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

To cite this article: Oliver Kost et al 2015 J. Phys.: Conf. Ser. 659 012057

View the article online for updates and enhancements.

Related content
- Nonlinear entanglement witnesses, covariance matrices and the geometry of separable states
  Otfried Gühne and Norbert Lütkenhaus
- Including parameter dependence in the data and covariance for cosmological inference
  Martin White and Nikhil Padmanabhan
- Adaptive Beamforming with Inadequate Snapshots
  Jing YU and Yaan LI
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.
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

\[ x_{k+1} = Fx_k + Nu_k + Mw_k, \quad k = 0, 1, 2, \ldots T, \]  
\[ z_k = Hx_k + Ov_k, \quad k = 0, 1, 2, \ldots T, \]  

where the vectors \( x_k \in \mathbb{R}^n \), \( z_k \in \mathbb{R}^p \) and \( u_k \in \mathbb{R}^m \) represent the system state, measurement and the input of the system at the time instant \( k \), respectively. The matrix \( F \in \mathbb{R}^{n \times n} \) is system matrix, \( H \in \mathbb{R}^{n \times n} \) is the measurement matrix, \( N \in \mathbb{R}^{n \times m} \) is the input matrix, \( M \in \mathbb{R}^{n \times n} \) is the state noise distribution matrix, and \( O \in \mathbb{R}^{p \times p} \) is the measurement noise distribution matrix. The initial state of the system \( x_0 \) has the mean \( \bar{x}_0 \) and the covariance matrix \( P_0 \). The vectors \( w_k \in \mathbb{R}^n \) and \( v_k \in \mathbb{R}^p \) are the noises in the state and measurement equations, which are supposed to be zero-mean and white with the covariance matrices \( Q \in \mathbb{R}^{n \times n} \) and \( R \in \mathbb{R}^{p \times p} \), 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

\[
p(x_k|z_k; u_{k-1}) = \frac{p(x_k|z_{k-1}; u_{k-1})p(z_k|x_k)}{\int p(x_k|z_{k-1}; u_{k-1})p(z_k|x_k)dx_k},
\]

(3)

\[
p(x_k|z_{k-1}; u_{k-1}) = \int p(x_{k-1}|z_{k-1}; u_{k-2})p(x_k|x_{k-1}; u_{k-1})dx_{k-1},
\]

(4)

where \(p(x_k|z_k; u_{k-1})\) is the filtering PDF, \(p(x_k|z_{k-1}; u_{k-1})\) is the one-step predictive PDF, \(p(x_k|x_{k-1}; u_{k-1})\) is the transition PDF stemming from (1), and \(p(z_k|x_k)\) 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:

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 w, v and the initial state x0 and are created by the commands:

\[
\begin{align*}
  w &= \text{nefGaussianRV}(\text{zeros}(n,1),Q); \\
  v &= \text{nefGaussianRV}(\text{zeros}(n,1),R); \\
  x0 &= \text{nefGaussianRV}(\text{x0mean},P0);
\end{align*}
\]

where the term \text{zeros}(n,1) means a zero vector of dimension \( n \times 1 \), \text{x0mean} and \text{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:

\[
\text{myModel} = \text{nefEqSystem}(f,h,w,v,x0);
\]

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:

\[
\text{myKalman} = \text{nefKalman}(\text{myModel});
\]

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

\[
[\text{xEst},\text{xPred}] = \text{myKalman}.\text{estimate}(z,u);
\]

The variable \text{xEst} is a required filtering estimate of the state and \text{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:

\[
\begin{align*}
  \text{xPredMean} &= \text{evalMean}(\text{xPred}); \\
  \text{xPredVar} &= \text{evalVariance}(\text{xPred});
\end{align*}
\]

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

\[
[z,\text{xTrue}] = \text{system}.\text{simulate}(T,u);
\]

where \( T \) is the simulation horizon and \( \text{xTrue} \) 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 \( F, N, M, H, O \)) and of the
stochastic part (i.e., the matrices \( Q, R \)). 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 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 $Q$ and $R$ 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

$$
\dot{x}_{k|k} = \dot{x}_{k|k-1} + Ke_k, \\
\dot{x}_{k+1|k} = F\dot{x}_{k|k} + Nu_k,
$$

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

$$
e_k = z_k - H\dot{x}_{k|k-1}, \forall k.
$$

The estimation error of the predictor $\epsilon_k = x_k - \dot{x}_{k|k-1}$ evolves according to

$$
\epsilon_{k+1} = (F - FKH)\epsilon_k + [M, -FKO] \left[ w_k \right] = \bar{F}\epsilon_k + G\bar{w}_k.
$$

