Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Maximum Likelihood Estimation of LTI Continuous-Time Grey-box Models

Jiří Řehoř * **Vladimír Havlena** **

* *Department of Control Engineering, FEE, CTU, Karlovo namesti 13, Praha 2, The Czech Republic (e-mail: jiri.rehor@fel.cvut.cz).*
** *Department of Control Engineering, FEE, CTU, Karlovo namesti 13, Praha 2, The Czech Republic (e-mail: vladimir.havlena@honeywell.com).*

Abstract: Grey-box models give us a welcomed opportunity to combine our prior knowledge with experimental data when a system is being identified. This benefit is redeemed by an unattractive non-convex and nonlinear optimisation problem that ensue from the parameter estimation. The article shows an efficient method how to speed up the arising iterative optimisation algorithm in the case of continuous, time-invariant, grey-box models. The method is based on a presented output-to-parameter sensitivity computation algorithm.

Keywords: Continuous-time model identification, Maximum Likelihood Estimation, Grey-box, Linear systems.

1. INTRODUCTION

Models have became a basic part of many advanced control systems. A typical example is the Model Predictive Control (MPC) that uses models to find an optimal control strategy. This work was originally motivated by the need for suitable models for the MPC design, however, it can be applied to a wide range of Linear Time-Invariant (LTI) modelling problems as well.

Basically, there are three kinds of LTI models that are used to describe the real system (see e.g. Ljung (1999)). The well-known black box model in which only a degree of the model is given *a priori*. On the opposite site, there is a white-box model, which describes the behaviour of the system using first principle knowledge. Finally, between them, there is a Grey-Box Model (GBM) which combines both approaches (we may know the model structure, but we don't know the exact values of parameters); see e.g. Johansen (1996), Kristensen et al. (2004) and Bohlin (2006).

This paper deals with the GBMs because they can be exploited in industrial practice; especially when a designer has only limited and usually insufficiently excited input/output experimental data. However, the lack of information in data could be replaced by additional *a priori* knowledge about the modelled process. This knowledge can be easily incorporated just into the GBM (Tulleken (1993)). The advantages of the GBMs are especially seen in the case of Multiple-Input-Multiple-Output (MIMO) systems, where the knowledge of the inner structure helps to reduce the number of parameters to be estimated and makes the identification more robust to errors in data.

The main aim of this work is to present a convenient approach to GBM parameter estimation. The GBMs are considered to be defined in the continuous-time domain, whereas the discrete-time models, which are used by optimisation algorithm, are effectively constructed by the proposed algorithm.

The structure of the paper is as follows: In the next section a GBM parametrisation will be introduced. In Section 3, the Maximum Likelihood estimation of GBM parameters will be briefly presented and the resulting criterion function and the corresponding gradient will be examined. Section 4 deals with a new algorithm of a structural exponential matrix computation. Finally, in Section 5, a grey-box identification example will be presented. The last section concludes the paper.

2. GREY-BOX MODEL DEFINITION

2.1 Grey-box model description

There are several ways how to define GBM. In our case, the most suitable form is a state-space description, where a physical insight into the modelled system can be easily incorporated.

Let $\boldsymbol{\theta}_c \in \mathbb{R}^{n_p}$ denote a vector of parameters. The parametrised state-space description of the GBM is

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\boldsymbol{\theta}_c)\mathbf{x}(t) + \mathbf{B}(\boldsymbol{\theta}_c)\mathbf{u}(t), \tag{1a}$$

$$\mathbf{y}(t) = \mathbf{C}(\boldsymbol{\theta}_c)\mathbf{x}(t) + \mathbf{D}(\boldsymbol{\theta}_c)\mathbf{u}(t), \tag{1b}$$

where $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is a state vector, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ and $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ are vectors of inputs and outputs, respectively. The parametrised matrices have dimensions: $\mathbf{A}(\boldsymbol{\theta}_c) \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B}(\boldsymbol{\theta}_c) \in \mathbb{R}^{n_x \times n_u}$, $\mathbf{C}(\boldsymbol{\theta}_c) \in \mathbb{R}^{n_y \times n_x}$ and $\mathbf{D}(\boldsymbol{\theta}_c) \in \mathbb{R}^{n_y \times n_u}$.

The prior knowledge about the modelled system has been moved into the structure of these matrices and to the expected values of parameters $\boldsymbol{\theta}_c \in \boldsymbol{\Theta}_c$. The goal is to estimate an "optimal" value of the parameter vector $\boldsymbol{\theta}_c^*$ using a measured experimental data.

2.2 Discrete-time GBM

The system is typically observed in discrete-time intervals. It is thus convenient to transform the system (1) into its discrete-time form.

Let $\boldsymbol{y}_k = \boldsymbol{y}(kT_s)$ be a sampled output and let the manipulated inputs be piece-wise constant between the sampling intervals; i.e. $\boldsymbol{u}(t) = \boldsymbol{u}_k$ for $t \in \langle kT_s, (k+1)T_s)$. The corresponding discrete-time GBM has the following form

$$\mathbf{x}_{k+1} = \mathbf{M}(\boldsymbol{\theta}_c, T_s)\mathbf{x}_k + \mathbf{N}(\boldsymbol{\theta}_c, T_s)\mathbf{u}_k, \tag{2a}$$

$$\mathbf{y}_k = \mathbf{C}(\boldsymbol{\theta}_c)\mathbf{x}_k + \mathbf{D}(\boldsymbol{\theta}_c)\mathbf{u}_k, \tag{2b}$$

where the matrices $\mathbf{M}$ and $\mathbf{N}$ could be computed using a matrix exponential (see Ljung (1999)) as

$$\begin{pmatrix} \mathbf{M} & \mathbf{N} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} = \exp\left[\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathrm{T_s}\right]. \tag{3}$$

2.3 Stochastic GBM

Both equations (1) and (2) define the GBM from the deterministic point of view. The stochastic description of disturbances affecting the model is usually unknown to the designer and some general stochastic model should be used to model them (for more information about the estimation of stochastic parameters see e.g. Matisko and Havlena (2013)). A suitable stochastic model is crucial for a consistent estimation of the deterministic parameters (Ljung (1999)).

