Journal of Physics: Conference Series **659** (2015) 012057 doi:10.1088/1742-6596/659/1/012057

Identification of State and Measurement Noise Covariance Matrices using Nonlinear Estimation Framework

Oliver Kost, Ondřej Straka, and Jindřich Duník

European Centre of Excellence - NTIS & Department of Cybernetics
Faculty of Applied Sciences, University of West Bohemia
Univerzitní 8, 306 14 Pilsen, Czech Republic

E-mail: kost@ntis.zcu.cz, straka30@kky.zcu.cz, dunikj@kky.zcu.cz

Abstract. The paper deals with identification of the noise covariance matrices affecting the linear system described by the state-space model. In particular, the stress is laid on the autocovariance least-squares method which belongs into to the class of the correlation methods. The autocovariance least-squares method is revised for a general linear stochastic dynamic system and is implemented within the publicly available MATLAB toolbox Nonlinear Estimation Framework. The toolbox then offers except of a large set of state estimation algorithms for prediction, filtering, and smoothing, the integrated easy-to-use method for the identification of the noise covariance matrices. The implemented method is tested by a thorough Monte-Carlo simulation for various user-defined options of the implemented method.

Keywords: State estimation; Identification; Noise covariance matrices; State-space models; Software toolbox; Implementation

1. Introduction

State estimate plays a key role in many applications including various signal processors, target tracking, satellite navigation, fault detection, and adaptive and optimal control problems. It constitutes an essential part of any decision-making process. Estimation of discrete-time stochastic dynamic systems is very important field of study which, for the past seventy years, has been considerably developed. In the beginning the development was focused on linear dynamic systems and it has been further developed until today with the stress on the state estimation of nonlinear stochastic dynamic systems.

Up to now many state estimation, both local and global[1], algorithms have been proposed. The algorithms differ in both the assumptions tied with the system description and in the expected performance and computational complexity of algorithms affected by a specification of one or more user-defined estimator parameters. Whereas the system description is given, selection of the estimation algorithm and description of all its parameters a key designer task. Often, the process of selection of a suitable estimator for a given task has to be supported by a simulation study for a given set-up. To facilitate the simulation study several toolboxes and software packages have been introduced within MATLAB, such as KalmTool [1], EKF/UKF Toolbox [2], and Nonlinear Estimation Framework (NEF) [3]. The first two packages provide user with implemented algorithms of certain local estimators in stand-alone files. They are advantageous for an experienced user who want only estimation algorithms. Contrary to this, the NEF is fully object-oriented package providing the user with a collection of mutually

[1] The local estimation algorithms provide estimates in the form of conditional mean and covariance matrix. The global estimation algorithms provide estimates in the form of conditional probability density functions.

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

linked functions forming an easy-to-use a user interface. The NEF offers both the local and global estimation algorithms.

The state estimators, however, require knowledge of the whole system; not only the deterministic part of the model is required, but also the stochastic part. The deterministic part of the model is mostly based on various laws of physics, mathematics, kinematics, etc., whereas the description of the noise statistics might questionable or even unknown in many cases. Incorrect description of the noise statistics may cause significant worsening of estimation or control quality or even divergence of the underlying algorithm output.

Therefore, along with the estimation algorithms development, the methods for identification of the covariance matrices of the noises have been designed. Methods for identification of the noises covariance matrices can be divided into several categories; correlation methods [4, 5, 6, 7, 8], Bayesian estimation methods [9], maximum likelihood estimation methods [10], covariance matching methods [11], methods based on the minimax approach [12], subspace methods [13], prediction error methods [14], the Kalman filter working as a parameter estimator [15] or methods tied with variational Bayesian approximation [16]. Besides the noise covariance matrix estimation methods, alternative approach directly estimating the gain of a linear estimator have been developed as well [4, 17, 18]. A characterisation of the methods with their assumptions, properties and limitations can be found in e.g., [19, 12, 6, 20, 8]. But contrary the state estimation methods, there is no publicly available toolbox offering the algorithms for the noise covariance matrices identification.

The goal of the paper is to describe extension of the NEF toolbox with the correlation method for the noise covariance matrices identification. Through this linking the user can get the estimate of the state of the discrete-time linear stochastic time-invariant system only from knowledge of the deterministic part of model and control and measurement sequences.

The rest of the paper is organized as follows. Section 2 provides system description. In Section 3 description of the state estimation methods and state estimation using the toolbox NEF is introduced. Section 4 is devoted to the noise covariance matrices estimation and implementation of the method in the toolbox. Numerical illustrations and conclusions are given in Sections 5 and 6, respectively.

2. System description

Let a discrete-time linear dynamic stochastic system be considered

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{N}\mathbf{u}_k + \mathbf{M}\mathbf{w}_k, k = 0, 1, 2, \dots T, \tag{1}$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{O}\mathbf{v}_k, k = 0, 1, 2, \dots T, \tag{2}$$

where the vectors $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{z}_k \in \mathbb{R}^p$ and $\mathbf{u}_k \in \mathbb{R}^m$ represent the system state, measurement and the input of the system at the time instant k, respectively. The matrix $\mathbf{F} \in \mathbb{R}^{n\times n}$ is system matrix, $\mathbf{H} \in \mathbb{R}^{n\times n}$ is the measurement matrix, $\mathbf{N} \in \mathbb{R}^{n\times m}$ is the input matrix, $\mathbf{M} \in \mathbb{R}^{n\times n}$ is the state noise distribution matrix, and $\mathbf{O} \in \mathbb{R}^{p\times p}$ is the measurement noise distribution matrix. The initial state of the system $\mathbf{x}_0$ has the mean $\bar{\mathbf{x}}_0$ and the covariance matrix $\mathbf{P}_0$. The vectors $\mathbf{w}_k \in \mathbb{R}^n$ and $\mathbf{v}_k \in \mathbb{R}^p$ are the noises in the state and measurement equations, which are supposed to be zero-mean and white with the covariance matrices $\mathbf{Q} \in \mathbb{R}^{n\times n}$ and $\mathbf{R} \in \mathbb{R}^{p\times p}$, respectively. State and measurement equations are assumed to be mutually independent and independent of the initial state.

