)}$**

The likelihood function requires all the data $\mathcal{Y}^{t-1}$ stored by the parallel Kalman filters, see (2.10). Although it is possible to calculate the whole trajectory of the state and output estimates for the Kalman filters set by $[\mathbf{Q}(\boldsymbol{\theta}_i^{(k+1)}), \mathbf{R}(\boldsymbol{\theta}_i^{(k+1)})]$, but it is not possible to calculate the cumulative likelihood for these new Kalman filters. For this reason, only several samples from $\mathcal{S}^{(t)}$ with the lowest probability are replaced by the newly generated samples to keep the maximum amount of information from the previously processed data. The number of replaced samples $\boldsymbol{\theta}_i^{(k)}$ from the set $\mathcal{S}^{(t)}$ is an empirically set constant. The main idea of this approach is to keep the members of $\mathcal{S}^{(t)}$ with a high posterior probability and replace those with low probability by newly generated vectors $\boldsymbol{\theta}_i^{(k+1)}$.

Select the number ν of the parameter vectors to be replaced in the each iteration. Generate ν new vectors $\boldsymbol{\theta}_i^{(k+1)}$ from the distribution $p\left(\boldsymbol{\theta}^{(k)} \mid \mathcal{Y}^t\right)$ using the importance sampling method

$$\boldsymbol{\theta}_i^{(k+1)} \sim p\left(\boldsymbol{\theta}^{(k)} \mid \mathcal{Y}^t\right). \tag{4.34}$$

Perturb the new vectors $\boldsymbol{\theta}_i^{(k+1)}$ (see the algorithm in Section 4.5, step 3). The number ν should not be higher than 2–4% of n_S to prevent the posterior cpdf quickly converge to a local maximum.

5) **Selection of the parameter vectors with a low probability to be replaced**

Copy all samples from $\mathcal{S}^{(t)}$ to $\mathcal{S}^{(t+1)}$. The new vectors $\boldsymbol{\theta}_i^{(t+1)}$ will replace the members of $\mathcal{S}^{(t+1)}$ with the lowest probability. Find ν members from the set $\mathcal{S}^{(t+1)}$ with the lowest posterior probability $p\left(\boldsymbol{\theta}^{(t)} \mid \mathcal{Y}^t\right)$ and assign indices $m_1 \ldots m_\nu$ to them. Insert the new samples $\boldsymbol{\theta}_i^{(t+1)}$ into the set $\mathcal{S}^{(t+1)}$ replacing the members with indices $m_1 \ldots m_\nu$. Further, the members $\boldsymbol{\theta}_{s_i}^{(t)}$ of the set $\mathcal{S}^{(t)}$ closest to the newly generated vectors $\boldsymbol{\theta}_i^{(t+1)}$

are found, indexed $s_1 \ldots s_\nu$. For each $\boldsymbol{\theta}_i^{(t+1)}$, find index s_i of $\boldsymbol{\theta}_{s_i}^{(t)} \in \mathcal{S}^{(t)}$ such, that the distance between $\boldsymbol{\theta}_i^{(t+1)}$ and $\boldsymbol{\theta}_{s_i}^{(t)}$ is minimal

$$s_i = \underset{j;\boldsymbol{\theta}_j^{(k)} \in \mathcal{S}}{\arg\min} \left(\left\| \boldsymbol{\theta}_j^{(k)} - \boldsymbol{\theta}_i^{(k+1)} \right\| \right), \tag{4.35}$$

where $\| \ \|$ can be 2–norm, ∞–norm or Frobenius f–norm.

6) **Assigning necessary information to the newly generated parameter vectors**

Now, the parallel Kalman filters with the new settings $[\mathbf{Q}(\boldsymbol{\theta}_i^{(k+1)}), \mathbf{R}(\boldsymbol{\theta}_i^{(k+1)})]$ need to be initiated. The new Kalman filters will not run from the time zero, because it would be extremely time consuming. They will inherit the values from the filters with indices s_i chosen according to their settings by the criterion (4.35).

The information about the members with indexes $s_1 \ldots s_\nu$ from the set $\mathcal{S}$ is copied to the newly generated Kalman filters. The newly generated members are inserted into the set $\mathcal{S}$ replacing the members with indices $m_1 \ldots m_\nu$, i.e.

$$\begin{aligned} \hat{\mathbf{x}}_{m_i}\left(t \mid t-1\right) &= \hat{\mathbf{x}}_{s_i}\left(t \mid t-1\right), \\ \mathbf{P}_{m_i}\left(t \mid t-1\right) &= \mathbf{P}_{s_i}\left(t \mid t-1\right), \\ p\left(\boldsymbol{\theta}_{m_i}^{(k+1)} \mid \mathcal{Y}^t\right) &= p\left(\boldsymbol{\theta}_{s_i}^{(k)} \mid \mathcal{Y}^t\right). \end{aligned} \tag{4.36}$$

7) **Estimation of the noise covariance matrices**

Find the parameter estimates using (4.28) or (4.29) and calculate the covariance matrices as a function of the parameters, $\mathbf{Q}(\hat{\boldsymbol{\theta}}^{(k)}), \mathbf{R}(\hat{\boldsymbol{\theta}}^{(k)})$.

8) **Increasing of the time index**

Increase the time index $t := t + 1$, and process the new data $\mathbf{y}(t)$. Return to step 3) .

Next t

End of algorithm

The recursive algorithm does not update the set $\mathcal{S}^{(t)}$ at each time step as the algorithm in Section 4.5 did, because it is significantly time consuming and the computational time increases with the amount of data. The recursive algorithm omits samples from $\mathcal{S}^{(t)}$ with the lowest probability and replaces them by newly generated samples. The advantage of the recursive algorithm is a possibility to adapt the covariance matrix estimates, if the true noise covariance matrices are time–varying.

4.7 Comparison of the methods for noise covariance estimation

In Section 4.2, the most recent methods for noise covariance estimation were shortly introduced. One of the key assumptions of ALS/scALS method is that the sample autocorrelation function given by (4.13) approaches its true value. This is theoretically true if the number of data approaches infinity. However, for finite data sets, the convergence of the sample autocorrelation function to its true values is of the order $1/N$ for $N \to \infty$, (Odelson *et al*., 2006), which is relatively slow. If the assumption of fast convergence is used, the developed methods will not give accurate results with small data sets. The problem is more significant in MIMO cases. This is shown in Matisko (2009), where tests with various systems were performed. Slow convergence can restrict the applicability of ALS method that works satisfactory only with small SISO systems.

Further, we compare the scALS method, Section 4.3, with Monte Carlo method described in this paper. System 1 used in the previous section and System 5 defined below will be employed. Estimation by both methods is repeated 50 times and then statistics of estimated covariances $\hat{Q}, \hat{R}$ are compared. The method scALS is tested with several settings that represent parameters for the initial Kalman gain calculation.

scALS is simulated with two different parameters *M*. Even when larger *M* was used, no significantly better results have been obtained. The M parameter sets how many lags of autocorrelation function of innovations will be used for *Q*, *R* estimation. When increasing this parameter, the solver may come to memory limits because the ALS algorithm uses Kronecker products and direct summations of high dimensions. This is especially true for the original ALS method of Odelson *et al*. The newer approach, scALS, reduces this problem and the al-

gorithm can handle larger systems. We also offer simulations results of the scALS+SDP approach, that is significantly better than the original ALS. This approach also enforces the positive definiteness of the resulting covariance matrices.

The simulations will use System 1 from the previous section and Systems 5, 6, 7 defined below. Estimation by both methods is repeated 50 times and then the statistics of the estimated covariances $\hat{Q}, \hat{R}$ are compared. The initial set $\mathcal{S}^{(0)}$ is obtained by a Cartesian combination of the sets $\mathcal{Q}$ and $\mathcal{R}$ each having 20 values with a logarithmic scale within interval (0.01; 15).

Tab. 7 – Comparison of MC method and scALS method using System 1 and 1000 samples.

Q_r=10 R_r=1	μ_Q	σ_Q [%]	μ_R	σ_R [%]
MC method, $[\hat{Q}, \hat{R}]_{\mathrm{MAP}}$	10.25	7.7	0.77	59.7
MC method, $[\hat{Q}, \hat{R}]_{\mathrm{MS}}$	10.62	6.1	0.99	51.4
Recursive MC, $[\hat{Q}, \hat{R}]_{\mathrm{MAP}}$	9.84	10.0	1.10	70.3
Recursive MC, $[\hat{Q}, \hat{R}]_{\mathrm{MS}}$	10.05	7.6	0.98	51.5
ALS, N = 10 Q_0=1, R_0 =1	9.88	11.6	0.96	84.0
ALS, N = 20 Q_0=1, R_0 =1	10.05	12.2	1.00	81.9
ALS, N = 20 Q_0=1,R_0 =10	10.15	13.9	0.84	117.4

ALS method was simulated with several settings representing parameters for the calculation of an initial Kalman gain. Further, considering the results in Tab. 7 and Tab. 8, it can be seen that MC method has an acceptable accuracy in noise covariance estimation and the obtained results have a smaller standard deviation than the values obtained by ALS. Tab. 7 and Tab. 8. contain also the simulation results of the recursive algorithm. It can be seen, that the accuracy of the recursive algorithm is comparable to the non–recursive one.

ALS was tested with two different parameters N, that sets how many lags of the autocorrelation function of innovations will be used for $\hat{Q}, \hat{R}$ calculation. Even when larger N was used, no significantly better results have been obtained. However, if N is increased, the solver may come to memory limits because the ALS algorithm uses Kronecker products and direct summations of high dimensions. This is especially true for the original ALS method of Odel-

son *et al*. Comparing the results in Tab. 7 and Tab. 8, we can conclude that the Bayesian approach together with Monte Carlo is more accurate and returns less outliers than ALS. Moreover, it does not return negative definite estimates.

Further, Bayesian Monte Carlo algorithm will be tested on several MIMO systems and compared to the results of the ALS method.

System 5: $\mathbf{A} = \begin{bmatrix} 0.9 & 0 & 1 & 1 \\ 0 & -0.8 & 0 & -1 \\ 0 & -0.1 & 0.85 & 1 \\ -0.2 & 0 & 0 & -0.9 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 0.1 \\ 1 \\ 10 \\ 0.1 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}$. The resulting statistics are in Tab. 8.

Tab. 8 – Comparison of MC method, Recursive MC method and ALS method using System 5 and 2000 samples.

Q_r=10 R_r=1	μ_Q	σ_Q [%]	μ_R	σ_R [%]
MC method, $[\hat{Q}, \hat{R}]_{\mathrm{MAP}}$	9.98	5.2	1.29	214.2
MC method, $[\hat{Q}, \hat{R}]_{\mathrm{MS}}$	9.93	3.3	2.06	330.5
Recursive MC, $[\hat{Q}, \hat{R}]_{\mathrm{MAP}}$	9.92	4.6	4.43	328.3
Recursive MC, $[\hat{Q}, \hat{R}]_{\mathrm{MS}}$	9.95	3.8	2.60	119.4
ALS, N = 20 Q_0=1, R_0 =1	10.01	6.8	–0.50	781.3
ALS, N = 20 Q_0=1, R_0 =10	9.59	5.6	1.93	956.1
ALS, N = 30 Q_0=1, R_0 =10	9.56	4.7	1.98	731.6
ALS+SDP, N = 20 Q_0=1, R_0 =1	9.97	3.7	4.15	547.6

System 6: $\mathbf{A} = \begin{bmatrix} 0.8 & 1 \\ 0 & -0.9 \end{bmatrix}$, $\mathbf{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. The resulting statistics are in Tab. 9.

From Tab. 7 and Tab. 8 it can be seen that scALS method estimation quality depends on the prior setting. The dependency on the prior information limits the usage of scALS. Further, considering results in Tab. 7 and Tab. 8, it can be seen that MC method has an acceptable accuracy in noise covariance estimation and the obtained results have smaller standard deviation than the values obtained by scALS. Tab. 7 and Tab. 8 contain simulation results of the

recursive algorithm. It can be seen, that the accuracy of the recursive algorithm is comparable to the non–recursive one. The improvement of the scALS+SDP estimation quality can be also highlighted. The variance among repeated estimates is lower than using the original ALS and there are no negative results.

Tab. 9 – Comparison of Recursive MC method and scALS method using System 6 and 3000 samples. The mean and standard deviation is shown for each matrix member individually.

$\mathbf{Q}_r = \begin{bmatrix} 10 & 0 \\ 0 & 5 \end{bmatrix}$ $\mathbf{R}_r = \begin{bmatrix} 1.5 & 0 \\ 0 & 3 \end{bmatrix}$	$\begin{bmatrix} \mu_{q_{11}} \\ \mu_{q_{22}} \end{bmatrix}$	$\begin{bmatrix} \sigma_{q_{11}} \\ \sigma_{q_{22}} \end{bmatrix}$ [%]	$\begin{bmatrix} \mu_{r_{11}} \\ \mu_{r_{22}} \end{bmatrix}$	$\begin{bmatrix} \sigma_{r_{11}} \\ \sigma_{r_{22}} \end{bmatrix}$ [%]
Recursive MC, $[\hat{\mathbf{Q}}, \hat{\mathbf{R}}]_{\text{MAP}}$	$\begin{bmatrix} 9.94 \\ 5.00 \end{bmatrix}$	$\begin{bmatrix} 10.8 \\ 10.0 \end{bmatrix}$	$\begin{bmatrix} 1.63 \\ 3.02 \end{bmatrix}$	$\begin{bmatrix} 41.9 \\ 10.9 \end{bmatrix}$
Recursive MC, $[\hat{\mathbf{Q}}, \hat{\mathbf{R}}]_{\text{MS}}$	$\begin{bmatrix} 9.96 \\ 5.00 \end{bmatrix}$	$\begin{bmatrix} 12.3 \\ 12.4 \end{bmatrix}$	$\begin{bmatrix} 1.61 \\ 3.03 \end{bmatrix}$	$\begin{bmatrix} 46.9 \\ 11.8 \end{bmatrix}$
ALS, $N = 30$ $Q_0=\mathbf{I}_2$, $R_0=\mathbf{I}_2$	$\begin{bmatrix} 15.69 \\ 11.63 \end{bmatrix}$	$\begin{bmatrix} 8.8 \\ 6.6 \end{bmatrix}$	$\begin{bmatrix} -3.55 \\ -3.14 \end{bmatrix}$	$\begin{bmatrix} 44.6 \\ 16.2 \end{bmatrix}$
ALS+SDP, $N = 30$ $Q_0=\mathbf{I}_2$, $R_0=\mathbf{I}_2$	$\begin{bmatrix} 9.88 \\ 5.02 \end{bmatrix}$	$\begin{bmatrix} 3.9 \\ 5.9 \end{bmatrix}$	$\begin{bmatrix} 3.61 \\ 3.00 \end{bmatrix}$	$\begin{bmatrix} 278.4 \\ 7.5 \end{bmatrix}$

System 7: $\mathbf{A} = \begin{bmatrix} 0.9 & 0 & 1 & 1 \\ 0 & -0.8 & 0 & -1 \\ 0 & -0.1 & 0.85 & 1 \\ -0.2 & 0 & 0 & -0.9 \end{bmatrix}$, $\mathbf{G} = \begin{bmatrix} 0 & 0 & 0.1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$. The resulting statistics are in Tab. 10.

It can be seen in Tab. 9 and Tab. 10, that the Recursive MC method calculates the results with satisfactory accuracy, however they are slightly worse than the non–recursive ones. This is caused by slower updating of the posterior cpdf compared to the algorithm in Section 4.5, which resamples the whole set $\mathcal{S}$ in each iteration.

The original ALS method can return negative definite estimates as demonstrated by the simulation results. The ALS+SDP enforces the positive definiteness and also improves the quality of the estimates. By comparing the results, it can be seen that the Bayesian and Monte Carlo algorithm has comparable or better results than the ALS+SDP method.

Tab. 10 – Comparison of Recursive MC method and scALS method using System 7 and 3000 samples. The mean and standard deviation is shown for each matrix member individually.