To introduce a stochastic part to the GBM, we extend the discrete-time version of the GBM as follows:

$$\mathbf{x}_{k+1} = \mathbf{M}(\boldsymbol{\theta}_c, T_s)\mathbf{x}_k + \mathbf{N}(\boldsymbol{\theta}_c, T_s)\mathbf{u}_k + \mathbf{K}(\boldsymbol{\theta}_\mathbf{K})\mathbf{e}_k, \tag{4a}$$

$$\mathbf{y}_k = \mathbf{C}(\boldsymbol{\theta}_c)\mathbf{x}_k + \mathbf{D}(\boldsymbol{\theta}_c)\mathbf{u}_k + \mathbf{e}_k, \tag{4b}$$

where $\mathbf{e}_k \in \mathbb{R}^{n_y}$ is a Gaussian white noise

$$\mathbf{e}_k \sim \mathcal{N}(\mathbf{0}; \mathbf{R}), \quad \mathrm{cov}\{\mathbf{e}_k, \mathbf{e}_{k+j}\} = 0, j \neq 0. \tag{5}$$

The model of this kind is sometimes referred to as a "hybrid" GBM (Young et al. (2008)), because it combines both domains, the continuous-time and the discrete-time. The vector of parameters of the stochastic GBM is obtained by completing the deterministic parameters $\boldsymbol{\theta}_c$ by parameters that describe the matrix $\mathbf{K} \in \mathbb{R}^{n_x \times n_y}$ and the symmetric matrix $\mathbf{R} \in \mathbb{R}^{n_y \times n_y}$.

3. MAXIMUM LIKELIHOOD ESTIMATION

When the GBM is constructed and parametrised, the next step is to estimate the optimal values of parameters using experimental data. There are numerous estimation methods that can be used. However, probably the most common approaches are Prediction Error Method (PEM), Least Squares Estimation (LSE), or Maximum Likelihood Estimation (MLE). In this section the more general MLE approach to the GBM parameter estimation will be introduced.

3.1 Derivation of the criterion function

The goal of the MLE method is: for a given data sequence

$$\left\{\mathcal{Y}^N; \mathcal{U}^N\right\} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N;\ \boldsymbol{u}_1, \ldots, \boldsymbol{u}_N\}$$

and for a given model structure (4) find an argument $\boldsymbol{\theta}^*$ that maximises the likelihood function

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\theta}|\mathcal{Y}^N, \mathcal{U}^N\right) &= p\left(\mathcal{Y}^N|\boldsymbol{\theta}, \mathcal{U}^N\right) \\ &= \left(\prod_{k=1}^{N} p\left(\mathbf{y}_k|\mathcal{Y}^{k-1}, \mathcal{U}^k, \boldsymbol{\theta}\right)\right). \end{aligned} \tag{6}$$

Assume that the data was generated by the stochastic GBM (4) under the conditions (5) and let $\mathbf{x}_0$ be a deterministic initial state. Then, the probability density function $p\left(\mathbf{y}_k|\mathcal{Y}^{t-1}, \mathcal{U}^t, \boldsymbol{\theta}\right)$ is Gaussian; i.e.,

$$p\left(\mathbf{y}_k|\cdot\right) = (2\pi)^{-\frac{n_y}{2}} \det \mathbf{R}^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{\epsilon}_k^T \mathbf{R}^{-1} \boldsymbol{\epsilon}_k\right), \tag{7}$$

where $\boldsymbol{\epsilon}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k$ is a prediction error and $\hat{\mathbf{y}}_k$ is given by a steady-state Kalman filter

$$\hat{\mathbf{x}}_{k+1} = (\mathbf{M} - \mathbf{KC})\,\hat{\mathbf{x}}_k + (\mathbf{N} - \mathbf{KD})\,\mathbf{u}_k + \mathbf{K}\mathbf{y}_k, \tag{8a}$$

$$\hat{\mathbf{y}}_k = \mathbf{C}\hat{\mathbf{x}}_k + \mathbf{D}\mathbf{u}_k, \tag{8b}$$

$$\hat{\mathbf{x}}_1 = \mathbf{x}_0. \tag{8c}$$

Lemma 1. Finding optimal arguments of the likelihood function (6) is equivalent to minimizing the (log) determinant of an estimated covariance matrix $\hat{\mathbf{R}}$

$$\{\boldsymbol{\theta}_c^*, \boldsymbol{\theta}_K^*, \mathbf{x}_0^*\} = \arg\min_{\theta_c, \theta_L, \mathbf{x}_0} \frac{1}{2} \log\det\left(\hat{\mathbf{R}}\right), \tag{9a}$$

$$\hat{\mathbf{R}} = \frac{1}{N}\sum_{k=1}^{N} \boldsymbol{\epsilon}_k \boldsymbol{\epsilon}_k^T. \tag{9b}$$