3. State estimation using NEF toolbox

In this section, state estimation and the toolbox NEF is introduced.

3.1. State Estimation

A general solution to the recursive state estimation problem is given by the Bayesian recursive relations (BRRs). The BRRs provide probability density functions (PDFs) of the state conditioned by the

measurements representing a full description of the state. The BRRs are given by

$$p(\mathbf{x}_k|\mathbf{z}_k;\mathbf{u}_{k-1}) = \frac{p(\mathbf{x}_k|\mathbf{z}_{k-1};\mathbf{u}_{k-1})p(\mathbf{z}_k|\mathbf{x}_k)}{\int p(\mathbf{x}_k|\mathbf{z}_{k-1};\mathbf{u}_{k-1})p(\mathbf{z}_k|\mathbf{x}_k)d\mathbf{x}_k}, \tag{3}$$

$$p(\mathbf{x}_k|\mathbf{z}_{k-1};\mathbf{u}_{k-1}) = \int p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1};\mathbf{u}_{k-2})p(\mathbf{x}_k|\mathbf{x}_{k-1};\mathbf{u}_{k-1})d\mathbf{x}_{k-1}, \tag{4}$$

where $p(\mathbf{x}_k|\mathbf{z}_k;\mathbf{u}_{k-1})$ is the filtering PDF, $p(\mathbf{x}_k|\mathbf{z}_{k-1};\mathbf{u}_{k-1})$ is the one-step predictive PDF, $p(\mathbf{x}_k|\mathbf{x}_{k-1};\mathbf{u}_{k-1})$ is the transition PDF stemming from (1), and $p(\mathbf{z}_k|\mathbf{x}_k)$ is the measurement PDF stemming from (2).

The closed-form solution to the BRRs is available only for a few special cases among which the linear Gaussian system plays an important role and the solution then leads to the Kalman filter (KF). In other cases, an approximate solution has to be used. The approximate methods can be divided with respect to the validity of the resulting estimates into two groups; the local methods and the global methods. The local methods are based on such approximations allowing the BRRs solution analogous to the KF design even for nonlinear systems. The approximations involve the Gaussian approximation of the conditional PDFs and system description approximations. The local methods are represented by the extended Kalman filter (EKF), unscented Kalman filter (UKF), divided difference filters (DDFs), or the stochastic integration filter (SIF). Contrary to the local methods, the global methods are often based on a numerical solution to the BRRs resulting in an approximate (but in general non-Gaussian) conditional PDFs within the almost whole state space. Within this group, the particle filter (PF), Gaussian sum filter (GSF), or the point-mass method can be mentioned.

3.2. Nonlinear estimation framework toolbox

The NEF[2] is a software toolbox for MATLAB, devoted to the state estimation of generally nonlinear discrete-time stochastic systems [3]. The NEF contains various local and global algorithms for filtering, prediction, and smoothing. Toolbox is based on mutually linked classes, which are divided into the following categories:

- System description,
- Random variable description,
- Function definition,
- Estimator definition,
- Performance evaluation.

The *System description* allows specification of the system model in a probabilistic or a structural fashion. The *Random variable description* (RV) enables description of the characteristics of RVs related to the model or the estimate. The following PDFs are supported: uniform, beta, gamma, Gaussian, Gaussian sum, and empirical. The *Function definition* enables description of equations in the state and measurement equations or a PDFs of RVs in the case of the probabilistic system description. The *Estimator definition* contains various local and global estimation algorithms as the KF, EKF, UKF, DDFs, PF, GSF, the ensemble Kalman filter, etc. The category *Performance evaluation* enables calculation of various performance metrics related to the estimate quality (mean square error (MSE), non-credibility index [21], etc.). The NEF thus represents a powerful and complex tool for the state estimation of the nonlinear systems. In this paper, the linear systems are of interests, therefore the NEF is illustrated via the KF design and application.

In the first step, the system is modelled. The state and measurement equations are created by the class `nefLinFunction` designed for description of the linear equations (1) and (2) as:

[2] The toolbox is publicly available at ***http://nft.kky.zcu.cz/nef***.

```
f = nefLinFunction(F,N,M);
h = nefLinFunction(H,[],O);
```

Then, the state and measurement noises are modeled using the class `nefGaussianRV` designed for description of the noises $\mathbf{w}$, $\mathbf{v}$ and the initial state $\mathbf{x}_0$ and are created by the commands:

```
w = nefGaussianRV(zeros(n,1),Q);
v = nefGaussianRV(zeros(n,1),R);
x0 = nefGaussianRV(x0mean,P0);
```

where the term `zeros(n,1)` means a zero vector of dimension $n \times 1$, `x0mean` and `P0` are the mean and covariance matrix of the initial state, respectively. The complete model of the system is created by the class `nefEqSystem` as:

```
myModel = nefEqSystem(f,h,w,v,x0);
```