The innovation $e_k$ is related to the estimation error $\epsilon_k$ by

$$
e_k = H(x_k - \dot{x}_{k|k-1}) + Ov_k = H\epsilon_k + Ov_k.
$$

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

$$
C_{e,0} = E[e_ke_k^T] = HP_H^T + ORO^T,
$$

$$
C_{e,j} = E[e_{k+j}e_k^T] = H\bar{F}^jP_H^T - H\bar{F}^{j-1}FKORO^T,
$$

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 $\bar{F}$ is stable, the matrix $P_e$ is the steady-state covariance matrix of the state prediction (estimation) error (9) given by

$$
(P_e)_s = (I_{n^2} - \bar{F} \otimes \bar{F}^T)^{-1}(G \otimes G^T)\Sigma_e,
$$

where the symbol $\otimes$ stands for the Kronecker product and the notation $A_s$ means the column-wise stacking of the matrix $A$ into a vector [25]. Note that the covariance matrix $P_e$ is a solution to the Lyapunov equation

$$
P_e = \bar{F}P_e\bar{F}^T + G\Sigma G^T
$$

with $\Sigma = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}$ stemming from (9).

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

$$
A\theta = b,
$$

where the design matrix $A$ is defined as

$$
A = [D(M \otimes M), D(FKO \otimes FKO) + (O \otimes I)],
$$

$$
D = (H \otimes O)(I_{n^2} - \bar{F} \otimes \bar{F})^{-1},
$$

with $\otimes$ standing for the Kronecker product.
with the observability matrix \( O \) and matrix \( \Gamma \) given as

\[
O = [H^T, (HF)^T, \ldots, (HF^{L-1})^T]^T,
\]

\[
\Gamma = [O, -(HKFO)^T, \ldots, -(HK^{L-2}FO)^T]^T.
\]

The vectors of parameters and dependent variables are of the form

\[
\theta = [Q_s^T, R_s^T]^T, b = (C_e(L))^s
\]

with

\[
C_e(L) = [C_{e,0}^T, C_{e,1}^T, \ldots, C_{e,L-1}^T]^T.
\]

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

\[
\hat{C}_{e,j} = \frac{1}{T-j} \sum_{k=\kappa}^{T-j} e_{k+j}^T e_k, j = 0, 1, \ldots, L - 1,
\]

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

\[
b = (\hat{C}_e(L))^s,
\]

where \( \kappa \) 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 \( Q \) and \( R \)) in the least-squares sense is given by

\[
\hat{\theta} = (A^T A)^{-1} A^T \hat{b} = A^\dagger \hat{b}.
\]

Note that \( K \) is, in the ALS method, considered as a user-defined parameter. Optimal gain can not be computed as the matrices \( Q \) and \( R \) 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 \( K \) can be selected if it results in the stable state prediction error transition matrix \( \bar{F} \) appearing in (9). A reasonable choice of \( K \), fulfilling the stability condition, is given by

\[
K = (\bar{P}H^T)(H\bar{P}H^T + R_A)^{-1}.
\]

where \( \bar{P} \) is the solution to the Riccati equation

\[
\bar{P} = FPP^T - FPH^T (H\bar{P}H^T + R_A)^{-1} HFP^T + Q_A.
\]

and \( Q_A \) and \( R_A \) 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.
System identification

Estimator definition and Performance evaluation

System definition (whole model)

Function description

Random variable description

System definition (deterministic part)

Function description

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 nefIdentification. The class contains the method gainCalculation, which can calculate the current estimator gain according to (26). The gain is computed by the following command:

\[
\text{gain} = \text{gainCalculation}(f,h,\text{wApriori},\text{vApriori});
\]

where \(f\) and \(h\) are the objects of the class nefLinFunction as defined in Section 3.2 and \(\text{wApriori}\) and \(\text{vApriori}\) are instances of the class nefGaussianRV defined as:

\[
\text{wApriori} = \text{nefGaussianRV}(\text{zeros}(n,1),\text{QA});
\]

\[
\text{vApriori} = \text{nefGaussianRV}(\text{zeros}(p,1),\text{RA});
\]

with \(\text{QA}\) and \(\text{RA}\) being the user-defined matrices defining the predictor gain (26). The class design supports fast extension with other supporting methods for the identetification.