Proof. Using the definition of the Gaussian pdf (7), the transformed log-likelihood function becomes

$$\log \mathcal{L}\left(\cdot\right) = \mathrm{const} - \frac{N}{2}\log\det\left(\mathbf{R}\right) - \frac{1}{2}\sum_{k=1}^{N} \boldsymbol{\epsilon}_k^T \mathbf{R}^{-1} \boldsymbol{\epsilon}_k,$$

where the "const" stands for parameter independent terms. Maximising the previous log-likelihood function is equivalent to minimising the following cost

$$\begin{aligned} V\left(\cdot\right) &= \frac{N}{2}\left[\log\det\left(\mathbf{R}\right) + \mathrm{trace}\left(\sum_{t=1}^{N} \frac{1}{N}\boldsymbol{\epsilon}(t)\boldsymbol{\epsilon}(t)^{\mathrm{T}}\mathbf{R}^{-1}\right)\right] \\ &= \frac{N}{2}\left[\log\det\left(\mathbf{R}\right) + \mathrm{trace}\left(\hat{\mathbf{R}}\mathbf{R}^{-1}\right)\right]. \end{aligned} \tag{10}$$

Let $\boldsymbol{\theta} = \left[\boldsymbol{\theta}_c^T, \boldsymbol{\theta}_L^T, \mathbf{x}_0^T\right]^T \in \mathbb{R}^n$ be a composed vector of unknown parameters. The optimisation problem can be rewritten as a nested optimisation

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \left[\min_{\mathbf{R}} V\left(\boldsymbol{\theta}, \mathbf{R}\right)\right]. \tag{11}$$

By using a necessary optimality condition

$$\begin{aligned} \mathbf{0} &= \frac{\partial}{\partial \mathbf{R}}\left(\frac{N}{2}\left[\log\det\left(\mathbf{R}\right) + \mathrm{trace}\left(\hat{\mathbf{R}}\mathbf{R}^{-1}\right)\right]\right) \\ &= \frac{N}{2}\left[\mathrm{trace}\left(\left(\mathbf{I} - \hat{\mathbf{R}}\mathbf{R}^{-1}\right)\mathbf{R}^{-1}\right)\right] \end{aligned}$$

and by using the fact that that $\mathbf{R} \succ 0$, an optimal argument of the inner optimisation becomes $\mathbf{R}^* = \hat{\mathbf{R}}$.

Note that in the previous derivations the following well known equalities were used:

$$\frac{\partial}{x_{i,j}} \log\left(\det\left(\mathbf{X}\right)\right) = \text{trace}\left(\mathbf{X}^{-1}\frac{\partial}{x_{i,j}}\mathbf{X}\right), \quad (12)$$

$$\frac{\partial}{x_{i,j}} \text{trace}\left(\mathbf{X}^{-1}\right) = -\text{trace}\left(\mathbf{X}^{-1}\left(\frac{\partial}{x_{i,j}}\mathbf{X}\right)\mathbf{X}^{-1}\right). \quad (13)$$

Inserting the optimal argument into the equation (10) we obtain

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \frac{N}{2}\left(\log\det\left(\hat{\mathbf{R}}\right) + \mathbf{I}\right),$$

which is equivalent to minimisation of the cost (9). □

3.2 Gradient computation

The main drawback of the proposed GBM MLE method is the resulting non-convex optimisation problem (9). Reaching the global optima of this computationally cumbersome task cannot be guaranteed and only a local optima is searched.

In general, the minimisation must be done by an iterative search

$$\hat{\boldsymbol{\theta}}^{(j+1)} = \hat{\boldsymbol{\theta}}^{(j)} + \Delta\hat{\boldsymbol{\theta}}^{(j)}, \quad (14)$$

where $\hat{\boldsymbol{\theta}}^{(j)}$ is an estimation in the j-th step and $\Delta\hat{\boldsymbol{\theta}}^{(j)}$ is an iteration update. Several solvers were designed to solve this problem and a basis of almost all of them is a gradient function [1]

Let $\mathbf{E}(\boldsymbol{\theta}) \in \mathbb{R}^{\mathrm{n_y}\times\mathrm{N}}$ denote a matrix of of residuals

$$\mathbf{E} = [\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_N] \quad (15)$$

and let

$$\mathbf{E}_{\theta_i} = \frac{\partial}{\partial\theta_i}\mathbf{E}(\boldsymbol{\theta}) = \left[\frac{\partial\boldsymbol{\epsilon}_1}{\partial\theta_i}, \ldots, \frac{\partial\boldsymbol{\epsilon}_N}{\partial\theta_i}\right] \quad (16)$$

be a matrix of first order derivatives with respect to parameter θ_i.

Lemma 2. The gradient of the criterion function (9) can be evaluated as follows:

$$\nabla V\left(\boldsymbol{\theta}\right) = \left[\frac{\partial V(\boldsymbol{\theta})}{\partial\theta_1}, \ldots, \frac{\partial V(\boldsymbol{\theta})}{\partial\theta_n}\right], \quad (17)$$

$$\frac{\partial V(\boldsymbol{\theta})}{\partial\theta_i} = \text{trace}\left(\left(\mathbf{E}\mathbf{E}^{\mathrm{T}}\right)^{-1}\mathbf{E}\mathbf{E}_{\theta_i}^{\mathrm{T}}\right). \quad (18)$$