In the second step, the estimator is created. The state of the linear system is estimated by the KF. The KF for the model is created by the class `nefKalman` as:

```
myKalman = nefKalman(myModel);
```

Having the model and estimator defined, the state estimation for a sequence of measurement data $\mathbf{z}$ and input data $\mathbf{u}$ is realized by a command:

```
[xEst,xPred] = myKalman.estimate(z,u);
```

The variable `xEst` is a required filtering estimate of the state and `xPred` is the predictive estimate of the state. The estimates are in the form of the Gaussian PDFs. The mean and covariance matrices of predictive estimate can be calculate by:

```
xPredMean = evalMean(xPred);
xPredVar = evalVariance(xPred);
```

The NEF can be also used for data simulation using a command:

```
[z,xTrue] = system.simulate(T,u);
```

where `T` is the simulation horizon and `xTrue` is the true state of the system. Further details of the NEF can be found in [3].

4. Extension of NEF toolbox with noise covariance matrices identification block

The estimation techniques implemented in the NEF toolbox require the complete description of the system i.e., the known model of the deterministic part (i.e., the matrices $\mathbf{F}$, $\mathbf{N}$, $\mathbf{M}$, $\mathbf{H}$, $\mathbf{O}$) and of the stochastic part (i.e., the matrices $\mathbf{Q}$, $\mathbf{R}$). However, the knowledge of the stochastic part might be the limiting assumption in the estimator design. Therefore, from seventies various techniques for estimation of the properties of the noises affecting the system have been proposed [9, 10, 12, 16] prevailingly for the linear systems. Among them, the correlation methods have attracted significant attention [4, 5, 6, 7, 8].

4.1. Noise covariance matrices identification by correlation method

In contrast to other methods, the correlation methods has several quite considerable positive including characteristics the unbiased estimate and small computing time even for high-dimensional systems. The methods are based on an analysis the innovation sequence of a "non-optimal" linear estimator and were originally derived in [4] and [5] in the form of the three-step procedure for estimation of the noise covariance matrices. The three-step procedure was reformulated in one-step method providing the unbiased estimate of the noise covariance matrices in [6, 7, 22, 8] assuming a discrete-time linear time-invariant system. The method is generally called the autocovariance least-squares (ALS) method[3].

[3] Note that the ALS method has been proposed also for nonlinear systems. Then, the ALS method takes an advantage of a linearised predictor following the structure of the EKF [23], [24].

4.2. Autocovariance least-square method

The ALS method, proposed for the identification of the noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ of linear systems, is based on an analysis of the second-order statistics of an innovation sequence produced by a linear (one-step) predictor [6, 22] of the form

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}\mathbf{e}_k, \tag{5}$$

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F}\hat{\mathbf{x}}_{k|k} + \mathbf{N}\mathbf{u}_k, \tag{6}$$

where $\hat{\mathbf{x}}_{k+1|k}$ is the one-step state prediction, $\hat{\mathbf{x}}_{k|k}$ is the filtering state estimate, $\mathbf{K}$ is a user-defined predictor gain, and $\mathbf{e}_k$ is the measurement prediction error (innovation) given by

$$\mathbf{e}_k = \mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}, \forall k. \tag{7}$$

The estimation error of the predictor

$$\boldsymbol{\epsilon}_k = \mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1} \tag{8}$$

evolves according to

$$\boldsymbol{\epsilon}_{k+1} = (\mathbf{F} - \mathbf{FKH})\boldsymbol{\epsilon}_k + [\mathbf{M}, -\mathbf{FKO}] \left[\begin{smallmatrix} \mathbf{w}_k \\ \mathbf{v}_k \end{smallmatrix}\right] = \bar{\mathbf{F}}\boldsymbol{\epsilon}_k + \mathbf{G}\bar{\mathbf{w}}_k. \tag{9}$$

The innovation $\mathbf{e}_k$ is related to the estimation error $\boldsymbol{\epsilon}_k$ by

$$\mathbf{e}_k = \mathbf{H}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) + \mathbf{O}\mathbf{v}_k = \mathbf{H}\boldsymbol{\epsilon}_k + \mathbf{O}\mathbf{v}_k. \tag{10}$$

The innovation sequence is, after reaching the steady-state, a zero-mean stochastic process with the autocovariance matrices defined as

$$\mathbf{C}_{e,0} = E[\mathbf{e}_k\mathbf{e}_k^T] = \mathbf{H}\mathbf{P}_\epsilon\mathbf{H}^T + \mathbf{ORO}^T, \tag{11}$$

$$\mathbf{C}_{e,j} = E[\mathbf{e}_{k+j}\mathbf{e}_k^T] = \mathbf{H}\bar{\mathbf{F}}^j\mathbf{P}_\epsilon\mathbf{H}^T - \mathbf{H}\bar{\mathbf{F}}^{j-1}\mathbf{FKORO}^T, \tag{12}$$

where $j = 1, 2, \ldots, L-1$ and L is the user-defined parameter defining the maximum lag (and thus, the number of equations used in the ALS). Assuming that the $\bar{\mathbf{F}}$ is stable, the matrix $\mathbf{P}_\epsilon$ is the steady-state covariance matrix of the state prediction (estimation) error (9) given by