$\mathbf{Q}_r = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 7 \end{bmatrix}$ $\mathbf{R}_r = \begin{bmatrix} 1.5 & 0 \\ 0 & 3 \end{bmatrix}$	$\begin{bmatrix} \mu_{q_{11}} \\ \mu_{q_{22}} \\ \mu_{q_{33}} \end{bmatrix}$	$\begin{bmatrix} \sigma_{q_{11}} \\ \sigma_{q_{22}} \\ \sigma_{q_{33}} \end{bmatrix}$ [%]	$\begin{bmatrix} \mu_{r_{11}} \\ \mu_{r_{22}} \end{bmatrix}$	$\begin{bmatrix} \sigma_{r_{11}} \\ \sigma_{r_{22}} \end{bmatrix}$ [%]
Recursive MC, $[\hat{\mathbf{Q}}, \hat{\mathbf{R}}]_{\mathrm{MAP}}$	$\begin{bmatrix} 10.89 \\ 3.08 \\ 6.98 \end{bmatrix}$	$\begin{bmatrix} 9.0 \\ 9.0 \\ 6.3 \end{bmatrix}$	$\begin{bmatrix} 1.36 \\ 2.78 \end{bmatrix}$	$\begin{bmatrix} 10.4 \\ 14.9 \end{bmatrix}$
ALS, $N = 30$ $Q_0=\mathbf{I}_2$, $R_0 =10\cdot\mathbf{I}_2$	$\begin{bmatrix} 85.14 \\ 31.40 \\ 1.23 \end{bmatrix}$	$\begin{bmatrix} 56.8 \\ 66.0 \\ 4.8 \end{bmatrix}$	$\begin{bmatrix} -13.50 \\ -23.19 \end{bmatrix}$	$\begin{bmatrix} 70.0 \\ 77.8 \end{bmatrix}$
ALS, $N = 30$ $Q_0=\mathbf{I}_2$, $R_0 =\mathbf{I}_2$	$\begin{bmatrix} 10.21 \\ 24.05 \\ 8.40 \end{bmatrix}$	$\begin{bmatrix} 16.5 \\ 44.1 \\ 7.7 \end{bmatrix}$	$\begin{bmatrix} -5.26 \\ -0.18 \end{bmatrix}$	$\begin{bmatrix} 29.7 \\ 46.3 \end{bmatrix}$
ALS+SDP, $N = 30$ $Q_0 = \mathbf{I}_2$, $R_0 = \mathbf{I}_2$	$\begin{bmatrix} 10.17 \\ 3.07 \\ 6.96 \end{bmatrix}$	$\begin{bmatrix} 8.7 \\ 9.0 \\ 5.3 \end{bmatrix}$	$\begin{bmatrix} 1.46 \\ 2.89 \end{bmatrix}$	$\begin{bmatrix} 17.3 \\ 31.5 \end{bmatrix}$

4.8 Computation complexity comparison

In this subsection, a brief comparison of computation complexity and time consumption will be given for ALS and the Bayesian method. The ALS method uses a least–square minimization of a linear function, which allows to formulate the solution in a closed form. If additional constraints are required (e.g. positive definiteness), the solution leads to a semi definite programming problem. Solving SDP for linear problems is well known and the algorithms are fast. The speed of calculation is a significant advantage of the ALS method.

The Bayesian and Monte Carlo approach is an iterative method. The probability distribution function is calculated for each sample of the set $\mathcal{S}$. This calculation is repeated for each time instant. After the pdf is formed on the set $\mathcal{S}$, the set is resampled using Monte Carlo

techniques and the computation is repeated with the new set $\mathcal{S}$. Comparing to the ALS method, this approach is significantly more time consuming. The computation time increases with a larger set $\mathcal{S}$ and a larger amount of data. However, the computational process can be easily parallelized. The time demands can be reduced using the recursive approach.

The ALS method uses Kronecker products and direct summations for the noise covariance calculation. These matrices are of large dimensions even for small problems and they grow rapidly with higher complexity of the system. The calculation requires advanced mathematical tools, which can handle large matrices, matrix inversions and can solve SDP problems. The Bayesian method has significantly less memory demands and the calculation does not require advanced mathematical tools. This can be an advantage especially for practical implementation at industry environment.

4.9 Comments on convergence of the Bayesian algorithm

The algorithm described in Section 4.4 is built on the Bayesian principles which guarantee effective usage of the information contained in data. It is known that if the efficient estimates exist, then one of them can be found by the maximum likelihood approach. Considering the Bayesian principles, the limitation of the estimation quality can be calculated using the Cramér–Rao bounds. The complete derivation of the Cramér–Rao bounds for the estimation of the noise covariances can be found in Chapter 6. The convergence of the Bayesian method is compared to the Cramér–Rao bounds, there.

5. Detection of colored noise

5.1 Introduction to the detection of colored noise

This chapter introduces detection of a colored noise. One of the key assumptions for the optimality of the Kalman filter is white noise entering the system. If this condition is satisfied, the true covariances of the noise can be found leading to the optimal Kalman gain calculation. In this case, KF is working optimally and the innovation sequence is white.

In many practical problems a colored noise has to be considered. For example, disturbances with particular frequencies, e.g. sea waves periodically affecting ship movement, wind affecting planes, etc. If the color property of the disturbances is neglected, it can significantly influence the quality of state estimation and further the control performance. The algorithms described in Chapter 4 will find some estimates of the noise covariances that minimizes the criteria or maximizes the probability, but the resulting quality of the state estimates is questionable.

Linear systems disturbed by a colored noise has had only a minor attention during past decades. In most practical problems, it is assumed that the disturbances are white and normal. One of the first papers dealing with this topic was written by Bryson and Johansen (1965). This paper considers continues time and discrete time systems and describe the main points of implementing the coloring filter and highlights the singular cases. This paper, however, does not discuss detection of the colored noise. The paper of Salzman and Teunissen (1990) summarizes principles of color noise detection using auto-regressive models in the time domain and the signal spectra in the frequency domain.

The newer brief paper of Popescu and Zeljkovic (1998) uses the incorporation of the coloring filter into the Kalman filter to improve SNR of a speech signal. It is demonstrated that considering the color property of the background noise can improve speech detection.

This chapter demonstrates an approach for the detection of colored noise. The time and frequency domain will be considered. Further, an appropriate shaping filter will be found. The

shaping filter can be incorporated into the KF and the whole system can be used as a state observer in the standard way. Application of the approach will be demonstrated on a numerical example. The state estimation quality will be commented.

Throughout the chapter, we will concentrate on the process noise. Assuming the measurement noise to be white is reasonable in many practical applications, because it is caused mainly by sensors inaccuracy. Moreover, we will discuss equivalency of the process and measurement noise models.

This chapter provides a basic overview of colored noise analysis and proposes approaches that can be used for solving this problem. The chapter does not contain rigorous solutions of the non–convex optimization problems, because it exceeds the topic of this thesis. Non–convex optimization is a well studied field and its connection with identification and estimation theory may bring new theoretical results.

For the ease of demonstration, only a simple SISO systems will be used. The analysis of higher order MIMO systems requires a further research.

5.2 Problem formulation

Consider having a set of scalar output data $y(0),...,y(T)$. The goal is to detect whether the data set was generated by a stable linear dynamic system of order *n* given by system matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{c} \in \mathbb{R}^n$ excited by white noise. For ease of analysis we employ a system in the innovation form

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{k}\varepsilon(t), \\ y(t) &= \mathbf{c}^T\mathbf{x}(t) + \varepsilon(t), \end{aligned} \tag{5.1}$$

where $\mathbf{k} \in \mathbb{R}^n$ and $\varepsilon(t)$ is the entering noise with covariance R_s. The transfer function of system (5.1) can be calculated using $\mathcal{Z}$–transform[6]

[6] This calculation method does not guarantee that the resulting transfer function has a minimal phase. However, due to the Spectral Factorization Theorem, it is always possible to find such k, that the resulting transfer function is stable with a minimal.

$$H(z) \triangleq \mathbf{c}^T \left(z\mathbf{I} - \mathbf{A}\right)^{-1} \mathbf{k} + 1 = \underbrace{\left[\frac{1}{\det\left(z\mathbf{I} - \mathbf{A}\right)} \mathbf{c}^T \operatorname{adj}\left(z\mathbf{I} - \mathbf{A}\right)\right]}_{\mathbf{F}} \mathbf{k} + 1, \tag{5.2}$$

where adj is the adjugate operator that forms a polynomial matrix with polynomials of maximal order $n-1$. Multiplying $(z\mathbf{I} - \mathbf{A})^{-1}$ and vector **c** forms a set of transfer functions that we will call the *system dynamic modes* or *base functions*, $\mathbf{F} = \left[F_1(z) \quad \dots \quad F_n(z)\right]$. The transfer function can be written as a linear combination of the known functions **F** and unknown input vector **k**

$$H(z) = \left[F_1(z) \quad F_2(z) \quad \cdots \quad F_n(z)\right] \mathbf{k} + 1. \tag{5.3}$$

The complex conjugate system is given by

$$H^*(z^{-1}) \triangleq \mathbf{k}^T \underbrace{\left(z^{-1}\mathbf{I} - \mathbf{A}\right)^{-T} \mathbf{c}}_{\mathbf{F}^*} + 1. \tag{5.4}$$

A spectral density (SD) of the output is defined as

$$S_{y,y}(z) \triangleq H(z) \cdot R_\varepsilon \cdot H^*(z^{-1}) \tag{5.5}$$

where $z = \exp(j\omega T_s), \omega \in \mathbb{R}$ and T_s is a sampling period. Spectral density $S_{y,y}(z)$ transformed to the time domain is of the form

$$R_{y,y} = h * h^* \cdot R_\varepsilon, \tag{5.6}$$

where $*$ is the convolution operator and $h(t), h^*(t)$ are impulse responses of systems $H(z), H^*(z^{-1})$ and l is the lag of autocorrelation function. The SD is symmetric with respect to *y*–axis. The definition (5.5) can be written using the base functions as

$$S_{y,y}(z) = H(z)H^*(z^{-1}) \cdot R_\varepsilon, \tag{5.7}$$

$$S_{y,y}(z) = \left(\begin{bmatrix} F_1(z) & F_2(z) & \cdots & F_n(z) \end{bmatrix}\mathbf{k} + 1\right) \cdot \left(\begin{bmatrix} F_1^*(z^{-1}) & F_2^*(z^{-1}) & \cdots & F_n^*(z^{-1}) \end{bmatrix}\mathbf{k} + 1\right) \cdot R_\varepsilon \quad (5.8)$$

leading to a formula

$$S_{y,y}(z) = R_\varepsilon \cdot \left(\sum_{\forall i,j \le n} k_i k_j F_i F_j^* + \sum_{\forall i \le n} k_i \left(F_i + F_i^* \right) + 1 \right), \quad (5.9)$$

where $\mathbf{k} = \begin{bmatrix} k_1 & \dots & k_n \end{bmatrix}^T$.

Transforming the spectral density of the output to the time domain requires the multiplication between the functions in (5.9) to be replaced by a convolution. For this purpose, the following normalized autocorrelation functions can be defined which are analogous to the convolution

$$\begin{aligned} R_{f_1,\delta}(l) &\triangleq \sum_{i=0}^{N-l-1} f_1(i)\delta(i+l), \\ R_{\delta,f_1}(l) &\triangleq \sum_{i=0}^{N-l-1} \delta(i) f_1(i+l), \\ R_{f_1,f_2}(l) &\triangleq \sum_{i=0}^{N-l-1} f_1(i) f_2(i+l), \end{aligned} \quad (5.10)$$

where $\delta(t)$ is the Kronecker delta function. Using the definitions (5.10) it holds

$$\begin{aligned} \mathcal{Z}^{-1}\left\{ F_1(z) F_2^*(z^{-1}) \right\} &= R_{f_1,f_2}, \\ \mathcal{Z}^{-1}\left\{ F_1(z) + F_1^*(z^{-1}) \right\} &= R_{f_1,\delta} + R_{\delta,f_1}. \end{aligned} \quad (5.11)$$

where $f_1(t), f_2(t)$ are the impulse responses of the systems given by $F_1(z), F_2(z)$. Functions $R_{f_i,\delta}, R_{\delta,f_i}$ are not symmetric themselves with respect to *y*–axis, but their sum is symmetric. The described spectral density and autocorrelation function will be used in the following sections to detect the color property of the entering noise.

5.3 Detection in the time domain

In the previous section, it was shown that the autocorrelation function of the output signal is formed by a set of base functions. The base functions are defined by the system modes and are combined by the output vector **c**. If there is only white noise entering the system, no other modes can appear on the output except the base ones. Now, the goal is to check if the output sample autocorrelation function can be formed by the base functions. If not, that means that the entering noise is not white and it adds some other dynamics to the output signal. The shaping filter, which forms the colored noise from a white noise, is to be found. It can be further used as a part of the Kalman filter tunable by the standard way using the white noise assumption.

In this section, an autocorrelation functions will be used, that depend on unknown parameters and lag l. The unknown parameters will be marked in brackets and lag l in the upper index. The lower index states what signals have been used for the autocorrelation calculation.

To start the analysis in the time domain, the formula (5.9) has to be transformed using (5.10) to the time domain

$$R_{y,y}^{l}(\mathbf{k}) = R_{\varepsilon} \cdot \left(\sum_{\forall i,j \le n} k_i k_j R_{f_i,f_j} + \sum_{\forall i \le n} k_i \left(R_{f_i,\delta} + R_{\delta,f_i} \right) + R_{\delta,\delta} \right). \tag{5.12}$$

Now, the gain **k** is to be found. For this purpose the following quadratic criterion will be used

$$\left[\hat{\mathbf{k}} \quad \hat{R}_{\varepsilon} \right] = \arg\min_{\mathbf{k},R_{\varepsilon}} \left\{ \left\| \hat{R}_{y,y}^{l} - R_{y,y}^{l}(\mathbf{k}) \right\|_2^2 \right\}, \tag{5.13}$$

where $\mathbf{k} = \begin{bmatrix} k_1 & \dots & k_n \end{bmatrix}^T$ and the sample autocorrelation function of the output $\hat{R}_{y,y}^{l}(\mathbf{k})$ is calculated using the measured data $y(t)$ for $l = 1,\dots,M$ by

$$\hat{R}_{y,y}^{l} = \frac{1}{N} \sum_{i=0}^{N} y(i) y(i+l), \tag{5.14}$$

where a reasonable choice for *M* is 10 – 20% of *N* depending on the amount of data.

Solving of (5.13) is a non–convex problem due to multiplication between the unknown coefficients and the solution is not unique. Further, the sample autocorrelation function $\hat{R}^l_{y,y}$ is compared to the true autocorrelation of the system output using the identified parameters $\hat{\mathbf{k}}, \hat{R}_\varepsilon$. If the sequence of residues of $\hat{R}^l_{y,y} - R^l_{y,y}(\hat{\mathbf{k}}, \hat{R}_\varepsilon)$ is white, it can be concluded that

a) there is no additional dynamics in the output signal except of the system modes and
b) the process noise entering the system is white.

If the residues do not form a white sequence, the dynamic modes of the noise shaping filter are to be found, which would improve the fitting of the autocorrelation function $R^l_{y,y}(\hat{\mathbf{k}}, \hat{R}_\varepsilon)$. This will be demonstrated on a numerical example.

5.3.1 Numerical example

As a data generator, consider a stable linear system of order *n* with a single output and a noise shaping filter of order n_F given by

$$\begin{aligned} \mathbf{f}(t+1) &= \mathbf{A}_F \mathbf{f}(t) + \mathbf{g}_F w(t), \\ v(t) &= \mathbf{c}_F^T \mathbf{f}(t) + w(t), \end{aligned} \tag{5.15}$$

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{g}v(t), \\ y(t) &= \mathbf{c}^T \mathbf{x}(t) + e(t), \end{aligned} \tag{5.16}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{f}(t) \in \mathbb{R}^{n_F}$, $v(t) \in \mathbb{R}$, $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{g} \in \mathbb{R}^n$, $\mathbf{A}_F \in \mathbb{R}^{n_F \times n_F}$, $\mathbf{g}_F \in \mathbb{R}^{n_F}$, $\mathbf{c}_F \in \mathbb{R}^{n_F}$. The noise sequences $w(t)$, $e(t)$ are white. The process is described by the model with states $\mathbf{x}(t)$ and the shaping filter is described by the states $\mathbf{f}(t)$. Alternatively, more stochastic inputs can be used with appropriate matrices $\mathbf{G}$ and $\mathbf{G}_F$ instead of vectors $\mathbf{g}$, $\mathbf{g}_F$. Matrix $\mathbf{A}$ and vector $\mathbf{c}$ are considered to be known. If it is detected that the entering noise is colored, the unknown shaping filter needs to be identified.