Proof. Using the differentiation rule (12)

$$\begin{aligned}\frac{\partial}{\partial\theta_i}V\left(\theta\right) &= \frac{\partial}{\partial\theta_i}\left(\frac{1}{2}\log\det\left(\frac{1}{N}\mathbf{E}\mathbf{E}^{\mathrm{T}}\right)\right)\\ &= \frac{1}{2}\text{trace}\left(\left(\mathbf{E}\mathbf{E}^{\mathrm{T}}\right)^{-1}\frac{\partial}{\partial\theta_i}\left(\mathbf{E}\mathbf{E}^{\mathrm{T}}\right)\right)\end{aligned}$$

and by exploiting the following properties of the trace

$$\text{trace}\left(\mathbf{E}\mathbf{E}^{T}\right) = \text{trace}\left(\mathbf{E}^{T}\mathbf{E}\right)$$

the result is obtained:

$$\frac{\partial}{\partial\theta_i}V\left(\theta\right) = \text{trace}\left(\left(\mathbf{E}\mathbf{E}^{\mathrm{T}}\right)^{-1}\mathbf{E}\mathbf{E}_{\theta_i}^{\mathrm{T}}\right).$$

□

The problem of the gradient computation has been translated into the evaluation of the prediction error sensitivity function

$$\boldsymbol{\varphi}_{k,i} = \frac{\partial}{\partial\theta_i}\boldsymbol{\epsilon}_k = -\frac{\partial}{\partial\theta_i}\hat{\mathbf{y}}_k.$$

[1] More comprehensive description can be found e.g. in Fletcher (1981).

3.3 Discrete-time output sensitivity

The iterations of the optimisation algorithm are based on the cost and gradient evaluation, for which the prediction errors $\boldsymbol{\epsilon}_k$ and their sensitivities $\boldsymbol{\varphi}_{k,i}$ have to be rapidly evaluated. In this section a recursive algorithm for computing both, $\boldsymbol{\epsilon}_k$ and $\boldsymbol{\varphi}_{k,i}$, will be presented.

It is assumed that the GBM is defined in the continuous-time domain (1). In order to evaluate the output sensitivity, it is necessary to compute the partial derivatives of the discrete-time version of the GBM (2). Whereas computation of matrix derivatives of the continuous-time GBM is relatively simple, just differentiating (1), the computation of derivatives of the discrete-time matrices $\mathbf{M}(\boldsymbol{\theta})$ and $\mathbf{N}(\boldsymbol{\theta})$ is not that straightforward. A solution to this problem is given in the following lemma.

Lemma 3. Let the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{A}_{\theta_i}$ and $\mathbf{B}_{\theta_i}$ be defined for the continuous-time GBM. The matrices $\mathbf{M}$, $\mathbf{N}$, $\mathbf{M}_{\theta_i}$ and $\mathbf{N}_{\theta_i}$ of the discrete-time GBM can be computed as follows

$$\begin{pmatrix}\mathbf{M} & \mathbf{0} & \mathbf{N}\\ \mathbf{M}_{\theta_\mathrm{i}} & \mathbf{M} & \mathbf{N}_{\theta_\mathrm{i}}\\ \mathbf{0} & \mathbf{0} & \mathbf{I}\end{pmatrix} = \exp\left[\begin{pmatrix}\mathbf{A} & \mathbf{0} & \mathbf{B}\\ \mathbf{A}_{\theta_\mathrm{i}} & \mathbf{A} & \mathbf{B}_{\theta_\mathrm{i}}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{pmatrix}\mathrm{T_s}\right], \quad (19)$$

where $\mathbf{X}_{\theta_i} = \frac{\partial}{\partial\theta_i}\mathbf{X}$ is an element-wise first derivative with respect to deterministic (continuous-time) parameters θ_i, $i = 1, \ldots, n_p$.

Proof. Let $\boldsymbol{\xi}_i(t)$ be a sensitivity of the continuous-time state

$$\boldsymbol{\xi}_i(t) = \frac{\partial}{\partial_{\theta_i}}\mathbf{x}(t).$$

Using the state evolution equation (1), the time derivative of $\boldsymbol{\xi}_i(t)$ is

$$\begin{aligned}\frac{d}{dt}\boldsymbol{\xi}_i(t) &= \frac{d}{dt}\left(\frac{\partial}{\partial\theta_i}\mathbf{x}(t)\right) = \frac{\partial}{\partial\theta_i}\left(\frac{d}{dt}\mathbf{x}(t)\right)\\ &= \frac{\partial}{\partial\theta_i}\left(\mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)\right),\\ &= \mathbf{A}_{\theta_i}\mathbf{x}(t) + \mathbf{A}\boldsymbol{\xi}_i(t) + \mathbf{B}_{\theta_i}\mathbf{u}(t).\end{aligned}$$

This can be rewritten using extended state as

$$\begin{bmatrix}\dot{\mathbf{x}}(\mathbf{t})\\ \dot{\boldsymbol{\xi}}(t)\end{bmatrix} = \begin{pmatrix}\mathbf{A} & \mathbf{0}\\ \mathbf{A}_{\theta_\mathrm{i}} & \mathbf{A}\end{pmatrix}\begin{bmatrix}\mathbf{x}(\mathbf{t})\\ \boldsymbol{\xi}(t)\end{bmatrix} + \begin{pmatrix}\mathbf{B}\\ \mathbf{B}_{\theta_\mathrm{i}}\end{pmatrix}\mathbf{u}(\mathbf{t}). \quad (21)$$

The result (19) is obtained by using discretization formulae (3) for the extended LTI system (21).