$$(\mathbf{P}_\epsilon)_s = (\mathbf{I}_{n_x^2} - \bar{\mathbf{F}} \otimes \bar{\mathbf{F}}^T)^{-1}(\mathbf{G} \otimes \mathbf{G}^T)\boldsymbol{\Sigma}_s, \tag{13}$$

where the symbol $\otimes$ stands for the Kronecker product and the notation $\mathbf{A}_s$ means the column-wise stacking of the matrix $\mathbf{A}$ into a vector [25]. Note that the covariance matrix $\mathbf{P}_\epsilon$ is a solution to the Lyapunov equation

$$\mathbf{P}_\epsilon = \bar{\mathbf{F}}\mathbf{P}_\epsilon\bar{\mathbf{F}}^T + \mathbf{G}\boldsymbol{\Sigma}\mathbf{G}^T \tag{14}$$

with $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{smallmatrix}\right]$ stemming from (9).

The relations (11) and (12) can be rewritten in a form suitable for the least-squares method [6, 7, 22, 8]

$$\mathcal{A}\boldsymbol{\theta} = \mathbf{b}, \tag{15}$$

where the design matrix $\mathcal{A}$ is defined as

$$\mathcal{A} = [\mathbf{D}(\mathbf{M} \otimes \mathbf{M}), \mathbf{D}(\mathbf{FKO} \otimes \mathbf{FKO}) + (\mathbf{O} \otimes \boldsymbol{\Gamma})], \tag{16}$$

$$\mathbf{D} = (\mathbf{H} \otimes \mathcal{O})(\mathbf{I}_{n^2} - \bar{\mathbf{F}} \otimes \bar{\mathbf{F}})^{-1}, \tag{17}$$

with the observability matrix $\mathcal{O}$ and matrix $\mathbf{\Gamma}$ given as

$$\mathcal{O} = [\mathbf{H}^T, (\mathbf{H}\bar{\mathbf{F}})^T, \ldots, (\mathbf{H}\bar{\mathbf{F}}^{L-1})^T]^T, \tag{18}$$

$$\mathbf{\Gamma} = [\mathbf{O}, -(\mathbf{HFKO})^T, \ldots, -(\mathbf{H}\bar{\mathbf{F}}^{L-2}\mathbf{FKO})^T]^T. \tag{19}$$

The vectors of parameters and dependent variables are of the form

$$\boldsymbol{\theta} = [\mathbf{Q}_s^T, \mathbf{R}_s^T]^T, \mathbf{b} = (\mathcal{C}_e(L))_s \tag{20}$$

with

$$\mathcal{C}_e(L) = [\mathbf{C}_{e,0}^T, \mathbf{C}_{e,1}^T, \ldots, \mathbf{C}_{e,L-1}^T]^T. \tag{21}$$

The unbiased estimate of the autocovariance matrices [6], and subsequently of the vector $\mathbf{b}$, computed on the basis of the innovation sequence (7), is computed as

$$\hat{\mathbf{C}}_{e,j} = \frac{1}{T-j-\kappa} \sum_{k=\kappa}^{T-j} \mathbf{e}_{k+j}\mathbf{e}_k^T, j = 0, 1, \ldots, L-1, \tag{22}$$

$$\hat{\mathcal{C}}_e(L) = [\hat{\mathbf{C}}_{e,0}^T, \hat{\mathbf{C}}_{e,1}^T, \ldots, \hat{\mathbf{C}}_{e,L-1}^T]^T, \tag{23}$$

$$\hat{\mathbf{b}} = \left(\hat{\mathcal{C}}_e(L)\right)_s, \tag{24}$$

where κ defines the time instant in which the effect of the initial conditions of the predictor (9) on the actual state estimate is negligible.

Then, the unbiased (ALS) estimate of the vector of parameters (composed by the elements of the unknown noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$) in the least-squares sense is given by

$$\hat{\boldsymbol{\theta}} = (\mathcal{A}^T\mathcal{A})^{-1}\mathcal{A}^T\hat{\mathbf{b}} = \mathcal{A}^\dagger\hat{\mathbf{b}}. \tag{25}$$

Note that $\mathbf{K}$ is, in the ALS method, considered as a user-defined parameter. Optimal gain can not be computed as the matrices $\mathbf{Q}$ and $\mathbf{R}$ are unknown and are being estimated. It means the estimator (5) and (6) is not optimal in the MSE sense nor the innovation sequence is a white process. In principle, any predictor gain $\mathbf{K}$ can be selected if it results in the stable state prediction error transition matrix $\bar{\mathbf{F}}$ appearing in (9). A reasonable choice of $\mathbf{K}$, fulfilling the stability condition, is given by

$$\mathbf{K} = (\bar{\mathbf{P}}\mathbf{H}^T)(\mathbf{H}\bar{\mathbf{P}}\mathbf{H}^T + \mathbf{R}_A)^{-1}. \tag{26}$$

where $\bar{\mathbf{P}}$ is the solution to the Riccati equation

$$\bar{\mathbf{P}} = \mathbf{F}\bar{\mathbf{P}}\mathbf{F}^T - \mathbf{F}\bar{\mathbf{P}}\mathbf{H}^T(\mathbf{H}\bar{\mathbf{P}}\mathbf{H}^T + \mathbf{R}_A)^{-1}\mathbf{H}\bar{\mathbf{P}}\mathbf{F}^T + \mathbf{Q}_A. \tag{27}$$

and $\mathbf{Q}_A$ and $\mathbf{R}_A$ are arbitrary user-defined positive semi-definite matrices [26].