For a numerical example, consider a discrete linear system (5.16) with the sampling period 1s given by

$$\mathbf{A} = \begin{bmatrix} 0.9 & -0.8 \\ 0.2 & 0.8 \end{bmatrix}, \mathbf{g} = \begin{bmatrix} 0.1 \\ 0.05 \end{bmatrix}, \mathbf{c}^T = \begin{bmatrix} 4 & 0.1 \end{bmatrix}. \tag{5.17}$$

The System has two base functions and the transfer function is of the form

$$H(z) = \left[\underbrace{\frac{4z - 3.22}{z^2 - 1.7z + 0.88}}_{F_1(z)} \quad \underbrace{\frac{0.1z - 3.29}{z^2 - 1.7z + 0.88}}_{F_2(z)} \right] \mathbf{k} + 1. \tag{5.18}$$

The base functions can be transformed to the time domain using the impulse response of the base functions in the $\mathcal{Z}$–transform. Assume having 8200 output data samples at hand. Using criterion (5.13) and autocorrelation (5.14), the parameters $\hat{\mathbf{k}}, \hat{R}_\varepsilon$ are found. Now, the sample autocorrelation function can be compared to the identified one with the parameters $\hat{\mathbf{k}} = \begin{bmatrix} 0.84 & 0.50 \end{bmatrix}^T$ and $\hat{R}_\varepsilon = 1.05$. The resulting graphs are in Figure 7.

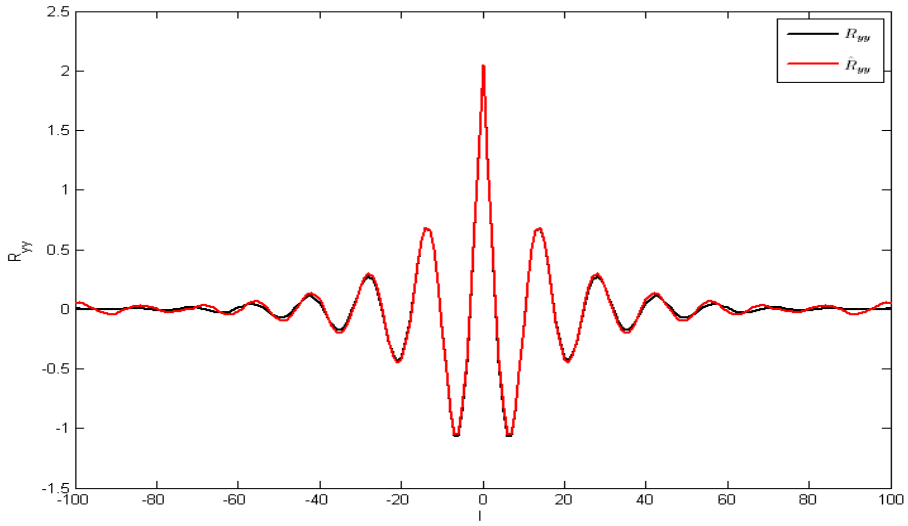

Figure 7 – Identified $R^l_{y,y}(\hat{\mathbf{k}}, \hat{R}_\varepsilon)$ and measured $\hat{R}^l_{y,y}$ autocorrelation function of the output data. The entering process noise is white.

It can be seen that the identified pair $\hat{\mathbf{k}}, \hat{R}_\varepsilon$ together with the system matrices are able to form the sample autocorrelation function of the measured data. It can be concluded, that the system is entered by white process noise. The Kalman filter for this system can be formed by system

matrices **A**, **c**. The noise covariances for tuning the filter can be obtained by the Bayesian method or ALS method, Chapter 4.

As a second example, consider colored noise entering the system (5.17). Repeating the same calculation process as in the previous example results in the following graphs, Figure 8. It is obvious that the sample autocorrelation function cannot be formed by the modes of system (5.17). Alternatively, we can plot impulse responses of the base functions and see, that some modes in $\hat{R}^l_{y,y}$ are missing in the system's dynamics. It can be concluded, that the system is either badly identified or the entering noise is colored. If we have prior knowledge about the system order, e.g. based on the physical background, we can conclude that the process noise is colored. The appropriate shaping filter can be found by optimization methods. The aim is to add new dynamic mode(s) such that $R^l_{y,y}(\hat{\mathbf{k}}, \hat{R}_\varepsilon)$ will fit better to $\hat{R}^l_{y,y}$. One possible approach will be shown in the following section.

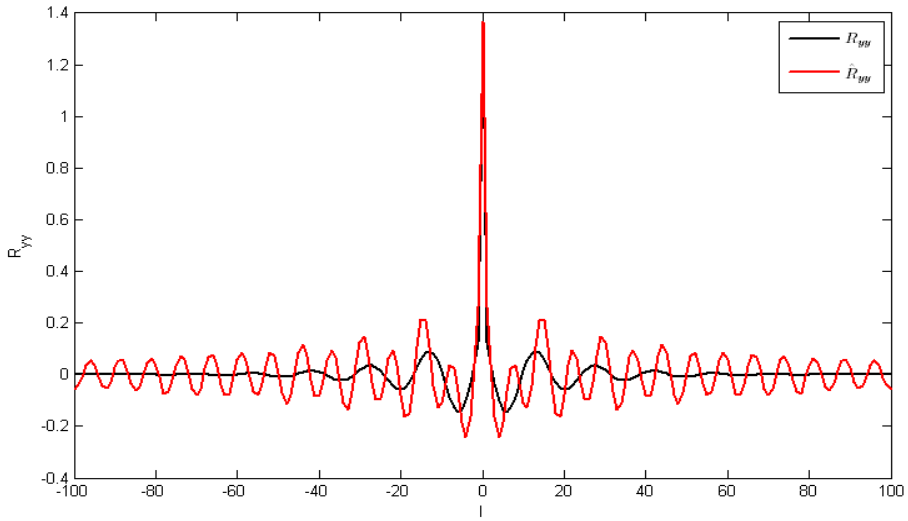

Figure 8 – Identified $R^l_{y,y}(\hat{\mathbf{k}}, \hat{R}_\varepsilon)$ and measured $\hat{R}^l_{y,y}$ autocorrelation function of the output data. The entering process noise is colored.

5.4 Detection in the frequency domain

In this section, a spectrum and a spectral density will be of interest for the color property detection. The transfer function of a dynamic system (5.1) can be calculated using (5.2) and the corresponding output spectral density using (5.5).

Consider again the system (5.17) excited by white noise. The estimated spectral density of the measured data, calculated by (5.14), is shown in Figure 9 together with the base functions of the system. In Figure 9, there are analogous graphs of the spectral densities. It can be seen, that the spectral density of the measured output can be formed by combining the base functions using vector **k**. It can be concluded that the entering noise is white and the system is well identified. Alternatively, vector **k** can be found by an analogous criterion as (5.13) with the cross–covariances replaced by spectral densities.

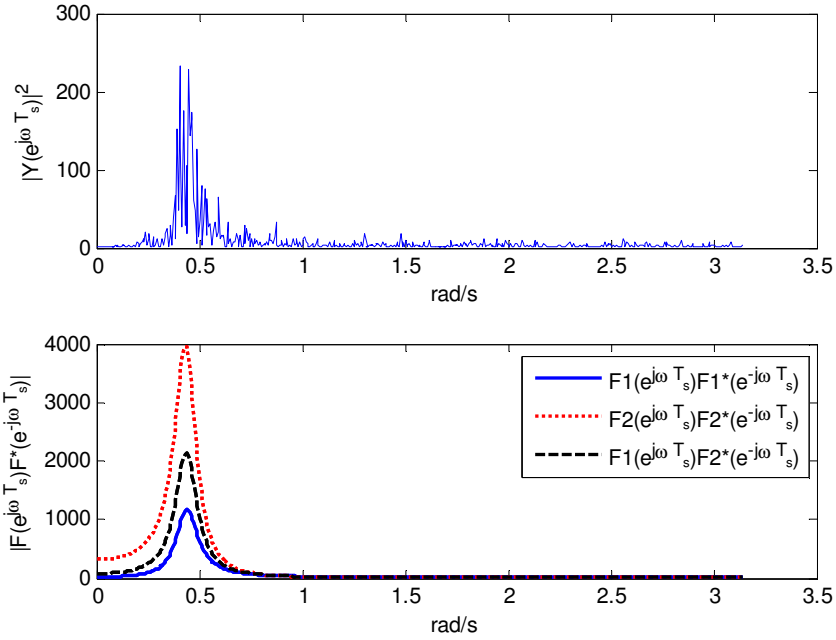

Figure 9 – a) One-side spectral density of the measured output signal. b) spectral density of the base functions. White noise is entering the system.

Now, consider the system (5.17) excited by colored noise. The measured spectrum can be seen in Figure 10. Compare the shape of the measured spectrum with the base functions. It is obvious that no combination of the base functions will lead to the measured spectrum. There is an additional dynamic mode with a frequency around 1.19 rads^{-1} that is entering the system

as colored noise. The shaping filter is to be found with such base functions that will be able to fit the peak around the 1.19 rads^{-1}.

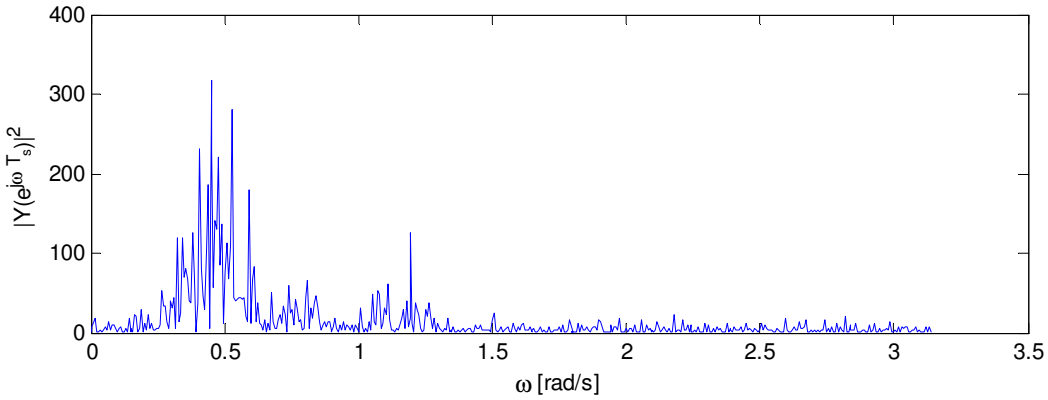

Figure 10 – A spectral density of the measured output signal when the system is excited by colored noise.

5.5 Shaping filer identification

Detection of colored noise was explained in the previous sections. The next problem is the identification of the noise shaping filter and its incorporation into the system model. The goal is to find such shaping filter $\hat{F}_{n_F}$ and parameters $\hat{\mathbf{k}}, \hat{R}_e$ that difference between the estimated and the measured spectral density is minimal.

$$\left[\hat{\mathbf{k}} \quad \hat{R}_e \quad \omega_2, \omega_1, \xi\right] = \underset{\mathbf{k}, R_e, \omega_2, \omega_1, \xi}{\arg\min} \left\{ \left\| \hat{S}_{y,y}(\omega) - S_{y,y}(\omega) \right\|_2^2 \right\}, \tag{5.19}$$

where ω_2, ω_1, ξ are the parameters of the transfer function F_{n_F} of the order n_F is an order of the shaping filter. The shaping filter can be searched in a particular form, e.g. general 2$^{\text{nd}}$ order system with one zero

$$F_{n_F} = \frac{z + \omega_2^2}{z^2 + 2\xi\omega_1 z + \omega_1^2}. \tag{5.20}$$

The optimization problem (5.19) is non–convex and therefore difficult for solving. However, inspecting the measured spectral density, e.g. Figure 10, can provide sufficient prior values

for the unknown parameters of (5.20). The form of the shaping filter (5.20) also provides the boundaries of the parameters such that the filter is stable.

The shaping filter can be, alternatively, found by hand using a spectral density (SD) of measured output. Compare the SD of the measured output signal and the base functions, Figure 10. We can model the part of the SD above frequency 0.15 Hz with a peak at 0.19 Hz. This part of the SD can be satisfactory fitted by the transfer function

$$F_{n_F}(z) = \frac{z+8}{z^2 - 0.6z + 0.84}. \tag{5.21}$$

The SD of the measured signal can be compared to the system base functions obtained by splitting (5.21) into two base functions of order 1, Figure 11.

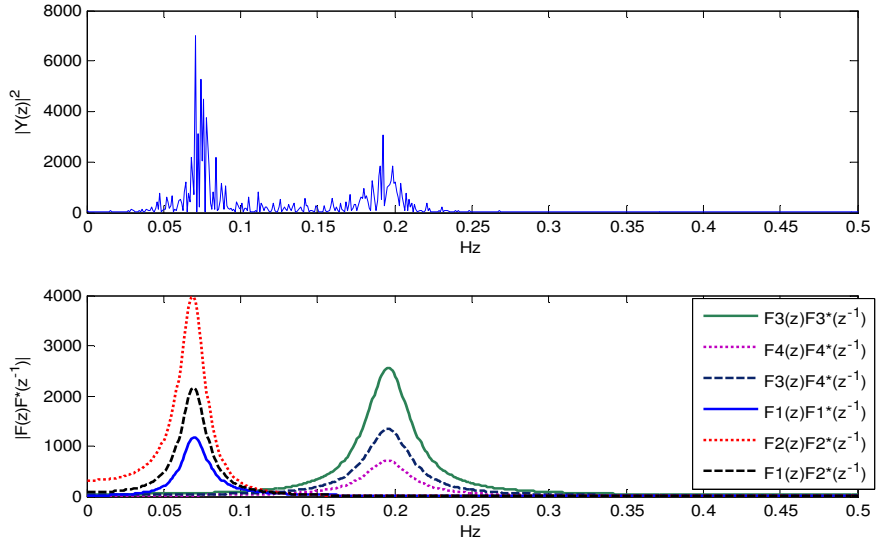

Figure 11 – a) spectral density of the measured output signal. b) spectral density of the base functions: the system – F1, F2; the shaping filter F3, F4. Colored noise is entering the system.

After identifying the shaping filter, it can be incorporated into the system model, see the following section eq. (5.23). Now, the noise covariances can be found to tune the Kalman filter for this model. The filter and the model can be used for output predictions or a state feedback control.

5.6 Process and measurement noise

In this section, we will discuss whether it is possible to recognize colored process noise and colored measurement noise from the output data. A fully observable system $(\mathbf{A}, \mathbf{g}, \mathbf{c}^T, d)$ with

a noise shaping filter $(\mathbf{A}_F, \mathbf{g}_F, \mathbf{c}_F^T, d_F)$ can be, without loss of generality, defined in the observable canonical form with a single stochastic input

$$\begin{aligned}\begin{bmatrix}\mathbf{f}(t+1)\\ \mathbf{x}(t+1)\end{bmatrix} &= \underbrace{\begin{bmatrix}\mathbf{A}_F & \mathbf{0}\\ \mathbf{g}\mathbf{c}_F^T & \mathbf{A}\end{bmatrix}}_{\mathbf{A}_P}\begin{bmatrix}\mathbf{f}(t)\\ \mathbf{x}(t)\end{bmatrix} + \underbrace{\begin{bmatrix}\mathbf{g}_F\\ \mathbf{g}d_F\end{bmatrix}}_{\mathbf{g}_P} w(t),\\ y_P(t) &= \underbrace{\begin{bmatrix}\mathbf{0} & \mathbf{c}^T\end{bmatrix}}_{\mathbf{c}_P^T}\begin{bmatrix}\mathbf{f}(t)\\ \mathbf{x}(t)\end{bmatrix} + \underbrace{d}_{d_P} w(t).\end{aligned} \tag{5.22}$$

System (5.22) is in the process noise configuration. To model colored measurement noise, the system combined with a shaping filter is given by

$$\begin{aligned}\begin{bmatrix}\mathbf{f}(t+1)\\ \mathbf{x}(t+1)\end{bmatrix} &= \underbrace{\begin{bmatrix}\mathbf{A}_F & \mathbf{0}\\ \mathbf{0} & \mathbf{A}\end{bmatrix}}_{\mathbf{A}_M}\begin{bmatrix}\mathbf{f}(t)\\ \mathbf{x}(t)\end{bmatrix} + \underbrace{\begin{bmatrix}\mathbf{g}_m\\ \mathbf{g}\end{bmatrix}}_{\mathbf{g}_M} w(t),\\ y_M(t) &= \underbrace{\begin{bmatrix}\mathbf{c}_F^T & \mathbf{c}^T\end{bmatrix}}_{\mathbf{c}_M^T}\begin{bmatrix}\mathbf{f}(t)\\ \mathbf{x}(t)\end{bmatrix} + \underbrace{(d+d_F)}_{d_M} w(t).\end{aligned} \tag{5.23}$$

The question is, whether and under what conditions it is possible to find $\mathbf{A}_M, \mathbf{g}_M, d_M$ such that the output spectral density of systems (5.22) and (5.23) is equal.