Note that these relations are not affected by the parameters $\boldsymbol{\theta}_K$ and $\boldsymbol{x}_0$. □

Obviously, the prediction error sensitivity can be computed using the extended state as follows

$$\boldsymbol{\varphi}(t) = -\begin{pmatrix}\mathbf{C}_{\theta_i} & \mathbf{C}\end{pmatrix}\begin{bmatrix}\mathbf{x}(t)\\ \xi_i(t)\end{bmatrix} - \mathbf{D}_{\theta_i}\mathbf{u}(t), \quad i = 1, \ldots, n.$$

Finally, the discrete-time prediction errors $\boldsymbol{\epsilon}_k$ and their sensitivities $\boldsymbol{\varphi}_{k,i}$, $i = 1, \ldots, n$ can be computed together as follows

$$\begin{aligned}\begin{bmatrix}\epsilon_k\\ \varphi_{k,i}\end{bmatrix} &= \begin{bmatrix}\mathbf{y}_k\\ \mathbf{0}\end{bmatrix} - \begin{pmatrix}\mathbf{C} & \mathbf{0}\\ \mathbf{C}_{\theta_i} & \mathbf{C}\end{pmatrix}\begin{bmatrix}\hat{\mathbf{x}}_k\\ \xi_{k,i}\end{bmatrix} - \begin{pmatrix}\mathbf{D}\\ \mathbf{D}_{\theta_i}\end{pmatrix}\mathbf{u}_k\\ \begin{bmatrix}\hat{\mathbf{x}}_{k+1}\\ \xi_{k+1,i}\end{bmatrix} &= \begin{pmatrix}\mathbf{M} & \mathbf{0}\\ \mathbf{M}_{\theta_i} & \mathbf{M}\end{pmatrix}\begin{bmatrix}\hat{\mathbf{x}}_k\\ \xi_{k,i}\end{bmatrix} + \begin{pmatrix}\mathbf{N}\\ \mathbf{N}_{\theta_i}\end{pmatrix}\mathbf{u}_k +\\ &+ \begin{pmatrix}\mathbf{K} & \mathbf{0}\\ \mathbf{K}_{\theta_i} & \mathbf{K}\end{pmatrix}\begin{bmatrix}\epsilon_k\\ \varphi_{k,i}\end{bmatrix}.\end{aligned} \tag{22}$$

In the last equation, the innovation noise model (4) was employed. $\boldsymbol{\theta}$ is the composed vector of all parameters. Obviously, the partial derivatives of matrices $\mathbf{M}$, $\mathbf{N}$, $\mathbf{C}$ and $\mathbf{D}$ are null for θ_i being a Kalman gain or an initial state parameter, and similarly, $\mathbf{K}_{\theta_i}$ is non-zero only for θ_i being a Kalman gain parameter.

The question that naturally arises is how complex the computation of the matrix exponential (19) is, considering that all n_p parameter sensitivities have to be evaluated each time the $\boldsymbol{\theta}$ is changed. The computational complexity of the matrix exponential is generally $\sim m^3$. In our case (19), the size of the matrix will be $m = 2n_x + n_u$. To compute the derivatives with respect to all n_p parameters the approximate computational cost will be $\sim n_p(2n_x + n_u)^3$. This computational burden could be further reduced by utilisation of a specific matrix structure as will be described in the following section.

Remark 1. Defining GBMs in the continuous-time domain is convenient and has another advantage comparing to the discrete-time definition. That is a block-oriented design. A complex modelled system could be usually divided into simple parts; model blocks, which are mutually connected. It is common that several possibilities of model structures and/or blocks are tested and validated using a data. Instead of redefining the necessary first derivatives of the complete model, already defined model blocks can be reused to build up the overall system definition.

Remark 2. The presented discrete-time prediction errors and their sensitivities can be used in different settings, such as LSE, to estimate the model parameters. Specially for the LSE, the criterion function is defined as

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i=0}^{N} \operatorname{trace}\left(\mathbf{E}\mathbf{E}^{\mathrm{T}}\right),$$

which can be solved by e.g. the Levenberg-Marquardt alghorithm (Marquardt (1963)), The required Jacobian matrix is given by $\mathbf{J}(\boldsymbol{\theta}) = [\mathrm{vec}(\mathbf{E}_{\theta_1}), \ldots, \mathrm{vec}(\mathbf{E}_{\theta_n})]$.

4. EFFECTIVE COMPUTATION OF THE STRUCTURED EXPONENTIAL MATRIX

The exponential matrix can be computed in many ways (see Moler and Van Loan (2003)). One of the preferable methods is the scaling and squaring algorithm with Padé approximation [2] (Higham (2005)). This method will be closely examined and modified to fit better into our structured matrix computation.

4.1 Scaling and squaring algorithm

Consider a matrix exponential function $\boldsymbol{\Psi} = \mathbf{e}^{\boldsymbol{\Omega}}$ The scaling and squaring algorithm has the following steps:

[2] The best approximation of a function by a rational function of a given order.

(1) **Scaling**. The matrix $\boldsymbol{\Omega}$ is scaled by power of 2, so that its norm is less than 1/2
$$\boldsymbol{\Omega}_s = \boldsymbol{\Omega}/2^s. \tag{23}$$
(2) **Padé approximation** $\mathbf{Z}_q \approx e^{\boldsymbol{\Omega}_\mathbf{s}}$ of an order q is computed by solving
$$\mathbf{D}_q\mathbf{Z}_q = \mathbf{E}_q, \tag{24}$$
where
$$\mathbf{E}_q = \mathbf{I} + c(1)\boldsymbol{\Omega}_s + \cdots + c(q)\boldsymbol{\Omega}_s^q \tag{25}$$
and
$$\mathbf{D}_q = \mathbf{I} - c(1)\boldsymbol{\Omega}_s + \cdots + (-1)^q c(q)\boldsymbol{\Omega}_s^q \tag{26}$$
are Padé approximants (matrix polynomials of $\boldsymbol{\Omega}_\mathbf{s}$) and $c(i)$ are Padé coefficients, defined as
$$c(i+1) = c(i)\frac{q-i+1}{i(2q-i+1)},$$
$$c(1) = 1/2.$$
(3) **Squaring** to undo the scaling
$$\boldsymbol{\Psi} \approx \boldsymbol{\Psi}_q = \mathbf{Z}_q^s. \tag{27}$$