It can be seen that state estimation (non-optimal in the MSE sense) by a linear estimator, providing innovation sequence, is the inherent part of the ALS method. The NEF toolbox is the ideal package for such a task.

4.3. Implementation of ALS method to NEF

The basic idea is utilize methods and objects of the NEF and extend them with the ALS method identifying the noise covariance matrices from the knowledge of the deterministic part of the system description, control sequence, and measurements of the system. By identification of the noises covariance matrices, the user gets complete description of the system, which can subsequently be used in the NEF for the optimal state estimation of a linear system.

Specifically, the NEF was extended with category *System identification* having two classes. The classes have been designed to be consistent with and utilizing other components of the NEF toolbox.

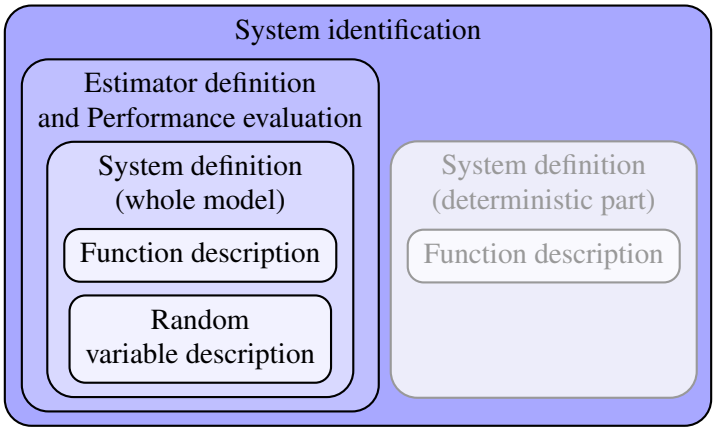


Figure 1. Structure of the NEF extended with the Identification class (including ALS method).

4.3.1. Identification class - The first class is the parent class `nefIdentificaton`. The class contains the method `gainCalculation`, which can calculate the current estimator gain according to (26). The gain is computed by the following command:

```
gain = gainCalculation(f,h,wApriori,vApriori);
```

where `f` and `h` are the objects of the class `nefLinFunction` as defined in Section 3.2 and `wApriori` and `vApriori` are instances of the class `nefGaussianRV` defined as:

```
wApriori = nefGaussianRV(zeros(n,1),QA);
vApriori = nefGaussianRV(zeros(p,1),RA);
```

with `QA` and `RA` being the user-defined matrices defining the predictor gain (26). The class design supports fast extension with other supporting methods for the indetification.

4.3.2. ALS class - The second class is `nefALS` class which contains the implemented method ALS detailed in Section 4.2. Class constructor is `nefALS` which can create object:

```
myALS = nefALS(f,h,x0,varargin);
```

where `x0` is the object of the class `nefGaussianRV` as defined in Section 3.2 and `myALS` is the new object of the class `nefALS`. The `varargin` is the MATLAB notation for input parameters consisting of two parts. The first part is the name and the second part is the specified value. The `varargin` of `nefALS` is currently the following:

- name `'numberCovEq'` with value:
  - an integer number in $\langle 2, \infty)$
    - the number coincides with the variable L as defined in (12), (20).
    - default value is $\lceil \frac{n+p}{p} \rceil$ as discussed in Appendix.
- name `'omittedInitPeriod'` with value:
  - an integer number in $\langle 0, 100\rangle$
    - defines percentage period of the time T in which the initial conditions of the predictor (9) are significant; after that period, the predictor is assumed to be in the steady-state and the innovation sequence independent of the initial condition.
    - default value is 50 resulting in $\kappa = \lfloor \frac{T}{2} \rfloor$ in (22).

The main method of the class is `identifyQR`, which identifies the noise covariance matrices. The noise covariance matrices are identified by the following command:

```
[QEst,REst] = myALS.identifyQR(z,u,varargin);
```

where `QEst` is the identified state noise covariance matrix $\mathbf{Q}$, `REst` is the identified measurement noise covariance matrix $\mathbf{R}$, `z` is the vector of measurements, and `u` is the vector of control input. The possible input parameters (`varargin`) of the method `identifyQR` are these:

- name `'ALSAprioriSetting'` with values:
  - `'fast'` - sets gain $\mathbf{K}$ to zero (fast computation because of simplifications in (5-16)),
  - `'automatic'` - selects optimal $\mathbf{K}$ to minimize $\mathbf{Q}$, $\mathbf{R}$ estimates variance (exact identification),
  - `aprioriQR` - computes $\mathbf{K}$ for the user-defined matrices $\mathbf{Q}_A$ and $\mathbf{R}_A$ (`aprioriQR` is a structure containing the user-define matrices),
- name `'knownQ'` with values:
  - `elemensQ` defining the known elements of the noise covariance matrix $\mathbf{Q}$ (if any),
- name `'knownR'` with values:
  - `elemensR` defining the known elements of the noise covariance matrix $\mathbf{R}$ (if any).

Detailed example of possible parameters is in following section.

The class `nefALS` contains a set of other auxiliary methods. The method `calcInnovationCov` calculates the innovation sequence covariance matrices (11), (12) using a command:

```
innovCov = calcInnovationCov(myALS,xPredMean,z);
```

where `innovCov` is a set of (cross-)covariance matrices of the innovation and `xPredMean` is a vector of predictive mean values as defined in Section 3.2.