Theorem: An observable dynamic system (5.22) is entered by colored process noise and system (5.23) is entered by colored measurement noise. It is possible to find $\mathbf{g}_M$, d_M such that the spectral density of both system is equal, iff the characteristic polynomials of the systems have no common factor, i.e. the greatest common divisor (gcd) of them equals 1.

Proof: Let $G(z)$ and $H(z)$ be a transfer functions of the system and the shaping filter of (5.23). Further, let $F(z)$ be a transfer function of (5.22) and let $S_F(z)$ be a spectral factor of $F(z)$, i.e. $S_F(z)$ is a rational function having numerator and denumerator proper and stable and it holds $S_F(z)S_F^*(z^{-1}) = F(z)F^*(z^{-1})$. For system (5.23) it holds that $Y_M(z) = G(z) + H(z)$. Both models (5.22), (5.23) have the same dynamic modes and their characteristic polynomials are equal. The goal is to find such parameters that it holds

$$\frac{g_{\text{num}}(z)}{g_{\text{den}}(z)} + \frac{h_{\text{num}}(z)}{h_{\text{den}}(z)} = \frac{f_{\text{num}}(z)}{f_{\text{den}}(z)}. \tag{5.24}$$

Due to the equality $g_{\text{den}}(z) \cdot h_{\text{den}}(z) = f_{\text{den}}(z)$ the problem reduces to solving of a Diophantine equation[7] of the form

$$g_{\text{num}}(z)h_{\text{den}}(z) + h_{\text{num}}(z)g_{\text{den}}(z) = f_{\text{num}}(z), \tag{5.25}$$

with unknowns $g_{\text{num}}(z), h_{\text{num}}(z)$. Equation (5.25) has a solution iff $\gcd(h_{\text{den}}(z), f_{\text{den}}(z))$ divides $f_{\text{num}}(z)$, (Kučera, 1991). However, any common factor of $h_{\text{den}}(z), f_{\text{den}}(z)$ would be canceled on the left side resulting to a violation of the denominators equality. Thus, the only acceptable common divisor is 1 and in such case, the equation (5.25) is always solvable. The theorem is proved. Δ

The theorem was formulated only for a SISO system. A MIMO case would be significantly more difficult leading to a matrix equation. It can be expected that, under some conditions, this theorem holds also for MIMO system. However, the more complicated case were not analyzed for this thesis.

A corollary of the theorem is that the colored process noise and the colored measurement noise cannot be distinguished from the output data. Transforming the model entered by a colored process noise to the model entered by colored measurement noise can reduce the number of unknown parameters and simplify the identification.

A question arises, how the quality of state estimates changes, if a colored process noise model is replaced by a colored measurement noise model. We can intuitively expect, that the trajectories of state estimates will differ from the optimal ones. This situation will be demonstrated in the following section.

[7] Diophantine equations and spectral factorization can be solved by *Polynomial Toolbox* for Matlab [www.polyx.com].

5.7 Numerical examples

Consider a data generator (5.26) having two stochastic inputs (process and measurement noise). The noise shaping filter of the data generator generates a dumped sinus signal with a period 46 s that enters the system as process noise. Consider further, that only the deterministic part of the system is known.

$$\begin{aligned}\begin{bmatrix}\mathbf{f}(t+1)\\ \mathbf{x}(t+1)\end{bmatrix} &= \underbrace{\begin{bmatrix}1.979 & 1 & & \\ -0.998 & 0 & \multicolumn{2}{c}{\mathbf{0}} \\ 0.008 & 0 & 1.94 & 1 \\ -0.0724 & 0 & -0.94 & 0\end{bmatrix}}_{\mathbf{A}_P}\begin{bmatrix}\mathbf{f}(t)\\ \mathbf{x}(t)\end{bmatrix} + \underbrace{\begin{bmatrix}0.04\\ 0.04\\ 0\\ 0\end{bmatrix}}_{\mathbf{g}_P} w(t),\\ y_P(t) &= \underbrace{\begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix}}_{\mathbf{c}_P^T}\begin{bmatrix}\mathbf{f}(t)\\ \mathbf{x}(t)\end{bmatrix} + e(t),\end{aligned} \tag{5.26}$$

where $w(t) \sim \mathcal{N}(0, 0.001)$ and $e(t) \sim \mathcal{N}(0, 0.05)$.

As the next step, the entering colored noise is detected using the approach from the previous sections. However, as it is not known, if the system is in colored process noise or colored measurement noise configuration, we choose the second alternative, which is simpler. We use the spectral density and the mean square method to find the noise shaping filter that best fits the spectral density of output data. Then, we compare the quality of output predictions with and without using the identified noise shaping filter.

$$\begin{aligned}\begin{bmatrix}\mathbf{f}(t+1)\\ \mathbf{x}(t+1)\end{bmatrix} &= \underbrace{\begin{bmatrix}1.979 & 1 & & \\ -0.998 & 0 & \multicolumn{2}{c}{\mathbf{0}} \\ & & 1.94 & 1 \\ \multicolumn{2}{c}{\mathbf{0}} & -0.94 & 0\end{bmatrix}}_{\mathbf{A}_M}\begin{bmatrix}\mathbf{f}(t)\\ \mathbf{x}(t)\end{bmatrix} + \underbrace{\begin{bmatrix}-1.604\\ 1.348\\ 1.619\\ -1.273\end{bmatrix}}_{\mathbf{g}_M} w(t),\\ y_M(t) &= \underbrace{\begin{bmatrix}1 & 0 & 1 & 0\end{bmatrix}}_{\mathbf{c}_M^T}\begin{bmatrix}\mathbf{f}(t)\\ \mathbf{x}(t)\end{bmatrix} + \underbrace{0.0029}_{d_M} \cdot w(t).\end{aligned} \tag{5.27}$$

Optimal Kalman gains for systems (5.26) and (5.27) are obviously different. They can be calculated using noise covariances estimated from the measured data, Chapter 4. A quality of the

output predictions can be compared in Tab. 11 using 3,000 samples of data. In the first example (first row), the system (5.27) is used as a predictor. In the second example (second row), only the deterministic model is used and the color property of the entering noise is neglected.

Tab. 11 – Comparison of the output prediction quality between the models with and without the noise shaping filter. Standard deviation of the output prediction error was calculated for several prediction horizons (in samples) using 30,000 output values.

Horizon (samples)	1	2	4	8	16
No shaping filter $\sigma\{y-\hat{y}\}$	1.45	1.22	2.96	5.32	9.16
With shaping filter $\sigma\{y-\hat{y}\}$	0.27	0.28	0.32	0.42	0.70
Improvement %	81.5	77.2	89.4	92.2	92.4

It can be seen, that the improvement of the prediction quality is significant especially for higher prediction horizons; demonstrated in Figure 12.

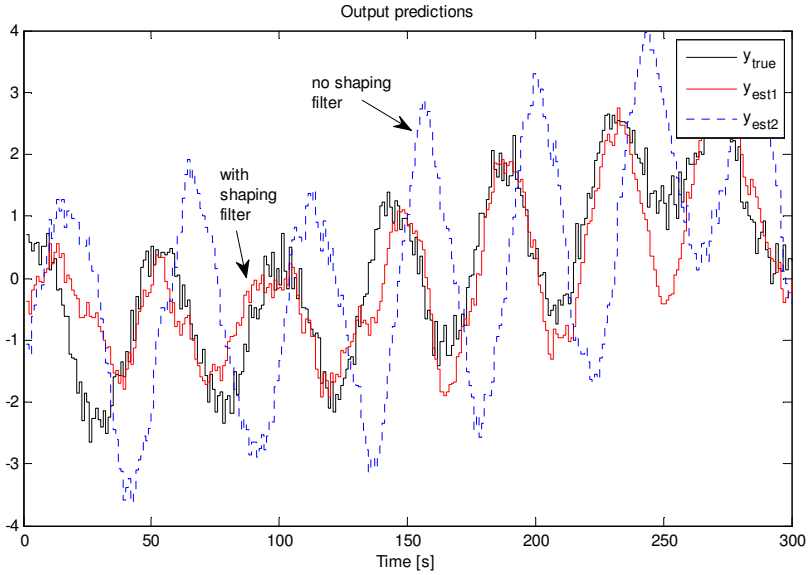

Figure 12 – Comparison of output predictions for predicting horizon 16 s. y_{true} is the measured output, y_{est1} is an output prediction of the system with the shaping filter and y_{est2} is a prediction with no shaping filter used. The data generating system and the shaping filter is in a colored process noise configuration.

A significant improvement can be seen in Tab. 11 and Figure 12. This would be an expected result as the systems (5.26) and (5.27) have the same (or similar, depending on identification precision) spectral density. However, our main interest is in the state estimates and their quality. We will briefly numerically demonstrate, that the selection of noise shaping filter configuration affects the quality of state estimates. Consider a system in the innovation form given by the matrices

$$\begin{aligned} \mathbf{A}_1 &= \begin{bmatrix} 0.9 & -0.7 \\ 0.2 & 0.8 \end{bmatrix}, \ \mathbf{g}_1 = \begin{bmatrix} 3 \\ 0.5 \end{bmatrix}, \\ \mathbf{c}_1^T &= \begin{bmatrix} 0.2 & 0.1 \end{bmatrix}. \end{aligned} \tag{5.28}$$

This system is excited by colored process noise obtained by filtering a white noise using the noise shaping filter

$$\begin{aligned} \mathbf{A}_F &= \begin{bmatrix} 0.6 & 1 \\ -0.86 & 0 \end{bmatrix}, \ \mathbf{g}_F = \begin{bmatrix} 0 \\ 1.38 \end{bmatrix}, \\ \mathbf{c}_F^T &= \begin{bmatrix} 0.1 & 0.4 \end{bmatrix}, d_F = 1. \end{aligned} \tag{5.29}$$

The overall system has the following form. Process (5.28) was transferred to the canonical form (assumed to be identified from the data). Vector **g** was find such that the modelled output spectral density fits the spectral density of the measured output.

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0.6 & 1 & 0 & 0 \\ -0.86 & 0 & 0 & 0 \\ 0.65 & 0 & 1.7 & 1 \\ -0.53 & 0 & -0.86 & 0 \end{bmatrix}, \mathbf{g} = \begin{bmatrix} 0.552 \\ 0.138 \\ 0.650 \\ -0.535 \end{bmatrix}, \\ \mathbf{c}^T &= \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}, \mathbf{d} = 1 \end{aligned} \tag{5.30}$$

having a single stochastic input $e(t) \sim \mathcal{N}(0, 0.05)$.

The system (5.28) with the noise shaping filter (5.29) generates an output data. We assume, that the system is known and the noise shaping filter has been identified by the approach described earlier in this chapter. Next, we analyze the quality of states estimates considering three different scenarios

1) the noise shaping filter is omitted, i.e. it is assumed that the system (5.28) is entered by a white noise,
2) color measurement noise configuration is considered,
3) color process noise configuration is considered.

In Figure 12, we compared output predictions and how the prediction quality is influenced when the noise shaping filter is not omitted. In this simulation, directly the state predictions are compared to the actual trajectories using a Kalman filter. The Kalman filter is tuned by the Bayesian and Monte Carlo method described in Chapter 4. It is obvious the the Kalman filters will be different for each of the three cases and thus different noise covariances will be found by the Bayesian algorithm.

The simulation results are as follows. The numbers show the trace of the covariance matrix of state prediction error. Two cases are considered for the simulation. First, Tab. 12, the data generator was excited by colored process noise. Next, Tab. 13, shows the results of the other case, when the data generator was excited by colored measurement noise.

Tab. 12 – A comparison of the state estimation quality for different Kalman filters. The data generating system is excited by colored process noise.

Kalman filter	trace of state prediction error covariance matrix
Colored process noise configuration	0.044
Colored measurement noise configuration	0.063
White noise assumption	0.12

Tab. 13 – A comparison of the state estimation quality for different Kalman filters. The data generating system is excited by colored measurement noise.

Kalman filter	trace of state prediction error covariance matrix
Colored process noise configuration	0.046
Colored measurement noise configuration	0.035
White noise assumption	0.072

It can be seen, that if the noise shaping filter is omitted (i.e. a white noise assumption, 3rd row), the quality of the state estimates decreases. This is caused by the fact, that the dynamic

modes presented in the noise shaping filter are missing in the Kalman filter. Further, it can be observed, that using the colored measurement noise configuration, the Kalman filter generates state estimates with a higher state prediction error than the optimal case. We have shown before, that it is not possible to recognize a process and a measurement noise configuration from the output data. However, the state estimation quality can differ in both cases. This might be intuitively expected as the Kalman filter is different from the actual process and thus the state trajectories differ either.

Comparing the results in Tab. 12 and Tab. 13, it can be seen, that omitting a noise shaping filter in case of colored process noise leads to less accurate predictions than when the color property of the measurement noise is neglected. This can intuitively be expected, because process noise affects not only the current state, but also all the following, while the measurement noise affects only the current output. We stated at the beginning of this chapter, that assuming white measurement noise is a reasonable assumption.

Further, we will demonstrate effect of the colored process noise configuration and the colored measurement noise configuration. Consider system (5.28) with the noise shaping filter (5.29) and calculate the transfer functions between measured output $y(t)$ and state estimates $\hat{x}_1, \hat{x}_2$ calculated by a Kalman filter. The amplitude frequency characteristics in Figure 13 show the influence of the configuration of a noise shaping filter.

It can be observed that if colored process noise enters the system (blue), the frequency response increases on the frequencies amplified by the noise shaping filter. There is an analogy with a higher covariance *Q* for particular frequencies. However, when the process is entered by colored measurement noise (green), the Kalman filter applies lower weight on the measurements on the frequencies which are amplified by the shaping filter. The Kalman filter applies lower weight on the measurements. This is an analogy of a higher covariance *R*.

5.8 Discussion and conclusions about the colored noise

This chapter introduced approaches for detection of the colored noise entering a linear stochastic system. The time and the frequency domain analysis were employed to decide whether the given system is able to generate the measured spectrum or not. If the output spectral density contains modes other than those generated by the system, the entering noise is not white or

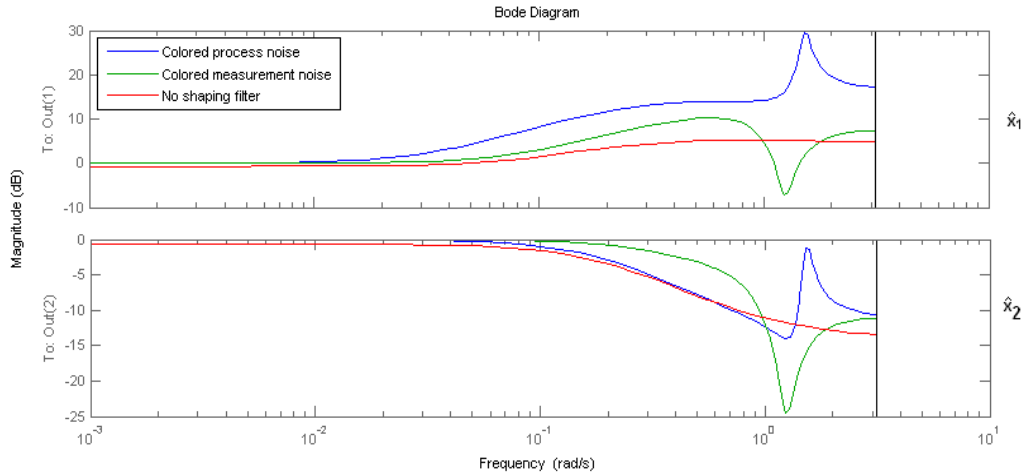

Figure 13 – Comparison of Bode responses for transfer functions $\frac{\hat{X}_1(z)}{Y(z)}, \frac{\hat{X}_2(z)}{Y(z)}$, i.e. between the measured output and the state estimates $\hat{x}_1, \hat{x}_2$ for three different configurations of the noise shaping filter.

there is a significant error in the system identification. The general optimization criterion was defined to find the best fitting shaping filter and the input gain. Solving of this problem leads to difficult non–convex optimization. We have shown how to find the appropriate noise shaping filter by hand in the frequency domain. The system model together with the shaping filter can be further used as a state/output predictor.