4.2 Structured scaling and squaring algorithm

The previous steps of the algorithm could be further simplified when the specific structure of matrix $\boldsymbol{\Omega}$ is considered; i.e., when

$$\boldsymbol{\Omega} = \mathrm{T_s}\begin{pmatrix}\mathbf{A} & \mathbf{0} & \mathbf{B}\\ \mathbf{A}_{\theta_i} & \mathbf{A} & \mathbf{B}_{\theta_i}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{pmatrix}. \tag{28}$$

Lemma 4. Using matrix algebra, the powers of the scaled matrix $\boldsymbol{\Omega}_s$ can be expressed as

$$\boldsymbol{\Omega}_s^k = \left(\frac{\mathrm{T_s}}{2^s}\right)^k \begin{pmatrix}\mathbf{A}^k & \mathbf{0} & \mathbf{A}^{k-1}\mathbf{B}\\ \mathbf{X}_i(k) & \mathbf{A}^k & \mathbf{Y}_i(k)\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{pmatrix}, \tag{29}$$

where

$$\mathbf{X}_i(k) = \sum_{i=1}^{k} \mathbf{A}^{i-1}\mathbf{A}_{\theta_i}\mathbf{A}^{k-i},$$
$$\mathbf{Y}_i(k) = \mathbf{A}^{k-1}\mathbf{B}_{\theta_i} + \mathbf{X}_i(k-1)\mathbf{B}.$$

□

Using the previous lemma and the definition of Padé approximants (25) and (26), the structured version of (24) can be expressed as

$$\begin{pmatrix}\mathbf{D}_{11} & \mathbf{0} & \mathbf{D}_{13}\\ \mathbf{D}_{21} & \mathbf{D}_{11} & \mathbf{D}_{23}\\ \mathbf{0} & \mathbf{0} & \mathbf{I}\end{pmatrix}\begin{pmatrix}\mathbf{Z}_{11} & \mathbf{0} & \mathbf{Z}_{13}\\ \mathbf{Z}_{21} & \mathbf{Z}_{11} & \mathbf{Z}_{23}\\ \mathbf{0} & \mathbf{0} & \mathbf{I}\end{pmatrix} = \begin{pmatrix}\mathbf{E}_{11} & \mathbf{0} & \mathbf{E}_{13}\\ \mathbf{E}_{21} & \mathbf{E}_{11} & \mathbf{E}_{23}\\ \mathbf{0} & \mathbf{0} & \mathbf{I}\end{pmatrix}.$$

Exploring the structure of the previous equation shows:

- Solution to the Padé approximation (24) of the derivative-independent terms is
$$\mathbf{Z}_{11} = \mathbf{D}_{11}^{-1}\mathbf{E}_{11},$$
$$\mathbf{Z}_{13} = \mathbf{D}_{11}^{-1}\left(\mathbf{E}_{13} - \mathbf{D}_{13}\right).$$
As these matrices are independent on matrix derivatives, they are computed only once.
- The derivative-dependent terms are
$$\mathbf{Z}_{21} = \mathbf{D}_{11}^{-1}\left(\mathbf{E}_{21} - \mathbf{D}_{21}\mathbf{E}_{11}\right),$$
$$\mathbf{Z}_{23} = \mathbf{D}_{11}^{-1}\left(\mathbf{E}_{23} - \mathbf{D}_{21}\mathbf{E}_{13} - \mathbf{D}_{23}\right).$$
These have to be computed with respect to each parameter θ_i.

Note that the approximants $\mathbf{D}_q$ and $\mathbf{E}_q$ are a q order polynomials of $\mathbf{\Omega}_s$, which is constructed using q order matrix polynomials of the matrix$\mathbf{A}$. The advantage of this structured algorithm is that the powers of $\mathbf{A}$ have to be computed only once for all i.

Finally the squaring have to be carried out. Let

$$\bar{\mathbf{M}}^{(0)} = \mathbf{Z}_{11},\ \bar{\mathbf{N}}^{(0)} = \mathbf{Z}_{13},\ \bar{\mathbf{M}}_{\theta_i}^{(0)} = \mathbf{Z}_{21},\ \bar{\mathbf{N}}_{\theta_i}^{(0)} = \mathbf{Z}_{13}.$$

Solving

$$\begin{aligned}
\bar{\mathbf{N}}_{\theta_i}^{(j+1)} &= \bar{\mathbf{M}}_{\theta_i}^{(j)}\bar{\mathbf{N}}^{(j)} + \bar{\mathbf{M}}^{(j)}\bar{\mathbf{N}}_{\theta_i}^{(j)} + \bar{\mathbf{N}}_{\theta_i}^{(j)} \\
\bar{\mathbf{M}}_{\theta_i}^{(j+1)} &= \bar{\mathbf{M}}_{\theta_i}^{(j)}\bar{\mathbf{M}}^{(j)} + \bar{\mathbf{M}}^{(j)}\bar{\mathbf{M}}_{\theta_i}^{(j)} \\
\bar{\mathbf{N}}^{(j+1)} &= \bar{\mathbf{M}}^{(j)}\bar{\mathbf{N}}^{(j)} + \bar{\mathbf{N}}^{(j)} \\
\bar{\mathbf{M}}^{(j+1)} &= \bar{\mathbf{M}}^{(j)}\bar{\mathbf{M}}^{(j)}
\end{aligned}$$

for $j = 0$ to s we obtain the desired matrices

$$\mathbf{M} \approx \bar{\mathbf{M}}^{(s)},\ \mathbf{N} \approx \bar{\mathbf{N}}^{(s)},\ \mathbf{M}_{\theta_i} \approx \bar{\mathbf{M}}_{\theta_i}^{(s)},\ \mathbf{N}_{\theta_i} \approx \bar{\mathbf{N}}_{\theta_i}^{(s)}.$$

Before introducing the final example of model building and identification, a short illustration shows the computational savings of the presented structured exponential matrix computation.