The method `calculateLSDesignMatrix` calculates the matrix $\mathcal{A}$ defined in (16). The method is called by the following command:

```
LSDM = calculateLSDesignMatrix(myALS,K);
```

where `K` is the predictor gain $\mathbf{K}$ defined in (26).

The method `reductionLSDesignMatrix` reduces duplicate columns in the matrix $\mathcal{A}$ be cause of symetry of estimate covariance matrices. By the reduction the calculation is faster. The method is called by:

```
reducedLSDM = reductionLSDesignMatrix(myALS,LSDM);
```

The last important method of the class `nefALS` is the method `minimalizationCrit` which finds a gain $\mathbf{K}$ resulting in alow variance of the $\mathbf{Q}$, $\mathbf{R}$ estimate by the following command:

```
[wALS,vALS,KALS] = minimalizationCrit(myALS);
```

where `KALS` is the gain used by the predictor (9), `wALS` and `vALS` are the `nefGaussian` object noises which were used for calculation of `KALS`. This method is employed if the input parameter `'ALSAprioriSetting'` of the method `identifyQR` is set to `'automatic'`.

All the classes and methods monitor the correctness of the input parameters including their dimensions and allowed settings. Moreover, the class `nefALS` also assesses the identifiability of the considered system with respect to the actual set-up. If any input condition is violated, the user is notified and a possible solution is proposed. For completeness, the overall illustration of the updated NEF toolbox with the identification block is given in Figure 1.

Use of the NEF toolbox for the noise covariance matrices identification by the implemented ALS method with different settings is illustrated in the next section.

5. Use case of ALS method in NEF and numerical illustration

Through this section, the linear model (1) and (2) with the following matrices:

$$n = 1, m = 1, p = 2, \tag{28}$$

$$F = 0.9, N = 0.5, M = 2, Q = 1.5, \tag{29}$$

$$\mathbf{H} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \mathbf{O} = \begin{bmatrix} 2 & 0.1 \\ 0.1 & 1 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 3 \end{bmatrix}, \tag{30}$$

$$\bar{\mathbf{x}}_0 = 0, P_0 = 1, \tag{31}$$

is used. The matrices Q, $\mathbf{R}$ are assumed to be unknown (except of the system output simulation) and thus estimated.

5.1. Model of deterministic part of system

In the first step, the deterministic part of the system is modelled for the purposes of the ALS method by a sequence of commands:

```
f = nefLinFunction(0.9,0.5,2);
H = [1;0.5];
O = [2 0.1;0.1 1];
h = nefLinFunction(H,[],O);
x0 = nefGaussianRV(0,1);
```

5.2. System simulation

The second (optional) step is related to the measured data generation. To simulate the system, the true characteristics of the state and measurement noises has to be defined by

```
w = nefGaussianRV(0,1.5);
v = nefGaussianRV([0;0],[2 0.5;0.5 3]);
```

Then, the object representing the complete model (28)–(31) is created by:

```
myModel = nefEqSystem(f,h,w,v,x0);
```

and the object is simulated by the NEF toolbox by commands:

```
T = 2e3;
u = [sin(0:pi/(T-1):pi)];
z = myModel.simulate(T,u);
```

where `T` is the number of simulation steps, `z` is the outputted measurement sequence, and `u` is the control sequence.

It should be mentioned that the simulation part can be skipped if the measurement and control sequences are available. Also note that the noises need not be only of the Gaussian PDFs (`nefGaussianRV`), but other RVs supported by the NEF e.g., the Gaussian sum PDF (`nefGaussianSumRV`) or the uniform PDF (`nefUniformRV`) can be selected, as well. The ALS method is not, in principle, based on the Gaussian assumption.

5.3. Noise covariance matrices identification

The third and last step is the identification of the noise covariance matrices Q and $\mathbf{R}$. Noise covariance matrices identification step starts with the creation of the `nefALS` class object:

```
myALS = nefALS(f,h,x0,'numberCovEq',3);
```

Having the simulated data `z`, the matrices Q and $\mathbf{R}$ can be identified using the method `identifyQR`. Four identification experiments[4] each with different setting of the ALS method, are considered:

Example 1 – estimate of *all* elements of Q and $\mathbf{R}$ with `'fast'` setting of the `nefALS`

```
[wEst,vEst] = myALS.identify(z,u,'ALSAprioriSetting','fast');
```

Example 2 – estimate of *all* elements of Q and $\mathbf{R}$ with `'automatic'` setting leading to the estimates with low variance

```
[wEst,vEst] = myALS.identify(z,u,'ALSAprioriSetting','automatic');
```

Example 3 – estimate of *all* elements of Q and $\mathbf{R}$ with predictor gain $\mathbf{K}$ given by the user defined matrices Q_A and $\mathbf{R}_A$

```
aprioriMatrix.Q=10;
aprioriMatrix.R=[6 0,0 6];
[wEst,vEst] = myALS.identify(z,u,'ALSAprioriSetting',aprioriMatrix);
```

Example 4 – estimation of a *subset* of elements of Q and $\mathbf{R}$ (just one element of $\mathbf{R}$) with `'fast'` setting of the `nefALS`

```
knownR =[2 0.5;0.5 NaN];
[wEst,vEst]=ALS.identify(z,u,'knownR',knownR,...
                         'ALSAprioriSetting','fast');
```

All examples were simulated using $M = 10^4$ Monte-Carlo (MC) simulations.