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:

\[
\text{myALS} = \text{nefALS}(f,h,\text{x0},\text{varargin});
\]

where \(\text{x0}\) is the object of the class nefGaussianRV as defined in Section 3.2 and \(\text{myALS}\) is the new object of the class nefALS. The \(\text{varargin}\) 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 \(\text{varargin}\) of \(\text{nefALS}\) is currently the following:

- **name ‘numberCovEq’ with value:**
  - an integer number in \((2, \infty)\)
    * 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 \((0, 100)\)
    * 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 \(\kappa = \left\lfloor \frac{T}{2} \right\rfloor\) 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 $Q$, REst is the identified measurement noise covariance matrix $R$, $z$ 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 $K$ to zero (fast computation because of simplifications in (5-16)),
  - 'automatic' - selects optimal $K$ to minimize $Q$, $R$ estimates variance (exact identification),
  - aprioriQR - computes $K$ for the user-defined matrices $Q_A$ and $R_A$ (aprioriQR is a structure containing the user-define matrices),

- **name 'knownQ'** with values:
  - elemensQ defining the known elements of the noise covariance matrix $Q$ (if any),

- **name 'knownR'** with values:
  - elemensR defining the known elements of the noise covariance matrix $R$ (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 $A$ defined in (16). The method is called by the following command:

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

where $K$ is the predictor gain defined in (26).

The method reductionLSDesignMatrix reduces duplicate columns in the matrix $A$ because of symmetry 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 $K$ resulting in a low variance of the $Q$, $R$ 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:

\[ n = 1, m = 1, p = 2, \]
\[ F = 0.9, N = 0.5, M = 2, Q = 1.5, \]
\[ H = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \]
\[ O = \begin{bmatrix} 2 & 0.1 \\ 0.1 & 1 \end{bmatrix}, \]
\[ R = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 3 \end{bmatrix}, \]
\[ \bar{x}_0 = 0, P_0 = 1, \]

is used. The matrices \( Q \), \( R \) 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:

\[
\begin{align*}
  f &= \text{nefLinFunction}(0.9, 0.5, 2); \\
  H &= [1; 0.5]; \\
  O &= [2, 0.1; 0.1, 1]; \\
  h &= \text{nefLinFunction}(H, [], O); \\
  x0 &= \text{nefGaussianRV}(0, 1); 
\end{align*}
\]

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

\[
\begin{align*}
  w &= \text{nefGaussianRV}(0, 1.5); \\
  v &= \text{nefGaussianRV}([0; 0], [2, 0.5; 0.5, 3]); 
\end{align*}
\]

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

\[
\text{myModel} = \text{nefEqSystem}(f, h, w, v, x0); 
\]

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

\[
\begin{align*}
  T &= 2e3; \\
  u &= [\sin(0:\pi/(T-1):\pi)]; \\
  z &= \text{myModelsimulate}(T, u); 
\end{align*}
\]

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 \( R \). Noise covariance matrices identification step starts with the creation of the nefALS class object:

\[
\text{myALS} = \text{nefALS}(f, h, x0, 'numberCovEq', 3); 
\]
Having the simulated data $z$, the matrices $Q$ and $R$ 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 $R$ with ‘fast’ setting of the nefALS

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

**Example 2** – estimate of all elements of $Q$ and $R$ 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 $R$ with predictor gain $K$ given by the user defined matrices $Q_A$ and $R_A$

```matlab
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 $R$ (just one element of $R$) 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.

<table>
<thead>
<tr>
<th>Noise covariance matrices</th>
<th>True</th>
<th>Example 1</th>
<th>Example 2</th>
<th>Example 3</th>
<th>Example 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q$</td>
<td>1.5</td>
<td>1.5037 ± 0.2446</td>
<td>1.5184 ± 0.1671</td>
<td>1.5213 ± 0.1865</td>
<td>1.5012 ± 0.2313</td>
</tr>
<tr>
<td>$R_{1,1}$</td>
<td>2</td>
<td>1.9982 ± 0.2202</td>
<td>1.9944 ± 0.1559</td>
<td>1.9918 ± 0.1669</td>
<td>–</td>
</tr>
<tr>
<td>$R_{1,2}$</td>
<td>0.5</td>
<td>0.4980 ± 0.2168</td>
<td>0.4937 ± 0.1447</td>
<td>0.4940 ± 0.1505</td>
<td>–</td>
</tr>
<tr>
<td>$R_{2,2}$</td>
<td>3</td>
<td>2.9964 ± 0.3254</td>
<td>2.9936 ± 0.1924</td>
<td>2.9926 ± 0.19687</td>
<td>2.9992 ± 0.3795</td>
</tr>
</tbody>
</table>

| 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](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

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

\[ N_{\text{maxIdentParam}} = np + p^2, \text{ if } n > p. \]  

(32)

However, with respect to the structure of \( b \) in (20) and symmetry of \( C_e, 0 \) (11), the ALS estimate is
based on

\[ N_{\text{independentEqs}} = \frac{p(p + 1)}{2} + p^2(L - 1) \]  

(33)

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

\[ L_{\text{min}} = \left\lceil \frac{p + n}{p} \rightceil. \]  

(34)
Acknowledgement
This work was supported by the Czech Science Foundation, project no. GA 15-12068S.

References