4.3 Example of computational savings

Consider a random matrices $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$ and $\mathbf{A}_{\theta_i} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B}_{\theta_i} \in \mathbb{R}^{n_x \times n_u}$, $i = 1, \ldots, n_p$. For various values of n_x, n_u and n_p the computation times of the standard discretisation method (given by (3)) and the new, presented one, will be compared.

For each settings, 40 repeated simulations were done. The median and the upper quartile of the computation times are depicted in the table below.

Table 1. Comparison of computational times $[u_{50\%}, u_{75\%}]$ of the standard (a) and the structured (b) discretisation methods.

n_x		5	25	65
n_u		1	5	13
$n_p = 5$	a	[0.927, 1.066]	[4.709, 5.963]	[46.99, 50.59]
$[ms]$	b	[0.678, 0.802]	[2.609, 3.217]	[17.90, 20.38]
rel.		26.85%	44.59%	61.89%
$n_p = 30$	a	[4.77, 5.70]	[27.07, 29.90]	[265.6, 275.5]
$[ms]$	b	[3.60, 4.35]	[15.41, 19.63]	[95.97, 99.21]
rel.		24.5%	43.078%	63.86%

As can be seen from the results above, the new algorithm is faster especially when the dimension of the problem is getting bigger. This improvement is even more visible when the discretisation algorithm is exploited iteratively inside the optimisation routine.

5. IDENTIFICATION EXAMPLE

5.1 Model definition

Consider a system of a boiler feeding steam into a header (Fig. 1). The input variables are the boiler fuel flow $f_f(t)$ and header steam demand $f_D(t)$, the measured variables are header pressure $p_H(t)$ and boiler-to-header steam flow $f_s(t)$. More detailed description of boiler-header modelling problem is given by Trnka et al. (2013).

$$\mathbf{u}(t) = [f_f(t),\, f_D(t)]^T,\ \mathbf{y}(t) = [p_H(t),\, f_s(t)]^T.$$

The simplified model has the following parts: The block B_1 transfers the input fuel into the generated steam, the steam is accumulated in a drum (B_2), from which it is sucked through the valve (B_3) to the header (B_4) using a drum-to-header pressure difference

$$f_s(t) = K_V\sqrt{\Delta p(t)} = K_V\sqrt{p_D(t) - p_H(t)},$$

where K_V is a boiler-header pipe conductivity. The non-linear valve characteristic is linearised using a steady state operation point definition $f_s(t_{ss}) = F_0$ as

$$f_s(t) \approx \frac{1}{2}\frac{K_V^2}{F_0}\Delta p_H(t) + \frac{1}{2}F_0 = k\Delta p_H(t) + b.$$

List of all parameters and their expected values is denoted in Tab.2.

Table 2. Parameters and their expected values.

param	value	description
K_F	(0; 1]	Fuel-to-steam gain [-],
T_F	[10; 500]	Fuel time constant [s],
V_D	[1e2; 5e5]	Drum (volume) const.[kg/MPa],
V_H	[1e1; 1e5]	Header (volume) const.[kg/MPa],
K_V	[20; 100]	Valve const.[kg/s/MPa],
F_0	[55; 65]	Nominal steam flow [kg/s],
x_{D0}	[9; 13]	Initial state of B_2 [MPa],
x_{H0}	[9; 13]	Initial state of B_4 [MPa],
x_{B0}	[55; 65]	Initial state of B_1 [kg/s].

The overall model description will be

$$\mathbf{A} = \begin{pmatrix} -\frac{k}{V_D} & \frac{k}{V_D} & \frac{1}{V_D} \\ \frac{k}{V_H} & -\frac{k}{V_H} & 0 \\ 0 & 0 & -\frac{1}{T_F} \end{pmatrix},\ \mathbf{B} = \begin{pmatrix} 0 & 0 & -\frac{b}{V_D} \\ 0 & -\frac{1}{V_H} & \frac{b}{V_H} \\ \frac{k_{F2S}}{T_F} & 0 & 0 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} 0 & 1 & 0 \\ k & -k & 0 \end{pmatrix},\qquad \mathbf{D} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & b \end{pmatrix},$$

where an augmented input vector $\bar{\mathbf{u}}(t) = [\mathbf{u}^T(t),\, 1]^T$ has been considered to cover the constant terms. The corresponding state vector is $\mathbf{x}(t) = [x_D(t),\, x_H(t),\, x_B(t)]^T$.

5.2 Simulation results

To preserve the industrial-like settings the following will be considered: a) the excited inputs are limited to a several successive steps; b) there is a bias in the system and model structure (the system has the described non-linearity, the B1 block has an input filter $1/(T_F s + 1)$); c) the discrete-time system state and the measured output are burdened by additive Gaussian noises $\mathbf{v}(t)$ and $\mathbf{e}(t)$, respectively.