5.4. Identification results

The results of the identification experiments for all examples are summarized in Table 1. The results are in the form of the sample mean value and standard deviation (STD) of the estimates of the noise covariance matrices elements over all MC simulations. The table also gives the average computational time over one MC simulations.

Noise covarinace matrices		Example 1	2	3	4
	True	Estimated [mean±standard deviation]			
Q	1.5	1.5037 ± 0.2446	1.5184 ± 0.1671	1.5213 ± 0.1865	1.5012 ± 0.2313
$R_{1,1}$	2	1.9982 ± 0.2202	1.9944 ± 0.1559	1.9918 ± 0.1669	–
$R_{1,2}$	0.5	0.4980 ± 0.2168	0.4937 ± 0.1447	0.4940 ± 0.1505	–
$R_{2,2}$	3	2.9964 ± 0.3254	2.9936 ± 0.1924	2.9926 ± 0.19687	2.9992 ± 0.3795
Identification time per simulation [sec:msec]		0:857	1:714	1:502	0:857

Table 1. ALS estimation statistics summary for all examples.

The results indicate that the ALS method provides unbiased estimates for an arbitrary stable gain of the linear predictor. The ALS method with the choice `'automatic'` selects such a gain which results in the estimates with the low covariance matrix (represented by the low STD). On the other hand, the choice `'fast'` simplifies the ALS method relations resulting in approximately 50 per cent reduction of

[4] The example are also publicly available at ***http://nft.kky.zcu.cz/nef***.

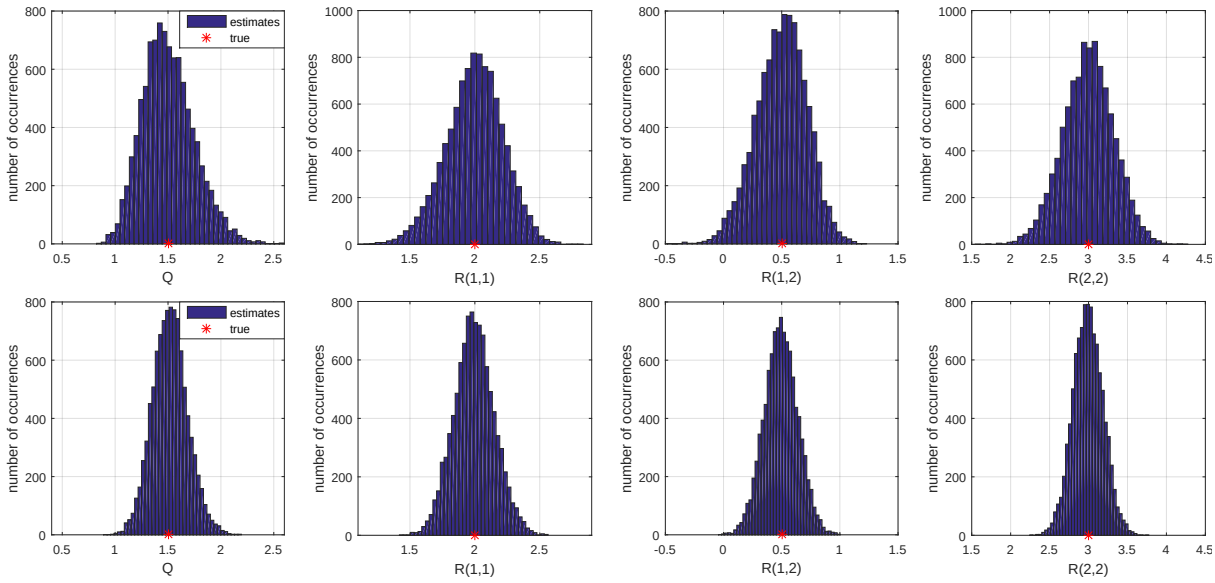


Figure 2. ALS estimates by NEF for Example 1 (upper row) and Example 2 (lower row)

the computational time. The possibility of the gain specification by the user (according to (26)) offers the trade-off between the computational complexity and effectiveness.

For completeness, the histograms of the identification experiments for Examples 1 and 2 are shown in Figure 2. The histograms illustrate the lower variance of the noise covariance matrices elements estimates of the ALS method in the set-up of Example 2 (lower row).

6. Conclusion

The paper was devoted to the identification of the noise covariance matrices of a linear stochastic dynamic system described by the state-space model. The identification method, the autocovariance least-squares method, was revised for a general state-space model and implemented within the Nonlinear Estimation Framework. Implementation of the method and the user-defined settings were discussed in detail and illustrated in examples. Usage of the toolbox and the performance of the implemented identification method were illustrated in numerical Monte-Carlo study. The introduced Nonlinear Estimation Framework is the only publicly available MATLAB toolbox offering a set of state estimation algorithms combined with the method for the system noise covariance matrices identification. The toolbox is available at ***http://nft.kky.zcu.cz/nef***.

Appendix: Number of ALS identifiable parameters

According to [22], it can be shown that the maximum number of the ALS identifiable parameters is

$$N_{maxIdentParam} = np + p^2, \text{ if } n > p. \tag{32}$$

However, with respect to the structure of $\mathbf{b}$ in (20) and symmetry of $\mathbf{C}e, 0$ (11), the ALS estimate is based on

$$N_{independentEqs} = \frac{p(p+1)}{2} + p^2(L-1) \tag{33}$$

independent equations. Then, comparing (32) with (33), the minimum number of required equations is

$$L_{min} = \lceil \frac{p+n}{p} \rceil. \tag{34}$$

Acknowledgement
This work was supported by the Czech Science Foundation, project no. GA 15-12068S.