We have further shown, that if the spectral density of the output data is inspected, colored measurement noise and colored process noise cannot be distinguished. There can be even a combination of both cases. We have proved, that a model with colored process noise can be transformed to a model with colored measurement noise leading to the same output spectral density. It is possible to identify the noise shaping filter, however, it is not easy to decide if the colored noise is entering the system as process noise or as measurement noise. Bode characteristics in Figure 13 demonstrate the difference between the two configurations. It can be seen, that exchange of the colored process and measurement noise configuration leads to different trajectories of the filter's state estimates.

Using a numerical example, we have demonstrated a significant improvement in the output prediction quality if the noise shaping filter is incorporated into the predictor. The improvement of the prediction performance depends on the noise character and its intensity.

6. Cramér–Rao bounds for estimation of the noise covariances

6.1 Introduction to Cramér–Rao bounds

The Cramér–Rao (CR) bounds represent a limitation of the estimation quality of unknown parameters from the given data. If the variance of the estimates reaches the CR bounds, it can be stated that the estimation algorithm works optimally. In this chapter[8], we will derive the CR bounds for the estimation of noise covariances of a linear stochastic system. The noise covariances are tuning parameters of the Kalman filter, and the filter performance depends on them. We will compare the Bayesian method and the ALS method to the CR bounds and their performance will be discussed. Deriving of noise covariance the CR bounds is based on Tichavský *et al*. (1998) and Šimandl *et al*. (2001). Additional information about CR bounds and Bayesian estimation can be found in Candy (2009).

6.2 The Cramér–Rao bounds

The task is to estimate a vector of parameters $\boldsymbol{\theta} = \left[\theta_1, \theta_2, ..., \theta_m\right]^T$ from a set of measured data $\mathbf{y}^N = \left[\mathbf{y}_1^T, \mathbf{y}_2^T, ..., \mathbf{y}_N^T\right]^T$. The upper index N emphasizes that vector $\mathbf{y}^N$ contains N data samples. The conditioned probability density function (pdf) $p(\mathbf{y}^N \mid \boldsymbol{\theta})$ is assumed to be known. Further, the *nabla* operator is defined as

$$\nabla_{\boldsymbol{\theta}} = \left[\frac{\partial}{\partial\theta_1}, \frac{\partial}{\partial\theta_2}, ..., \frac{\partial}{\partial\theta_m}\right]. \tag{6.1}$$

[8] This paper was originally published in IFAC–PapersOnLine, www.ifac–papersonline.net, and is reproduced here with permission from IFAC, Matisko and Havlena (2011).

The Fisher information matrix (FIM) $\mathbf{J}(\boldsymbol{\theta})$ for parameters $\boldsymbol{\theta}$ is defined as

$$\mathbf{J}(\boldsymbol{\theta}) = -\mathcal{E}_{\mathbf{y}}\left\{\nabla_{\boldsymbol{\theta}}\left[\nabla_{\boldsymbol{\theta}} \ln p\left(\mathbf{y}^N \mid \boldsymbol{\theta}\right)\right]^T\right\}, \tag{6.2}$$

where operator $\mathcal{E}_{\mathbf{y}}\left\{\cdot \mid \cdot\right\}$ is the conditioned expected value defined as

$$\mathcal{E}_{\mathbf{y}}\left\{f(\mathbf{y},\boldsymbol{\theta}) \mid \boldsymbol{\theta}\right\} = \int_{-\infty}^{\infty} f(\mathbf{y},\boldsymbol{\theta}) p(\mathbf{y} \mid \boldsymbol{\theta}) d\mathbf{y}. \tag{6.3}$$

Alternatively, we can define a posterior FIM (and posterior CR bounds) using joint probability $p(\mathbf{y}^N, \boldsymbol{\theta})$ instead of the conditioned one, where $p(\mathbf{y}^N, \boldsymbol{\theta}) = p(\mathbf{y}^N \mid \boldsymbol{\theta}) p\left(\boldsymbol{\theta}\right)$. Probability function $p\left(\boldsymbol{\theta}\right)$ represents prior information about the parameter vector $\boldsymbol{\theta}$.

Assume having an unbiased estimate of the parameter vector $\hat{\boldsymbol{\theta}}$ obtained by an arbitrary method. When the true value of the parameter vector is compared to the unbiased estimated set, the following inequality holds, Papoulis (1991).

$$\mathcal{E}_{\boldsymbol{\theta}}\left\{\left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\right)\left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\right)^T\right\} \geq \mathbf{J}^{-1}\left(\boldsymbol{\theta}\right), \tag{6.4}$$

where $\mathbf{J}^{-1}\left(\boldsymbol{\theta}\right)$ represents the Cramér–Rao bound. Reminding, inequality $\mathbf{A} \geq \mathbf{B}$ means that $\mathbf{A} - \mathbf{B}$ is a positive semi definite matrix. Based on (6.4), the following statement can be concluded. Assuming that the CR bound is reachable, any estimation algorithm working optimally (in the sense of the smallest covariance of obtained estimates), must give estimates whose variance is equal to the Cramér–Rao bound. If the CR bound is reachable, then the optimal result can be achieved by a maximum likelihood approach.

Consider a scalar linear stochastic system

$$\begin{aligned} x(t+1) &= ax(t) + v(t), \\ y(t) &= x(t) + e(t), \end{aligned} \tag{6.5}$$

where $x(t)$ and $y(t)$ represent the state and the measured output, respectively. Stochastic inputs $v(t)$ and $e(t)$ are described by normal distributions

$$v(t) \sim \mathcal{N}\left(0, Q\right),\ e(t) \sim \mathcal{N}\left(0, R\right). \tag{6.6}$$

The sets $\mathcal{X}^t, \mathcal{Y}^t$ represent the data up to the time *t*, i.e. $\mathcal{Y}^t = \left\{\mathbf{y}(0), \mathbf{y}(1), \ldots, \mathbf{y}(t)\right\}$ and $\mathcal{X}^t = \left\{\mathbf{x}(0), \mathbf{x}(1), \ldots, \mathbf{x}(t)\right\}$. The logarithm of the conditioned probability of the state, measurements and unknown parameters $\boldsymbol{\theta} = \left[Q, R\right]^T$ can be expressed as

$$\ln p\left(\mathcal{X}^t, \mathcal{Y}^t \mid \boldsymbol{\theta}\right) = \sum_{i=0}^{t} \ln p\left(\mathbf{y}_i \mid \mathbf{x}_i, \boldsymbol{\theta}\right) + \sum_{i=0}^{t-1} \ln p\left(\mathbf{x}_{i+1} \mid \mathbf{x}_i, \boldsymbol{\theta}\right) + \ln p\left(\mathbf{x}_0\right). \tag{6.7}$$

The following matrices are used for FIM calculation

$$\mathbf{K}_{t+1}^{t} = \mathcal{E}_{x_{t+1}}\left\{-\nabla_{\mathbf{x}_t}\left[\nabla_{\mathbf{x}_t} \ln p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t\right)\right]^T \middle| \mathbf{x}_t\right\}, \tag{6.8}$$

$$\mathbf{K}_{t+1}^{t,t+1} = \mathcal{E}_{x_{t+1}}\left\{-\nabla_{\mathbf{x}_{t+1}}\left[\nabla_{\mathbf{x}_t} \ln p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t\right)\right]^T \middle| \mathbf{x}_t\right\}, \tag{6.9}$$

$$\mathbf{K}_{t+1}^{t+1} = \mathcal{E}_{x_{t+1}}\left\{-\nabla_{\mathbf{x}_{t+1}}\left[\nabla_{\mathbf{x}_{t+1}} \ln p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t\right)\right]^T \middle| \mathbf{x}_t\right\}, \tag{6.10}$$

$$\mathbf{K}_{t+1}^{\theta,t} = \mathcal{E}_{x_{t+1}}\left\{-\nabla_{\mathbf{x}_t}\left[\nabla_{\boldsymbol{\theta}} \ln p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \boldsymbol{\theta}\right)\right]^T \middle| \mathbf{x}_t, \boldsymbol{\theta}\right\}, \tag{6.11}$$

$$\mathbf{K}_{t+1}^{\theta,t+1} = \mathcal{E}_{x_{t+1}}\left\{-\nabla_{\mathbf{x}_{t+1}}\left[\nabla_{\boldsymbol{\theta}} \ln p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \boldsymbol{\theta}\right)\right]^T \middle| \mathbf{x}_t, \boldsymbol{\theta}\right\}, \tag{6.12}$$

$$\mathbf{K}_{t+1}^{\theta} = \mathcal{E}_{x_{t+1}}\left\{-\nabla_{\boldsymbol{\theta}}\left[\nabla_{\boldsymbol{\theta}} \ln p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \boldsymbol{\theta}\right)\right]^T \middle| \mathbf{x}_t, \boldsymbol{\theta}\right\}, \tag{6.13}$$

$$\mathbf{L}_{t}^{t} = \mathcal{E}_{y_t}\left\{-\nabla_{\mathbf{x}_t}\left[\nabla_{\mathbf{x}_t} \ln p\left(\mathbf{y}_t \mid \mathbf{x}_t\right)\right]^T \middle| \mathbf{x}_t\right\}, \tag{6.14}$$

$$\mathbf{L}_{t}^{\theta,t} = \mathcal{E}_{y_t}\left\{-\nabla_{\mathbf{x}_t}\left[\nabla_{\boldsymbol{\theta}} \ln p\left(\mathbf{y}_t \mid \mathbf{x}_t, \boldsymbol{\theta}\right)\right]^T \middle| \mathbf{x}_t, \boldsymbol{\theta}\right\}, \tag{6.15}$$

$$\mathbf{L}_{t}^{\theta} = \mathcal{E}_{y_t}\left\{-\nabla_{\boldsymbol{\theta}}\left[\nabla_{\boldsymbol{\theta}} \ln p\left(\mathbf{y}_t \mid \mathbf{x}_t, \boldsymbol{\theta}\right)\right]^T \middle| \mathbf{x}_t, \boldsymbol{\theta}\right\}, \tag{6.16}$$

Further, recursive formulas for FIM of the state and unknown parameters are defined as

$$\begin{aligned}
\mathbf{J}^{\theta,\theta}_{t+1|t} &= \mathbf{J}^{\theta,\theta}_{t|t-1} + \mathbf{L}^{\theta}_{t} + \mathbf{K}^{\theta}_{t+1} - \Delta^{\theta,x}_{tt}\Delta^{-1}_{tt}\Delta^{x,\theta}_{tt}, \\
\mathbf{J}^{x,\theta}_{t+1|t} &= \mathbf{K}^{t+1,\theta}_{t+1} - \mathbf{K}^{t+1,t}_{t+1}\Delta^{-1}_{tt}\Delta^{x,\theta}_{tt}, \\
\mathbf{J}^{x,x}_{t+1|t} &= \mathbf{K}^{t+1}_{t+1} - \mathbf{K}^{t+1,t}_{t+1}\Delta^{-1}_{tt}\mathbf{K}^{t,t+1}_{t+1}, \\
\mathbf{J}^{\theta,x}_{t+1|t} &= \left(\mathbf{J}^{x,\theta}_{t+1|t}\right)^{T}
\end{aligned} \tag{6.17}$$

where

$$\Delta_{tt} = \mathbf{J}^{x,x}_{t|t} + \mathbf{K}^{t}_{t+1}, \quad \Delta^{\theta,x}_{tt} = \mathbf{J}^{\theta,x}_{t|t} + \mathbf{K}^{\theta,t}_{t+1}, \tag{6.18}$$

$$\Delta^{x,\theta}_{tt} = \left[\Delta^{\theta,x}_{tt}\right]^{T}, \quad \mathbf{K}^{t+1,t}_{t+1} = \left[\mathbf{K}^{t,t+1}_{t+1}\right]^{T}, \tag{6.19}$$

and

$$\mathbf{J}^{x,\theta}_{t|t} = \mathbf{J}^{x,\theta}_{t|t-1} + \mathbf{L}^{t,\theta}_{t}, \quad \mathbf{J}^{x,x}_{t|t} = \mathbf{J}^{x,x}_{t|t-1} + \mathbf{L}^{t}_{t}. \tag{6.20}$$

Initial conditions of the recursive algorithm are set as

$$\begin{aligned}
\mathbf{J}^{x,\theta}_{0|-1} &= \mathbf{K}^{0,\theta}_{0} = 0, \\
\mathbf{J}^{x,x}_{0|-1} &= \mathbf{K}^{0}_{0} = \mathcal{E}_{\mathbf{x}_0}\left\{-\nabla_{\mathbf{x}_0}\left[\nabla_{\mathbf{x}_0}\ln p\left(\mathbf{x}_0\right)\right]^{T}\right\}, \\
\mathbf{J}^{\theta,\theta}_{0|-1} &= \mathbf{A}_{0} = \mathcal{E}_{\boldsymbol{\theta}}\left\{-\nabla_{\boldsymbol{\theta}}\left[\nabla_{\boldsymbol{\theta}}\ln p\left(\boldsymbol{\theta}\right)\right]^{T}\right\},
\end{aligned} \tag{6.21}$$

where $\mathbf{K}^{0}_{0}$ and $\mathbf{A}_{0}$ represent prior information. If no prior probability function is known, then $\mathbf{A}_{0} = 0$, $\mathbf{K}^{0}_{0} = 0$.

The final form of FIM of state and parameters is

$$\mathbf{J}_{t|t} = \begin{bmatrix} \mathbf{J}^{x,x}_{t|t} & \mathbf{J}^{x,\theta}_{t|t} \\ \mathbf{J}^{\theta,x}_{t|t} & \mathbf{J}^{\theta,\theta}_{t|t} \end{bmatrix}. \tag{6.22}$$

6.3 The Cramér–Rao bounds for noise covariance estimation

Consider a stable scalar stochastic system given by (6.5). The noise sources are not correlated and are defined by (6.6). The probability distributions used in (6.7) can be defined as

$$\begin{aligned} p\left(x_{t+1} \mid x_t, Q\right) &= \mathcal{N}\left(ax_t, Q\right), \\ p\left(y_t \mid x_t, R\right) &= \mathcal{N}\left(x_t, R\right). \end{aligned} \tag{6.23}$$

Now, matrices (6.8)–(6.16) can be calculated using the following general integral formulas, assuming $\alpha > 0$

$$\begin{aligned} &\int_{-\infty}^{\infty} e^{-\alpha x^2} dx = \sqrt{\frac{\pi}{\alpha}}, \\ &\int_{-\infty}^{\infty} x e^{-\alpha(x-\mu)^2} dx = \mu\sqrt{\frac{\pi}{\alpha}}, \\ &\int_{-\infty}^{\infty} (x-\mu) e^{-\alpha(x-\mu)^2} dx = 0, \\ &\int_{-\infty}^{\infty} \left(x-\mu\right)^2 e^{-\alpha\left(x-\mu\right)^2} dx = \sqrt{\frac{\pi}{4\alpha^3}}. \end{aligned} \tag{6.24}$$

Further, logarithm of the Gaussian function is given by

$$\ln \mathcal{N}(\mu, P) = -\frac{1}{2}\ln 2\pi - \frac{1}{2}\ln P - \frac{1}{2P}\left(x-\mu\right)^2. \tag{6.25}$$

The CR bounds of *Q*, *R* estimates are to be found, considering a scalar system with two unknown parameters. The ∇ (nabla) operator is in this case of the form

$$\nabla_{\boldsymbol{\theta}} = \left[\frac{\partial}{\partial Q}, \frac{\partial}{\partial R}\right], \tag{6.26}$$

where $\boldsymbol{\theta} = \left[Q,\ R\right]^T$. At the first step, the arguments of the expectation operators in (6.11)–(6.16) are calculated as follows

$$\nabla_{\mathbf{x}_t}\left[\nabla_{\boldsymbol{\theta}} \ln p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \boldsymbol{\theta}\right)\right]^T = \begin{bmatrix} -\frac{a}{Q^2}\left(x_{t+1} - ax_t\right) \\ 0 \end{bmatrix}, \tag{6.27}$$

$$\nabla_{\mathbf{x}_{t+1}}\left[\nabla_{\boldsymbol{\theta}} \ln p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \boldsymbol{\theta}\right)\right]^T = \begin{bmatrix} \frac{1}{Q^2}\left(x_{t+1} - ax_t\right) \\ 0 \end{bmatrix}, \tag{6.28}$$

$$\nabla_{\boldsymbol{\theta}}\left[\nabla_{\boldsymbol{\theta}} \ln p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \boldsymbol{\theta}\right)\right]^T = \begin{bmatrix} \frac{1}{2Q^2} - \frac{1}{Q^3}\left(x_{t+1} - ax_t\right)^2 & 0 \\ 0 & 0 \end{bmatrix}, \tag{6.29}$$

$$\nabla_{\boldsymbol{\theta}}\left[\nabla_{\boldsymbol{\theta}} \ln p\left(\mathbf{y} \mid \mathbf{x}_t, \boldsymbol{\theta}\right)\right]^T = \begin{bmatrix} 0 & 0 \\ 0 & \dfrac{1}{2R^2} - \dfrac{1}{R^3}\left(y_t - x_t\right)^2 \end{bmatrix}. \tag{6.30}$$

Next, matrices (6.11)–(6.16) are obtained using integral formulas (6.24) as

$$\mathbf{K}_{t+1}^{\theta,t} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{K}_{t+1}^{\theta,t+1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{K}_{t+1}^{\theta} = \begin{bmatrix} \dfrac{1}{2Q^2} & 0 \\ 0 & 0 \end{bmatrix}, \tag{6.31}$$

$$\mathbf{L}_{t}^{\theta,t} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{L}_{t}^{\theta} = \begin{bmatrix} 0 & 0 \\ 0 & \dfrac{1}{2R^2} \end{bmatrix}, \tag{6.32}$$

$$A_0 = \mathcal{E}_{\boldsymbol{\theta}}\left\{-\nabla_{\boldsymbol{\theta}}\left[\nabla_{\boldsymbol{\theta}} \ln p\left(\boldsymbol{\theta}\right)\right]^T\right\}. \tag{6.33}$$

The term $\mathbf{K}_{t+1}^{\theta,t}$, which is further used to calculate $\Delta_{tt}^{\theta,x}$ in (6.18) is zero. This allows us to omit calculation of the terms (6.8)–(6.10) only used to calculate the term Δ_{tt}.