The Input/Output data and the resulting model fit is depicted in Fig 2. The figure shows a noise-free and noisy response of the system. Obviously, the effect of the noise in the integrating (Pressure) channel is remarkable. In order to evaluate the quality of the model multi-step ($Np = 20$) output predictions were simulated and depicted.

Several noise realizations and initial estimates (randomly sampled within the interval - see Tab 2) were considered. The parameters were normalized by their initial values in order to improve numerical properties ($\theta_n = \theta/\theta_0$) and

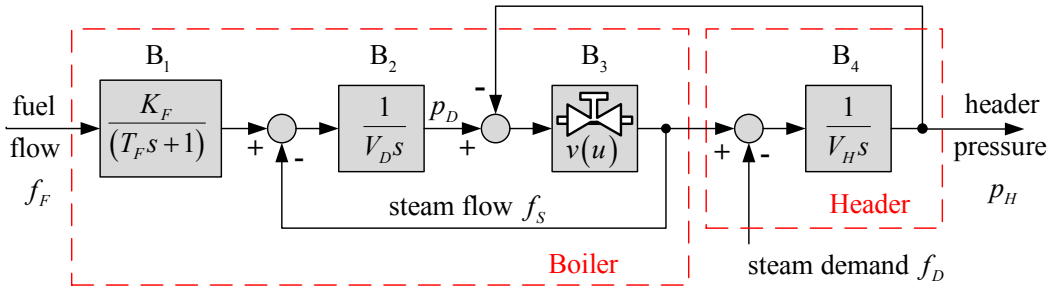

Fig. 1. Boiler-header block model.

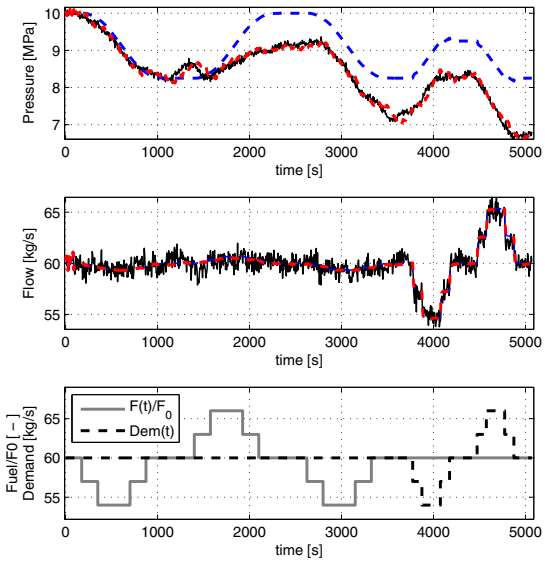

Fig. 2. Trend plot. The noise-free system output (blue), the noisy system output (black) and the 20-step-ahead predictions (red).

a standard Interior-Point algorithm [3] was chosen to solve the optimization problem (9). The algorithm takes from 10 to appx. 200 iterations in less then a couple of seconds.

6. CONCLUSION

A comprehensive description of the maximum likelihood estimation of parameters of a continuous-time, grey-box model was introduced. The approach exploits a novel algorithm of a structured exponential matrix computation that reduced the computational complexity. The practicality of this method consists in defining the model in the continuous-time domain, which is convenient for the grey-box modelling. Once the continuous-time model is defined, the discrete-time counterpart is rapidly evaluated, so that the discrete-time prediction errors and their sensitivities are computed in a numerically effective manner. The sensitivities can be used to evaluate the gradient function of the presented criterion function.

The method could be easily extended to a block-oriented identification that simplifies the model creation and modification, so that the particular sensitivities will be automatically constructed using underlying block sensitivities.

[3] Matlab Optimization tlbx. R2013b.

ACKNOWLEDGEMENTS

This work was partially supported by the grant P103 / 11 / 1353 of the Czech Science Foundation.

REFERENCES

Bohlin, T. (2006). *Practical Grey-box Process Identification*. Springer-Verlag London Limited.

Fletcher, R. (1981). *Practical Methods of Optimization: Vol. 2: Constrained Optimization*. John Wiley & sons, INC., One Wiley Dr., Somerset, N. J. 08873.

Higham, N.J. (2005). The scaling and squaring method for the matrix exponential revisited. *SIAM Journal on Matrix Analysis and Applications*, 26(4), 1179–1193.

Johansen, T.A. (1996). Identification of non-linear systems using empirical data and prior knowledgean optimization approach. *Automatica*, 32(3), 337–356.

Kristensen, N.R., Madsen, H., and Jørgensen, S.B. (2004). Parameter estimation in stochastic grey-box models. *Automatica*, 40(2), 225–237.

Ljung, L. (1999). *System Identification: Theory for the User (2nd Edition)*. Prentice Hall.

Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2), 431–441.

Matisko, P. and Havlena, V. (2013). Noise covariance estimation for kalman filter tuning using bayesian approach and monte carlo. *International Journal of Adaptive Control and Signal Processing*, 27(11), 957–973.

Moler, C. and Van Loan, C. (2003). Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM review*, 45(1), 3–49.

Trnka, P., Sturk, C., Sandberg, H., Havlena, V., and Rehor, J. (2013). Structured model order reduction of parallel models in feedback. *Control Systems Technology, IEEE Transactions on*, 21(3), 739–752. doi: 10.1109/TCST.2012.2192735.

Tulleken, H. (1993). Grey-box modelling and identification using physical knowledge and bayesian techniques. *Automatica*, 29(2), 285–308.

Young, P.C., Garnier, H., and Gilson, M. (2008). Refined instrumental variable identification of continuous-time hybrid box-jenkins models. In *Identification of Continuous-Time Models from Sampled Data*, 91–131. Springer.