References
[1] M. Norgaard, N. K. Poulsen, and O. Ravn, "Kalmtool for use with Matlab," in *Proceedings of the 13th IFAC Symposium on System Identification*, pp. 1490–1495, 2003. http://server.oersted.dtu.dk/personal/or/kalmtool3/.
[2] S. Sarkka and J. Hartikainen, "EKF/UKF Toolbox," 2007. http://www.lce.hut.fi/research/mm/ekfukf/.
[3] O. Straka, M. Flidr, J. Dunik, and M. Simandl, "A software framework and tool for nonlinear state estimation," in *Proceedings of the 15th IFAC Symposium on System Identification*, (Saint-Malo, France), pp. 510–515, 2009.
[4] R. K. Mehra, "Fault-on the identification of variances and adaptive filtring," *IEEE Transactions on Automatic Control*, vol. 15, no. 2, pp. 175–184, 1970.
[5] P. R. Belanger, "Estimation of noise covariance matrices for a linear-varying stochastic process," *Automatica*, vol. 10, no. 3, pp. 267–275, 1974.
[6] B. J. Odelson, M. R. Rajamani, and J. B. Rawlings, "A new autocovariance least-squares method for estimating noise covariances," *Automatica*, vol. 42, no. 2, pp. 303–308, 2006.
[7] B. M. Akesson, J. B. Jorgensen, N. K. Poulsen, and J. S. B., "A generalized autocovariance least-squares method for Kalman filter tuning.," *Journal of Process Control*, vol. 18, no. 7–8, pp. 769–779, 2008.
[8] M. Simandl and J. Dunik, "Estimation of noise covariance matrices for periodic systems," *International Jornal of Adaptive Control and Signal Processing*, vol. 25, pp. 928–942, 2009.
[9] D. G. Lainiotis, "Optimal adaptive estimation: Structure and parameters adaptation," *IEEE Transactions on Automatic Control*, vol. 16, no. 2, pp. 160–170, 1971.
[10] R. L. Kashyap, "Maximum likelihood identification of stochastic linear systems," *IEEE Transactions on Automatic Control*, vol. 15, no. 1, pp. 25–34, 1970.
[11] K. A. Myers and B. D. Tapley, "Adaptive sequential estimation wtih unknown noise statistics," *IEEE Transactions on Automatic Control*, vol. 21, no. 8, pp. 520–523, 1976.
[12] S. Verdu and H. V. Poor, "Minimax linear observers and regulators for stochastic systems with uncertain second-order statistics," *IEEE Transactions on Automatic Control*, vol. 26, no. 6, pp. 499–511, 1984.
[13] P. van Overshee and B. de Moor, *Subspace Identification for Linear Systems: Theory-Implementation-Aplications*. Kluwer Academic Publishers, 1996.
[14] L. Ljung, *System Identification: Theory for the User*. UpperSaddle River, New Jersey, USA: Prentice-Hall, 1999.
[15] D. M. Wiberg, T. D. Powel, and D. Ljungquist, "An on-line parameter estimator for quick convergence and time-varying linear system," *IEEE Transactions on Automatic Control*, vol. 45, no. 10, pp. 1854–1863, 2000.
[16] S. Sarkka and A. Nummenmaa, "Recursive noise adaptive Kalman filtering by variational bayesian approximations," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 596–600, 2009.
[17] B. Carew and P. R. Belanger, "Identification of optimum steady-state gain for system with unknown noise covarinaces," *IEEE Transactions on Automatic Control*, vol. 18, no. 6, pp. 582–587, 1973.
[18] R. Bos, X. Bombois, and P. Van den Hof, "Desinging a Kalman filter when no noise covariance information is available," in *Proceedings of 16th IFAC world Congress*, (Prague, Czech Republic), 2005.
[19] R. K. Mehra, "Approaches to adaptive filtering," *IEEE Transactions on Automatic Control*, vol. 17, no. 10, pp. 693–698, 1972.
[20] J. Dunik, M. Simandl, and O. Straka, "Methods for estimating state and measurement noise covariance matices: Aspects and comparison," in *Proceedings of 15th IFAC Symposiun on System Identification*, vol. 25, (Saint-Malo, France), 2009.
[21] X. R. Li and Z. Zhao, "Evaluation of estimation algorithms part I: Incomprehensive measures of performance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 4, pp. 1340–1358, 2006.
[22] M. R. Ramajani and J. B. Rawlings, "Estimation of the disturbance structure from data using semedefinite programing and optimal weighthing," *Automatica*, vol. 45, no. 1, pp. 142–148, 2009.
[23] F. V. Lima, M. R. Rajamani, T. A. Soderstrom, and J. B. Rawlings, "Covariance and state estimation of weakly observable systems: Application to polymerization processes," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1249–1257, 2013.
[24] J. Duník, O. Straka, and M. Šimandl, "Estimation of noise covariance matrices for linear systems with nonlinear measurements," in *Proceedings of the 17th IFAC Symposium on System Identification*, (Beijing, China), 10 2015.
[25] J. W. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Transactions on Circuits and Systems*, vol. 25, no. 9, pp. 772–781, 1978.
[26] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.