It can be seen from (6.31)–(6.32), that several of the matrices are zero which significantly simplifies formula (6.17). In particular matrix $\mathbf{J}_{t|t}^{\theta,x}$ in (6.22) is zero and this fact allows us to calculate FIM for unknown covariances independently on the states. The resulting FIM can be written as

$$\mathbf{J}_{t+1|t}^{\theta,\theta} = \mathbf{J}_{t|t-1}^{\theta,\theta} + \mathbf{L}_{t}^{\theta} + \mathbf{K}_{t+1}^{\theta}, \tag{6.34}$$

where $\mathbf{J}_{0|-1}^{\theta,\theta}$ represents the prior information about unknown parameters Q, R. If no prior information is available, it can be set to zero, $\mathbf{J}_{0|-1}^{\theta,\theta} = 0$. From the above equations and the form of matrices in (6.31)–(6.32), it can be seen that the FIM is calculated for each covariance independently. Formula for the FIM is of the form

$$\mathbf{J}_{t+1|t}^{\theta,\theta} = \mathbf{J}_{t|t-1}^{\theta,\theta} + \begin{bmatrix} 0 & 0 \\ 0 & \dfrac{1}{2R^2} \end{bmatrix} + \begin{bmatrix} \dfrac{1}{2Q^2} & 0 \\ 0 & 0 \end{bmatrix}. \tag{6.35}$$

A particular closed form solution to recursive formula (6.35) is given by

$$\mathbf{J}_{t|t-1}^{\theta,\theta} = \begin{bmatrix} \dfrac{t}{2Q^2} & 0 \\ 0 & \dfrac{t}{2R^2} \end{bmatrix}. \tag{6.36}$$

6.4 The Cramér–Rao bounds for a system in an innovation form

Consider a scalar dynamic system in an innovation form

$$\begin{aligned} x_{t+1} &= ax_t + k\varepsilon_t, \\ y_t &= x_t + \varepsilon_t, \end{aligned} \tag{6.37}$$

where k is a scalar steady state Kalman gain and $\varepsilon(t)$ is the stationary innovation process. Then

$$\varepsilon(t) \sim \mathcal{N}\left(0, R_\varepsilon\right). \tag{6.38}$$

Cramér–Rao bounds for k, R_ε estimation are derived using the same formulas as for system (6.5). Analogously to formulas (6.27)–(6.30) we define

$$\nabla_{\mathbf{x}_t}\left[\nabla_{\boldsymbol{\theta}} \ln p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \boldsymbol{\theta}\right)\right]^T = \begin{bmatrix} -\dfrac{a}{k^3 R_\varepsilon}\left(x_{t+1} - ax_t\right) \\ -\dfrac{a}{k^2 R_\varepsilon^2}\left(x_{t+1} - ax_t\right) \end{bmatrix}, \tag{6.39}$$

$$\nabla_{\mathbf{x}_{t+1}}\left[\nabla_{\boldsymbol{\theta}} \ln p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \boldsymbol{\theta}\right)\right]^T = \begin{bmatrix} \dfrac{1}{k^2 R_\varepsilon}\left(x_{t+1} - ax_t\right) \\ \dfrac{1}{k^2 R_\varepsilon^2}\left(x_{t+1} - ax_t\right) \end{bmatrix}, \tag{6.40}$$

$$\nabla_{\boldsymbol{\theta}}\left[\nabla_{\boldsymbol{\theta}} \ln p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \boldsymbol{\theta}\right)\right]^T = \begin{bmatrix} \dfrac{1}{2k^2} - \dfrac{1}{R_\varepsilon k^4}\left(x_{t+1} - ax_t\right)^2 & -\dfrac{1}{R_\varepsilon^2 k^3}\left(x_{t+1} - ax_t\right)^2 \\ -\dfrac{1}{R_\varepsilon^2 k^3}\left(x_{t+1} - ax_t\right)^2 & \dfrac{1}{2R_\varepsilon^2} - \dfrac{1}{R_\varepsilon^3 k^2}\left(x_{t+1} - ax_t\right)^2 \end{bmatrix} \tag{6.41}$$

$$\nabla_{\boldsymbol{\theta}}\left[\nabla_{\boldsymbol{\theta}}\ln p\left(\mathbf{y}\mid\mathbf{x}_t,\boldsymbol{\theta}\right)\right]^T=\begin{bmatrix}0 & 0\\ 0 & \dfrac{1}{2R_\varepsilon^2}-\dfrac{1}{R_\varepsilon^3}\left(y_t-x_t\right)^2\end{bmatrix}. \tag{6.42}$$

The expected values are calculated using integral formulas (6.24). Matrices (6.11)–(6.16) are of the form

$$\mathbf{K}_{t+1}^{\theta,t}=\begin{bmatrix}0\\0\end{bmatrix},\ \mathbf{K}_{t+1}^{\theta,t+1}=\begin{bmatrix}0\\0\end{bmatrix},\ \mathbf{K}_{t+1}^{\theta}=\begin{bmatrix}\dfrac{2}{k^2} & \dfrac{1}{kR_\varepsilon}\\ \dfrac{1}{kR_\varepsilon} & \dfrac{1}{2R_\varepsilon^2}\end{bmatrix} \tag{6.43}$$

$$\mathbf{L}_t^{\theta,t}=\begin{bmatrix}0\\0\end{bmatrix},\ \mathbf{L}_t^{\theta}=\begin{bmatrix}0 & 0\\ 0 & \dfrac{1}{2R_\varepsilon^2}\end{bmatrix}, \tag{6.44}$$

The formula for FIM calculation is of the form (6.34). Substituting (6.43) and (6.44) to (6.34) leads to the resulting formula

$$\mathbf{J}_t^{\theta,\theta}=\begin{bmatrix}\dfrac{2t}{k^2} & \dfrac{t}{kR_\varepsilon}\\ \dfrac{t}{kR_\varepsilon} & \dfrac{t}{R_\varepsilon^2}\end{bmatrix}. \tag{6.45}$$

In the previous sections, we have considered only SISO systems. The MIMO case is significantly more complicated, because derivatives with respect to matrices need to be used. This would lead to tensors of high dimensions. However, if any new algorithm for the estimation of noise covariances is proposed, it can be tested employing scalar examples and compare the results to the CR bounds. If the algorithm does not work well for a SISO system, it will most likely not work properly for MIMO systems.

6.5 Comparison of the methods for noise covariance estimation

In the previous sections, we derived the CR bounds for noise covariances estimation. In this chapter, a comparison of sALS method and Bayesian method will be presented according to

the CR bounds. The scALS algorithm prior setting contains the optimal Kalman filter gain and the maximum lag for autocorrelation calculation $M = 15$. The Bayesian method (without using the Monte Carlo grid updating) searches the maximum in the interval 0.1 to 10. The grid is logarithmic and contains 80×80 points.

Consider a system of the form (6.5) where $a = -0.5$, $Q = 10$ and $R = 2$. The estimation algorithm uses *k* time samples of data. Each estimating process is repeated 300 times and then the statistics of the repeated estimates are examined; concretely the variance of the *Q*, *R* –estimates. The dependence of the variance of the estimated *Q*–parameters on the number of used data samples is shown in Figure 14.

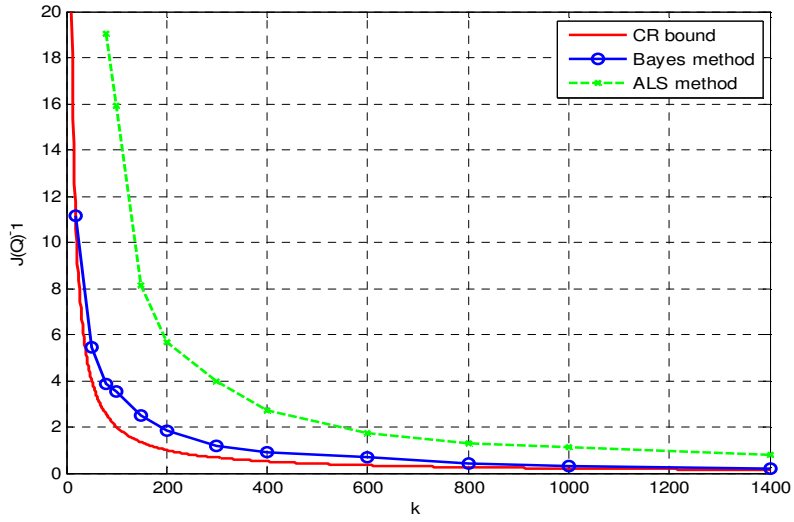

Figure 14 – The Cramér–Rao bound for *Q* estimation and variance of estimated *Q* using scALS and the Bayesian method (Section 4.4).

In Figure 15, an analogous result for *R* estimation is shown. We can state that Bayesian method converges faster to the CR bound than scALS. Bayesian algorithm gives better estimates in case of covariance *Q* than ALS even with very small sets of data (around 50 samples).

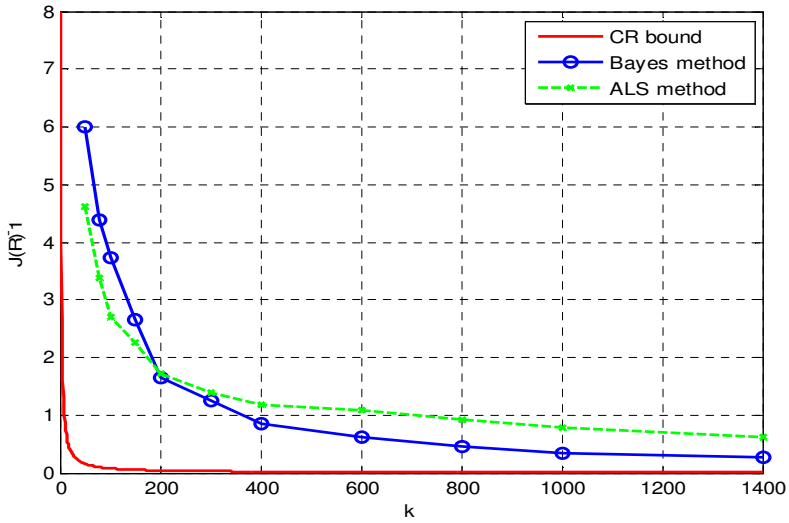

Figure 15 – The Cramér–Rao bound for R estimation and variance of estimated R using scALS and the Bayesian method (Section 4.4).

6.6 Discussion of Cramér–Rao bounds

The Cramér–Rao bounds for the noise covariance estimation were calculated for two different stochastic models. From (6.31)–(6.32), it can be seen that matrices $\mathbf{K}_{t+1}^{\theta,t}$, $\mathbf{K}_{t+1}^{\theta,t+1}$ and $\mathbf{L}_{t}^{\theta,t}$ are zero, that means that the information between state and noise covariances is not correlated. It can be concluded that the covariances and the states can be estimated independently of each other. Another observation can be done on matrices (6.31) and (6.32). It can be seen that there is no correlation between information about the covariances. That means that CR bounds can be calculated for each covariance separately. Considering limit cases, Q converging to zero and R being arbitrary large, the CR bound for Q estimation will converge to zero either. It means that there exists a method that can estimate Q up to an arbitrary level of accuracy. The unsolved question about the CR bounds is, whether it is reachable or not. The simulation, Figure 15, and the fast convergence indicate that the CR bound might be reachable. The variables Q and R can be estimated from the values $y(t)$ and $\tilde{\mathbf{x}}(t)$, if the estimates was obtained by optimally set KF. However, the optimal setting is given by unknown covariances Q, R.

6.7 Relationship between Cramér–Rao bounds and Riccati equation

In this section, we will shortly point out the relationship between the Riccati equation (RE) and the CR bounds. It is well known, that under some conditions, the Kalman filter is optimal. It was stated in the previous section, that the CR bounds represent a limit of estimation quality. Therefore, it can be expected that RE should converge to the CR bounds for state estimation of a linear system. Using formulas (6.8)–(6.10) and (6.14), the CR bounds for state estimation can be easily obtained, Tichavský *et al*. (1998). It holds,

$$\begin{aligned} &\mathbf{K}_{t+1}^{t} = \mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A},\ \mathbf{K}_{t+1}^{t,t+1} = -\mathbf{A}^T\mathbf{Q}^{-1}, \\ &\mathbf{K}_{t+1}^{t+1} = \mathbf{Q}^{-1},\ \mathbf{L}_t^t = \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}. \end{aligned} \tag{6.46}$$

Now, the recursive formula for FIM calculation, Šimandl *et al*. (2001), is of the form

$$\mathbf{J}_{t+1} = \mathbf{K}_{t+1}^{t+1} - \mathbf{K}_{t+1}^{t+1,t}\left(\mathbf{J}_t + \mathbf{K}_{t+1}^{t} + \mathbf{L}_t^t\right)\mathbf{K}_{t+1}^{t,t+1}. \tag{6.47}$$

Using (6.46) we obtain resulting formula

$$\mathbf{J}_{t+1} = \mathbf{Q}^{-1} - \mathbf{Q}^{-1}\mathbf{A}\left(\mathbf{J}_t + \mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\mathbf{A}^T\mathbf{Q}^{-1}. \tag{6.48}$$

The Riccati equation is of the form

$$\mathbf{P}_{t+1} = \mathbf{A}\mathbf{P}_t\mathbf{A}^T - \mathbf{A}\mathbf{P}_t\mathbf{C}^T\left(\mathbf{C}\mathbf{P}_t\mathbf{C}^T + \mathbf{R}\right)^{-1}\mathbf{C}\mathbf{P}_t\mathbf{A}^T + \mathbf{Q}. \tag{6.49}$$

It can be stated, that $\mathbf{P}_t$ and $\mathbf{J}_t^{-1}$ converge to the same matrix. Both equations are equivalent and $\mathbf{P}_t = \mathbf{J}_t^{-1}$ for sufficient large t, not necessarily $t \to \infty$. Moreover, if initial values $\mathbf{P}_0 = \mathbf{J}_0^{-1}$, then trajectories of $\mathbf{P}_t$ and $\mathbf{J}_t^{-1}$ are the same. This fact proves optimality of the Kalman filter.

Consider two recursive equations (6.50) and (6.51) that are the Fisher information matrix recursive equation and the discrete Riccati equation.

$$\mathbf{J}_{t+1} = \mathbf{Q}^{-1} - \mathbf{Q}^{-1}\mathbf{A}\left(\mathbf{J}_t + \mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\mathbf{A}^T\mathbf{Q}^{-1} \tag{6.50}$$

$$\mathbf{P}_{t+1} = \mathbf{A}\mathbf{P}_t\mathbf{A}^T - \mathbf{A}\mathbf{P}_t\mathbf{C}^T\left(\mathbf{C}\mathbf{P}_t\mathbf{C}^T + \mathbf{R}\right)^{-1}\mathbf{C}\mathbf{P}_t\mathbf{A}^T + \mathbf{Q}. \tag{6.51}$$

The right side of the equation (6.50) is modified using substitution $\mathbf{J}_t^{-1} = \mathbf{P}_t$

$$\begin{aligned}
&\mathbf{Q}^{-1} - \mathbf{Q}^{-1}\mathbf{A}\left(\mathbf{J}_t + \mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\mathbf{A}^T\mathbf{Q}^{-1} = \\
&= \mathbf{Q}^{-1}\mathbf{A}\left(\mathbf{A}^{-1}\mathbf{Q}\mathbf{A}^{-T} - \left(\mathbf{P}_t^{-1} + \mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\right)\mathbf{A}^T\mathbf{Q}^{-1} = \\
&= \mathbf{Q}^{-1}\mathbf{A}\left(\mathbf{A}^{-1}\mathbf{Q}\mathbf{A}^{-T}\left(\mathbf{P}_t^{-1} + \mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right) - 1\right)\left(\mathbf{P}_t^{-1} + \mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\mathbf{A}^T\mathbf{Q}^{-1} = \\
&= \mathbf{Q}^{-1}\mathbf{A}\left(\mathbf{A}^{-1}\mathbf{Q}\mathbf{A}^{-T}\left(\mathbf{P}_t^{-1} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)\right)\left(\mathbf{P}_t^{-1} + \mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\mathbf{A}^T\mathbf{Q}^{-1}
\end{aligned}$$

Now, both sides of the equation are inverted, so the left side is $\mathbf{P}_{k+1}$ and the right side is of the form

$$\begin{aligned}
&\mathbf{Q}^{-1}\mathbf{A}\left(\mathbf{A}^{-1}\mathbf{Q}\mathbf{A}^{-T}\left(\mathbf{P}_t^{-1} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)\right)\left(\mathbf{P}_t^{-1} + \mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\mathbf{A}^T\mathbf{Q}^{-1} = \\
&= \mathbf{Q}\mathbf{A}^{-T}\left(\mathbf{P}_t^{-1} + \mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)\left(\left(\mathbf{P}_t^{-1} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A}^1\right)\mathbf{A}^{-1}\mathbf{Q} = \\
&= \mathbf{Q}\mathbf{A}^{-T}\left(\mathbf{P}_t^{-1} + \mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)\left(\left(\mathbf{P}_t^{-1} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\mathbf{A}^T\right) = \\
&= \mathbf{A}\left(\mathbf{P}_t^{-1} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\mathbf{A}^T + \mathbf{Q}
\end{aligned} \tag{6.52}$$

Further, an inversion lemma

$$\left(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}\right)^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1}\mathbf{C}\mathbf{A}^{-1} \tag{6.53}$$

will be applied on the expression $\left(\mathbf{P}_t^{-1} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)^{-1}$. The final formula is of the form

$$\mathbf{P}_{t+1} = \mathbf{A}\mathbf{P}_t\mathbf{A}^T - \mathbf{A}\mathbf{P}_t\mathbf{C}^T\left(\mathbf{C}\mathbf{P}_t\mathbf{C}^T + \mathbf{R}\right)^{-1}\mathbf{C}\mathbf{P}_t\mathbf{A}^T + \mathbf{Q},$$

which is exactly the Riccati equation.

7. Conclusions

The doctoral thesis covered a large part of the stochastic properties estimation for linear dynamic systems. First part discussed optimality tests, that can be used for evaluation of the state estimation quality using the innovation sequence. The whiteness tests are widely used by statisticians, however, they can be hardly found in the control field literature. Technologists and scientists who use the Kalman filter for state predictions usually do not use any quality measure. We have chosen the most significant whiteness property tests and demonstrate their performance with various systems. For this purpose we analyzed and compared several estimators for the autocorrelation function. We have proposed an estimator that generates independently identically distributed values. Further, a quality measure has been proposed together with the decision about optimality. The quality measure can be used to compare the settings of a Kalman filter that are available. It can be also used to detect changes in the noise intensity or structure.

The second part of the thesis described algorithms for the noise covariance estimation. It started with demonstration that the Bayesian principles can extract enough information from the output data for the estimation of noise covariances. Together with Monte Carlo methods, we have proposed an algorithm that is better than previously published methods and works also with MIMO systems. Combining the quality measures and the estimation algorithm we have proposed an adaptive Kalman filter, that can use a newly measured data to update the covariances and evaluate the filter performance.

The third part of the thesis used the knowledge about the noise covariance estimation and the optimality tests. We have shown how to detect that the entering noise is colored. The time and the frequency domain analyses were employed to decide whether the given system is able to generate the output spectral density the same as the measured signal has. We also introduced a simple practical method for finding the shaping filter. It was demonstrated on a numerical example, how the Kalman filter performance can be improved when the appropriate

shaping filter is used and the color property of the entering noise is not neglected. We pointed out the spectrum of the innovation sequence and also the state prediction error.

The last part discussed the limits of the estimation quality for the noise covariances. We have employed the Cramér–Rao bounds and calculate the bounds for this particular problem using the approach proposed by Šimandl *et al*. (2006). We have shown some interesting properties of the Cramér–Rao bounds for this problem. The fact that we have calculated the Cramér–Rao bounds only for a scalar system does not mean any significant disadvantage. Any newly proposed algorithm can be compared to the Cramér–Rao bounds employing the scalar system. It can be expected that if the performance for small system is far from the optimum, the algorithm would not work well for larger systems.

Several of the functions used for calculations throughout the thesis are attached in Appendix A in m–code for Matlab. Most of them are necessary for ALS/scALS method, however, they are ready for a general use.

The doctoral thesis meets all its assignments and goals listed in Chapter 1. The original results have been presented at high impact conferences and journals. The results presented here can be used for practical applications and also as an inspiration for a further research in the field of stochastic system identification.

8. Notation and summary of mathematical operations

Format of mathematical formulas is as follows. Vectors are represented as bold small letters, matrices are bold capital letters and scalars are in italics. Covariances and transfer functions are represented as capital letters. Time *t* is considered to be discrete. Vectors are columns. Brackets [ ] always represent matrices and are not used for other purposes. Braces { } define an argument of the operator; exceptionally can be used for separation of large formulas. The hat symbol ^ over the variable represents an estimate, e.g. the state estimate $\hat{x}$.

8.1 Symbols and abbreviations

$\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D},\mathbf{G}$	system matrices
n	number of the states, system order
p	number of stochastic inputs
r	number of outputs
$\mathbf{x}(t), \mathbf{x}_t \in \mathbb{R}^n$	state of the system
$\mathbf{u}(t) \in \mathbb{R}^m$	deterministic input of the system
$\mathbf{y}(t) \in \mathbb{R}^r$	measured output of the system
$\hat{\mathbf{x}}(t) \in \mathbb{R}^n$	expected value of the state $\mathbf{x}(t)$
$\hat{\mathbf{x}}\left(t \mid \tau\right)$	expected value of the state at time *t*, conditioned by the data up to the time τ
$\hat{\mathbf{x}}_{t\|t-1}\left(\boldsymbol{\theta}\right)$	expected value of the state as a function of parameter vector $\boldsymbol{\theta}$
$\tilde{\mathbf{x}} = \mathbf{x}(t) - \hat{\mathbf{x}}\left(t \mid t-1\right)$	state prediction error
$\boldsymbol{\varepsilon}\left(t \mid t-1\right) \in \mathbb{R}^r$	innovation sequence
$\mathbf{v}(t) \in \mathbb{R}^p$	process noise, stochastic input
$\mathbf{e}(t) \in \mathbb{R}^r$	measurement noise, stochastic input
$\mathbf{Q} \in \mathbb{R}^{p\times p}$	covariance matrix of the process noise
$\mathbf{R} \in \mathbb{R}^{r\times r}$	covariance matrix of the measurement noise
$\mathbf{S} \in \mathbb{R}^{p\times r}$	cross covariance between the process and measurement noise

$\mathbf{P}_0 \in \mathbb{R}^{n\times n}$	covariance matrix of the initial state estimate
$\mathbf{P}$, $\mathbf{P}_x \in \mathbb{R}^{n\times n}$	covariance matrix of the state prediction error
$\mathbf{P}_y \in \mathbb{R}^{r\times r}$	covariance matrix of the output prediction error
$\mathbf{I}_a \in \mathbb{R}^{a\times a}$	identity matrix
$\mathbf{A}^T$	matrix transpose
$\mathcal{E}\{\ \}$, μ	expected value operator, mean value
$\mathrm{cov}\{\ \}$	covariance operator
σ^2	variance
σ	standard deviation
$*$	convolution
$p(x \mid y)$	conditional probability distribution of x conditioned by y
$p(x, y)$	joint probability distribution of values x, y
$\delta(t)$	Dirac impulse or a unit impulse in discrete time
$\mathrm{tr}(\mathbf{A})$	trace of matrix $\mathbf{A}$
$\mathbf{A}^\dagger$	pseudo inverse of the matrix
$\mathrm{diag}(\mathbf{d})$	matrix having vector $\mathbf{d}$ on the diagonal
$\bigoplus_{i=1}^{N} \mathbf{A}_i$	direct sum
$\otimes$	Kronecker product
$\oplus$	Kronecker summation
$\mathrm{vec}\{\mathbf{A}\}$	vectorization of matrix $\mathbf{A}$
$\mathbf{A}_{\mathrm{vec}}, (\mathbf{A})_{\mathrm{vec}}$	vectorized matrix $\mathbf{A}$, after operator $\mathrm{vec}\{\ \}$ was applied
$\mathrm{vecMin}\{\mathbf{A}\}$	minimal vectorization of a symmetric matrix $\mathbf{A}$
$\mathbf{A}_{\mathrm{mvec}}, (\mathbf{A})_{\mathrm{mvec}}$	vectorized matrix $\mathbf{A}$, after operator $\mathrm{vecMin}\{\ \}$ was applied
$\boldsymbol{\mathcal{P}}_{p,q,N}$	permutation matrix
$\boldsymbol{\mathcal{D}}_n$	duplication matrix

$\boldsymbol{\mathcal{K}}_{n,p}$	commutation matrix
a_{ij}, $\left[\mathbf{A}\right]_{ij}$	member of matrix $\mathbf{A}$ with index i, j
$\left\|.\right\|_2, \left\|.\right\|$	quadratic norm
$\left\|.\right\|_\infty$	infinite norm
$\left\|.\right\|_F$	Frobenius norm
$\triangleq$	definition of a new expression
$:=$	assignment of the right side value to the variable on the left
$\approx$	approximate equality
$\propto$	equal up to a multiplicative constant
$\sim \mathcal{N}\left(\mu, \mathbf{R}\right)$	generated from the normal distribution given by mean vector μ and covariance matrix $\mathbf{R}$

pdf	probability distribution function
cpdf	conditional probability distribution function
MS	mean square
LMS	linear mean square, minimization of quadratic criterion
ML	maximum likelihood
KF	Kalman filter
SD	Spectral density
LTI	linear time invariant
SISO	single input, single output system
MIMO	multi input, multi output system

8.2 Mathematical definitions

Several necessary definitions are listed in this section. To shorten some formulas, the following abbreviation will be used

$$\mathcal{E}\left\{\left(\mathbf{x}-\hat{\mathbf{x}}\right)^T\left(...\right)\right\}=\mathcal{E}\left\{\left(\mathbf{x}-\hat{\mathbf{x}}\right)^T\left(\mathbf{x}-\hat{\mathbf{x}}\right)\right\}. \tag{8.1}$$

Further, some mathematical operations will be defined. A derivation of a multivariate function *f* is defined as

$$\frac{\partial f\left(x_1,x_2,...,x_n\right)}{\partial \mathbf{x}}=\left[\frac{\partial f\left(x_1,x_2,...,x_n\right)}{\partial x_1}\quad\frac{\partial f\left(x_1,x_2,...,x_n\right)}{\partial x_2}\quad\cdots\quad\frac{\partial f\left(x_1,x_2,...,x_n\right)}{\partial x_n}\right]. \tag{8.2}$$

Differential formulas with respect to a vector are as follows

$$\frac{\partial \mathbf{x}^T\mathbf{A}\mathbf{x}}{\partial \mathbf{x}}=\mathbf{x}^T\left(\mathbf{A}+\mathbf{A}^T\right),\qquad\frac{\partial \mathbf{x}^T\mathbf{B}\mathbf{y}}{\partial \mathbf{x}}=\mathbf{y}^T\mathbf{B}^T,\qquad\frac{\partial \mathbf{y}^T\mathbf{C}\mathbf{x}}{\partial \mathbf{x}}=\mathbf{y}^T\mathbf{C}.$$

A matrix pseudo inverse can be defined in two ways. The first formula is

$$\mathbf{A}^\dagger=(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T, \tag{8.3}$$

where $\mathbf{A}\in\mathbb{R}^{m\times n}$ and $\operatorname{rank}\mathbf{A}=\min(m,n)$. If the rank condition is not satisfied, a Moore–Penrose method can be used. The calculation uses SVD decomposition of matrix **A**, i.e. $\mathbf{A}=\mathbf{U}\mathbf{S}\mathbf{V}^T$. Further, each non–zero singular value is inverted and zero singular values are left zero

$$\mathbf{S}^+=\operatorname{diag}\left[\frac{1}{\sigma_1},\quad\cdots,\quad\frac{1}{\sigma_m},\quad\sigma_{m+1},\quad\cdots,\quad\sigma_n\right],\ \sigma_1\ldots\sigma_m>0,\ \sigma_{m+1}\ldots\sigma_n=0.$$

Then, the MP–pseudo inverse is calculated as

$$\mathbf{A}^\dagger=\mathbf{U}\mathbf{S}^+\mathbf{V}^T \tag{8.4}$$

and has the following properties

$$\begin{aligned}&\mathbf{A}=\mathbf{A}\mathbf{A}^\dagger\mathbf{A},\\&\mathbf{A}^\dagger=\mathbf{A}^\dagger\mathbf{A}\mathbf{A}^\dagger,\\&\left(\mathbf{A}^\dagger\mathbf{A}\right)^T=\mathbf{A}^\dagger\mathbf{A}.\end{aligned} \tag{8.5}$$

The trace of a square matrix is defined as a sum of its diagonal members and is equal to the sum of all its eigenvalues λ_i

$$\mathrm{tr}\left(\mathbf{A}\right) = \sum_i a_{ii} = \sum_i \lambda_i. \tag{8.6}$$

Following formulas hold for the trace operator

$$\begin{aligned} \mathrm{tr}\left(\mathbf{AB}\right) &= \mathrm{tr}\left(\mathbf{BA}\right), & \frac{\partial \,\mathrm{tr}\left(\mathbf{ABA}^T\right)}{\partial \mathbf{A}} &= 2\mathbf{AB} \quad \text{where } \mathbf{B} = \mathbf{B}^T, \\ \mathrm{tr}\left(\mathbf{A}\right) &= \mathrm{tr}\left(\mathbf{A}^T\right), & \frac{\partial \,\mathrm{tr}\left(\mathbf{AXB}\right)}{\partial \mathbf{X}} &= \mathbf{A}^T\mathbf{B}^T, \\ \mathrm{tr}\left(\alpha\mathbf{A} + \beta\mathbf{B}\right) &= \alpha\,\mathrm{tr}\left(\mathbf{A}\right) + \beta\,\mathrm{tr}\left(\mathbf{B}\right), & \frac{\partial \,\mathrm{tr}\left(\mathbf{AX}^T\mathbf{B}\right)}{\partial \mathbf{X}} &= \mathbf{AB}. \end{aligned} \tag{8.7}$$

Matrix inequality $\mathbf{A} \geq \mathbf{B}$ means, that the difference between the matrices $\mathbf{A} - \mathbf{B}$ is a positive semi definite matrix having all eigen values nonnegative.

The Kronecker product and Kronecker sum of matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{m\times m}$ are defined as follows

$$\mathbf{A} \otimes \mathbf{B} \triangleq \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}, \qquad \mathbf{A} \oplus \mathbf{B} \triangleq \left(\mathbf{A} \otimes \mathbf{I}_m\right) + \left(\mathbf{I}_n \otimes \mathbf{B}\right). \tag{8.8}$$

Direct sum of matrices $\mathbf{A}_i$ is defined as

$$\bigoplus_{i=1}^{N} \mathbf{A}_i \triangleq \begin{bmatrix} \mathbf{A}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{A}_N \end{bmatrix}. \tag{8.9}$$

Vectorization and minimal vectorization[9] operators of a matrix can be defined as follows

[9] Matlab functions for vectorization and other operators can be found in Appendix A

$$\mathbf{A}=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad \begin{aligned} \mathbf{A}_{\text{vec}} &= \text{vec}\left\{\mathbf{A}\right\} = \begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{12} & a_{22} & a_{32} & a_{13} & a_{23} & a_{33} \end{bmatrix}^T, \\ \mathbf{A}_{\text{mvec}} &= \text{vecMin}\left\{\mathbf{A}\right\} = \begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{22} & a_{32} & a_{33} \end{bmatrix}^T. \end{aligned}$$

Gaussian (normal) distribution is given by a probability density function of the form

$$\mathcal{N}\left(\mu, \sigma^2\right): \quad f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}\left(x-\mu\right)^2\right), \tag{8.10}$$

where μ, σ^2 are a mean value and a variance. For the multivariate normal distribution the probability density is given by

$$\mathcal{N}\left(\boldsymbol{\mu}, \mathbf{P}\right): \quad f(x_1, x_2, ..., x_n) = \frac{1}{\left(2\pi\right)^{n/2}\sqrt{\det\left(\mathbf{P}\right)}} \exp\left(-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}\right)^T \mathbf{P}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}\right)\right), \tag{8.11}$$

where $\boldsymbol{\mu}$ is a vector of the mean values and the covariance matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$ is symmetric and positive definite.

The Frobenius norm $\left\|.\right\|_F$ is defined for matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$ as

$$\left\|\mathbf{X}\right\|_F = \sqrt{\sum_{j=1}^{m}\sum_{i=1}^{n} x_{i,j}} = \sqrt{\text{tr}\left(\mathbf{X}\mathbf{X}^T\right)}. \tag{8.12}$$

9. Appendix A – Functions for Matlab[10]

```
function v = vec (A)
%VEC  A columnwise stacking of a matrix into a vector
%
%v = vec (A)

v = A(:);
```

```
function V = vecMin (v)
%VECMIN     A columnwise stacking of a matrix to vector with
repetitions
%removed
%
%v = vecMin (A)
%
%Parameters:
%A - square symetric input matrix of size p x p
%
%Output
%v - vector of lenght p(p-1)/2
%
%Example:
%A = [1 2 3;
%     2 5 6;
%     3 6 7];
%v = [1 2 5 3 6 7]';
%
%The repetitions are removed if i < j, where 'i' is a row num-
ber and 'j' is
%a column number

L = zeros(size(v));
j = 1;

for i=1:size(v,2);
    L(i, 1:j) = 1;
    j = j + 1;
end
L = logical(L);
V = v(L);
```

[10] When copying the Matlab code from the appendix, beware of wordwrap and hyphenation.

```
function V = unvecMin (v, height)
%UNVECMIN    A columnwise unstacking of vector back to matrix.
Inverse
%operation to vecMin function
%
%A = unvecMin (v, height)
%
%Parameters:
%v      - vector returned by vecMin function of size p(p+1)/2
%height - the number of rows of returned matrix
%
%Output
%A - square symetric matrix of size height x height
%
%Example:
%v = [1 2 5 3 6 7]'; h = 3;
%A = unvecMin(v, h);
%
%A = [1 2 3;
%     2 5 6;
%     3 6 7];
%
%The repetitions were removed in vecMin function if i < j,
where 'i' is a %row number and 'j' is a column number. unvec-
Min is inverse operation.

V = zeros(height, height);
j = 0;  vi = 0;

for i=1: height
    V(i:end, i) = v(i+vi: height +vi);
    j = height - i;
    vi = vi + j;
end
V = V + V' - diag(V(logical(eye(height))));
```

```
function V = unvec (v, height)
%UNVEC    Unstacking the vector back to matrix
%
% V = unvec(A, height)
%
% A      - input matrix
% height - the number of rows of the final matrix
```

```
V = reshape(v, height, length(v)/height);
function DS = direct_sum(A, N)
%DIRECT_SUM   Creates a direct sum of input matrix
%
%D = direct_sum(A, N)
%
%Parameters:
%
%A - a square matrix of size p x p
%N - a number of sumation
%
%Output:
%
%D - the matrix of size N*p x N*p
%D is block diagonal matrix (A, A, ..., A), where the number
of diagonal blocks is N

n = size(A,1);
m = size(A,2);
DS = zeros(n, m);

for i=0:N-1
    DS(i*n+1:i*n+n, i*m+1:i*m+m) = A;
end
```

```
function R = kron_sum(A, B)
%KRON_SUM   Kronecker sumation
%
%R = kron_sum(A, B)
%
%R = kron(A, Im) + kron(In, B)
%
%where A is of size n x n
%      B is of size m x m
%
%      In = eye(n)
%      Im = eye(m)

Im = eye(size(B,1));
In = eye(size(A,1));

R = kron(A, Im) + kron(In, B);
```

```
function D = duplication_matrix(n)
%DUPLICATION_MATRIX      The function creates a special dupli-
cation matrix
%
%D = duplication_matrix(n)
%
%Duplication matrix consist of zeros and ones so that next eq-
uation holds
%
% D*vecMin(A) =  vec(A)
%
%Parameters:
%
%n - the size of square matrix A
%
%Output:
%
%The matrix of size n^2 x n(n+1)/2

D = zeros(n^2, n*(n+1)/2);
i = 0;
m = n;

while i < n

    o = i + 1;
    k = n;
    for j=1:i
        D(i*n+j, o) = 1;
        o = sum(n:-1:k) + i-j+1;
        k = k - 1;
    end

    D(i*n+1+i:i*n+n,m-n+1:m-i) = eye(n-i);
    m = m + n - i;
    i = i + 1;
end
```

```
function P = permuteMN (p, q, N)
%PERMUTEM       The function creates a special permutation ma-
trix
%
%Permutation matrix consist of zeros and ones so that next eq-
uation holds
%
%vec( direct_sum(A, N) ) = P_pqN*vec(A)
%
%Parameters:
%
%p, q - the size of matrix A, [p, q] = size(A)
%N - the number of direct sumations at direct_sum(A, N)
%
%Output:
%
%The matrix of size p*q*N^2 x p*q

P = zeros(p*q*N^2, p*q);
I = eye(p);

i = 1;
m = 1;

while i*p <= p*q*N^2

    P(i*p-p+1:i*p, m*p-p+1:m*p) = I;

    i = i + N;
    m = m + 1;
    if m > q
        m = 1;
        i = i + 1;
    end
end
```

10. References

Akesson B. M., J. B. Jorgensen, N. K. Poulsen and S. B. Jorgensen (2008). A generalized autocovariance least–squares method for Kalman filter tuning. *Journal of Process control*, Vol. 18, pp. 769–779.

Anderson B. D. O. and J. N. Moore (1979). *Optimal filtering*. Prentice Hall, USA, ISBN: 0–486–43938–0.

Antsaklis P. J. and Michel A. N. (1997). *Linear Systems*. McGraw Hill, New York, USA, ISBN: 978–0–81–764434–5.

Åström K. J. (1970). *Introduction to Stochastic Control Theory*. Dover Publications, USA, ISBN: 0–486–445313–3.

Belanger P. R. (1974). Estimation of noise covariance matrices for a linear time–varying stochastic process. *Automatica*, Vol. 10, pp. 267–275.

Bryson A. E and D. E. Johansen (1965). Linear filtering for time-varying systems using measurements containing colored noise. *IEEE Transactions on Automatic Control*, Vol. 10, No. 1, pp. 4 - 10.

Candy J. V. (2009). *Bayesian Signal Processing, Classical, Modern, and Particle Filtering Methods*. John Wiley & Sons, USA, ISBN: 978–0–470–18094–5.

Carew B. and P. R. Belanger (1973). Identification of optimum filter steady–state gain for systems with unknown noise covariances. *IEEE Trans. on Automat. Control*, Vol. 18, pp. 582–587.

Duník J., M. Šimandl and O. Straka (2009). Methods for estimating state and measurement noise covariance matrices: Aspects and comparison. *Proceedings of the 15th IFAC Symposium on System Identification*, Saint–Malo, France, pp. 372–377.

Eshel G. The Yule Walker Equations for the AR Coefficients. [pdf] Available at <www.stat.sc.edu/~vesselin/STAT520_YW.pdf> [Accessed 30th January 2012].

Fuller W. A (1996). *Introduction to statistical time series*. John Wiley & Sons, USA, ISBN: 978–0–47–155239–0.

Gibbs B. P. (2011). *Advanced Kalman Filtering, Least–squares and Modeling*. John Wiley & Sons, USA, ISBN: 978–0–470–89004–2.

Hendry D. F. and Juselius K. (2000). Explaining Cointegration Analysis: Part II, *The Energy Journal*, Vol. 22, No. 2, pp. 1–52.

Kailath T. (1979). *Linear Systems*. Prentice Hall, USA, ISBN: 978–0–13–536961–6.

Kailath T., Sayed A. H. and Hassibi B. (2000). *Linear Estimation*. Prentice Hall, USA.

Kalman R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*. Vol. 82, No. 4, pp. 35 – 45.

Kalman R. E., Bucy R. S. (1961). New Results in Linear Filtering and Prediction Theory. *Journal of Basic Engineering*. Vol. 83, No. 1, pp. 95 – 109.

Kalouptsidis N. (1997). *Signal Processing Systems. Theory and Design*. John Wiley & Sons, USA.

Katayama T. (2005). *Subspace Methods for System Identification*. Springer, Germany, ISBN: 978–1–85233–981–4.

Kunt M. R. (2011). *Vector autoregressions*, Educational material, University of Vienna [pdf] Available at: http://homepage.univie.ac.at/robert.kunst/varpres.pdf [Accessed 2nd April 2012].

Ljung L. (1987). *System identification: Theory for the user*. Prentice Hall, USA, ISBN: 0–13–881640–9.

Lütkepohl H. (2005). *New Introduction to Multiple Time Series Analysis*. Springer, Germany, ISBN 3–540–40172–5.

Magnus J. R., Neudecker H. (1999). *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley & Sons. USA, ISBN: 0–471–98632–1.

Mehra R.K. (1970). On the identification of variances and adaptive Kalman filtering. *IEEE Trans. on Automat. Control*, Vol. 15, No. 2, pp. 175–184

Mehra R. K. (1972). Approaches to Adaptive Filtering. *IEEE Trans. on Automat. Control*, Vol. 17, No. 5, pp. 693–698.

Neethling C. and P. Young (1974). Comments on „Identification of optimum filter steady–state gain for systems with unknown noise covariances“. *IEEE Trans. on Automat. Control*, Vol. 19, No. 5, pp. 623–625.

Odelson B. J., M. R. Rajamani and J. B. Rawlings (2005). A new autocovariance least–squares method for estimating noise covariances. *Automatica*,Vol. 42, No. 2, pp. 303–308

Papoulis A. (1991). *Probability, Random Variables, and Stochastic Processes*. Third edition, Mc. Graw–Hill, Inc., USA, ISBN: 0–07–048477–5.

Popescu D. C and I. Zeljkovic (1998). Kalman filtering of colored noise for speech enhancement. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 997–1000.

Peterka V. (1981). Bayesian approach to system identification. In P. Eykhoff (Ed.), *Trends and Progress in System Identification*, Oxford, UK.

Pour N.D., B. Huang and S.L. Shah (2009). Consistency of noise covariance estimation in joint input–output closed loop subspace identification with application in LQG benchmark. *Journal of Process Control*, Vol. 19, No. 10, pp. 1649–1657.

Pukkila M. T. and P. R. Krishnaiah (1988). On the use of autoregressive order determination criteria in univariate white noise tests. *IEEE Trans. on Acoustic, Speech and Signal Processing*, Vol. 36, No. 5, pp. 764–774.

Pukkila M. T. and P. R. Krishnaiah (1988). On the use of autoregressive order determination criteria in multivariate white noise tests. *IEEE Trans. on Acoustic, Speech and Signal Processing*, Vol. 36, No. 9, pp. 1396–1403.

Rajamani M. R. and J. B. Rawlings (2009). Estimation of the disturbance structure from data using semidefinite programming and optimal weighting. *Automatica*, Vol. 45, pp.1–7.

Rajamani M. R. and J. B. Rawlings. [online] Available at <http://jbrwww.che.wisc.edu/software/als> [Accessed 1st March 2012].

Rencher A. C. (2002). *Methods of Multivariate Analysis, 2nd Edition*. John Wiley & Sons, USA, ISBN: 978–0–47–127135–2.

Řehoř J. and V. Havlena (2011). A Practical Approach to Grey–box Model Identication. *Proceedings of 18th IFAC World Congress*, Milano, Italy.

Řehoř J. and V. Havlena (2010). Grey–box model identication – control relevant approach. *Proceedings of IFAC Workshop Adaptation and Learning in Control and Signal Processing*, Antalya, Turkey.

Salzmann M and P. Teunissen (1990). Detection *and Modelling of Coloured Noise for Kalman Filter Applications*. [pdf] Available at <http://saegnss1.curtin.edu.au/ Publications/1990/Salzmann1990Detection.pdf> [Accessed 17th April 2012].

Seber G. A. F. (2008). *Multivariate Observations*. John Wiley & Sons, USA, oISBN: 978–0–47–031664–1.

Simon D. (2006). *Optimal state estimation, H_{inf}, and Nonlinear Approaches*. John Wiley & Sons, Inc., New Jersey.

Šimandl M., J. Královec and P. Tichavský (2001). Filtering, predictive, and smoothing Cramér–Rao bounds for discrete–time nonlinear dynamic systems. *Automatica*. Vol. 37, No. 11, pp. 1703–1716.

Šimandl M., J. Královec and T. Söderström (2006). Advanced point–mass method for nonlinear state estimation. *Automatica,* Vol. 42, No. 7, pp. 1133–1145.

Štecha J. and V. Havlena (1999). *Moderní teorie řízení*. Ediční středisko ČVUT, Prague.

Tichavský P., Muravchik C. H. and Nehorai A. (1998). Posterior Cramér–Rao Bounds for Discrete–Time Nonlinear Filtering. *IEEE Trans. on Signal Processing*. Vol. 46, No. 5, pp. 1386–1396.

Wald A. (1945). Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, Vol. 16, No. 2, pp. 117 – 186.

Author's publications

Matisko P. (2009). *Estimation of covariance matrices of the noise of linear stochastic systems*. Diploma Thesis, Czech Technical University in Prague.

Matisko P. and V. Havlena (2010). Noise covariances estimation for Kalman filter tuning. *Proceedings of IFAC Workshop Adaptation and Learning in Control and Signal Processing*, Antalya, Turkey.

Matisko P. and V. Havlena (2011). Cramer–Rao bounds for estimation of linear system noise covariances. *Proceedings of 18th IFAC World Congress*, Milano, Italy.

Matisko P. and V. Havlena (2012a). Optimality tests and adaptive Kalman filter. *Proceedings of 16th IFAC System Identification Symposium*, Brussels, Belgium.

Matisko P. and V. Havlena (2012b). Cramer–Rao bounds for estimation of linear system noise covariances. *Journal of Mechanical Engineering and Automation.* Vol. 2, No. 2, pp. 6–11, e–ISSN: 2163–2413, DOI: 10.5923/j.jmea.20120202.02.

Matisko P. and V. Havlena (2013). Noise covariance estimation for Kalman filter tuning using Bayesian approach and Monte Carlo. Accepted in *International Journal of Adaptive Control and Signal* Processing, DOI: 10.1002/acs.2369.

11. Index

O

P

R

S

T

V

W

Y

12. Author index

13. Curriculum Vitae

Peter Matisko was born in Zvolen, Slovakia in 1985. He received his bachelor's and master's degree at the Czech Technical University in Prague, Faculty of Electrical Engineering. In 2009, he started PhD studies at the Department of Control Engineering. His studies and research work were concentrated on control, stochastic systems and estimation theory. In February 2011, he took the Modern predictive control course at ETH in Zurich.

During the doctoral studies he was participating on teaching at the Department of Control Engineering. The teaching activities covered Systems and Models, Automatic Control, Theory of Dynamic Systems, Estimation Filtration and Detection.

His research results were presented at conferences IFAC ALCOSP&PSYCHO (Turkey), IFAC World Congress (Italy) and IFAC System Identification Symposium (Belgium). The main results were published in the Journal of Adaptive Control and Signal Processing.