Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

Process noise covariance estimation via stochastic approximation

Federico Bianchi* | Simone Formentin | Luigi Piroddi

[1]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy, 20133 Milano (Italy), via Ponzio 34/5

Correspondence
*Federico Bianchi. Email: federico.bianchi@polimi.it

Summary

Kalman filtering for linear systems is known to provide the minimum variance estimation error, under the assumption that the model dynamics is known. While many system identification tools are available for computing the system matrices from experimental data, estimating the statistics of the output and process noises is still an open problem. In fact, although some methods based on maximum likelihood and correlation approaches have been proposed in the literature, it turns out that the existing techniques are either too computationally expensive or not accurate enough. In this paper, we propose an alternative solution – tailored for process noise covariance estimation and based on stochastic approximation and gradient-free optimization – that provides a better trade-off in terms of performance and computational load. The effectiveness of the method as compared to the state of the art is shown on a number of recently proposed benchmark examples.

KEYWORDS:
process noise covariance, state-space models, state estimation, stochastic approximation, gradient-free optimization

1 | INTRODUCTION

Kalman filtering is probably the most widespread tool for designing virtual sensors and state estimators in dynamical systems [1,2,3]. Within the linear framework, it can be shown that the Kalman filter provides the minimum variance state estimator, under the assumption that the employed model accurately describes the real system dynamics. The model is characterized as a stochastic dynamical system in the state-space form, that includes both a model of the process relating the inputs to the outputs and a noise model accounting for both noise effects and unmodeled dynamics. Such noise model is a crucial complement to the process model, which is typically the result of some approximations. This is even more true in practical applications where the physics underlying the systems is unknown or too complex to model from first principles.

System identification [4,5], namely the science of learning dynamical models from experimental data, plays a key role in such conditions and has been extensively applied to compute the system matrices from data, resulting in several well-established techniques. On the other hand, a relatively smaller effort has been devoted to characterize the noise model, which however is a key factor affecting the quality of the state estimator [6]. In the Kalman filtering context, the noise model is expressed in terms of the noise Covariance Matrices (CMs) acting on the state and output equations. In particular, while the variance of the output noise can be roughly estimated by looking at the output spectrum, the process noise statistics are generally unknown, given that the process noise model accounts for the unmodeled dynamics on which typically no prior knowledge is available. It follows that, in many applications, tuning the process noise covariance matrix requires extensive experimental campaigns and time-consuming trial-and-error procedures.

Some research works in the literature have investigated the problem of jointly estimating the state and the noise covariance matrices (CMs). The available approaches can be classified into two main categories. The *feedback methods* estimate simultaneously the unknown CMs and the state of the system by defining a vector of the extended state, which includes the parameters of the unknown CMs. In *feedback-free methods* instead, the states are first estimated by a sub-optimal filter and then the prediction error of the latter is used to calculate the CMs. The main drawback of feedback methods is the fact that the extended system is nonlinear. Feedback-free methods are then usually preferred, in view of their usually lower computational burden. Moreover, consistency and unbiasedness properties can often be shown to hold for such methods. Among feedback-free techniques, we can distinguish between *correlation-based methods*, like[7,8,9,10], and *maximum-likelihood methods*, such as[11] and[12]. Overall (see, *e.g.*, the survey[13] which carries out a full comparison), an accuracy/complexity trade-off applies to noise CM estimation methods, in that some methods have good accuracy properties obtained at the price of a large computational burden, while efficient methods have to settle for less accurate estimates.

The above observations led us to introduce a novel estimation method, based on stochastic approximation and gradient-free optimization, which aims at obtaining a more balanced trade-off between estimation performance and computational load. The main idea is to recast the estimation of the process noise CM as an optimization problem, whose objective function is defined in terms of the existing relation between the unknown CM and the Kalman filter state estimation error. In this way, the search for the minimum of the so-defined objective function actually corresponds to the search for the process noise CM that best approximates the optimal behavior of the Kalman filter. The resulting optimization is solved by means of a novel iterative method inspired by the stochastic approximation method presented in[14] and exploiting a parabolic interpolation[15] approach in the optimization.

The remainder of the paper is structured as follows. The process noise covariance estimation problem is formally stated in Section 2. Section 3 briefly reviews the existing methods to highlight the main features and the limitations of the state of the art. The proposed algorithm is introduced and analysed in Section 4, while the results of the experimental tests on some benchmark examples are illustrated and discussed in Section 5. The paper ends with some concluding remarks.

2 | PROBLEM STATEMENT

Consider the state-space model of a linear time-invariant discrete-time dynamic stochastic system S with additive noise (and no external input, for simplicity):

$$\begin{aligned} x(t+1) &= Fx(t) + v(t) \\ y(t) &= Hx(t) + w(t), \end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ denotes the state vector, $y(t) \in \mathbb{R}^p$ is the output vector, $v(t) \sim WGN(0, Q)^\dagger$ is the process noise with $Q = Q^T \succeq 0$, $Q \in \mathbb{R}^{n\times n}$, and $w(t) \sim WGN(0, R)$ indicates the measurement noise with $R = R^T \succ 0$, $R \in \mathbb{R}^{p\times p}$. F and H are the dynamic and output matrices, respectively. Without loss of generality, we assume here that v and w are uncorrelated.

In state estimation problems, the objective is to estimate the value of the state $\hat{x}(t|t)$ at time t, based on the available past output measurements up to the same instant t. In the Kalman setting, these estimates are obtained by a two-step procedure. In the *predictive phase*, the measurements up to $t-1$ are used to estimate the value of the states at the current instant t, as well as $P(t)$, which is the CM of the *state estimation error*:

$$\hat{x}(t|t-1) = F\hat{x}(t-1|t-1) \tag{2}$$

$$\hat{y}(t|t-1) = H\hat{x}(t|t-1) \tag{3}$$

$$P(t|t-1) = FP(t-1|t-1)F^T + Q. \tag{4}$$

Then, the measurement relative to the current time instant is used to adjust both the estimates of $x(t)$ and $P(t)$ (*update phase*):

$$\hat{x}(t|t) = \hat{x}(t|t-1) + K(t)\left[y(t) - H\hat{x}(t|t-1)\right] \tag{5}$$

$$P(t|t) = \left[I - K(t)H\right]P(t|t-1). \tag{6}$$

The Kalman gain $K(t)$ in (5) is obtained as:

$$K(t) = P(t|t-1)H^T\left[HP(t|t-1)H^T + R\right]^{-1}. \tag{7}$$

[†]WGN is short for White Gaussian Noise.

The *a priori* estimate of $x(t)$ calculated in the predictive phase is therefore adjusted through the *innovation* term $e(t) = y(t) - H\hat{x}(t|t-1)$, weighted by $K(t)$. The whole procedure is repeated at each time instant.

The formula (5) is known to minimize the variance of the state filtering error, provided that both the system matrices and the noise statistics are correctly estimated. If this is not the case, a robust implementation of the filter is needed, see, *e.g.*,[16,17], at the price of a more conservative performance. Sometimes, in practical applications, even if F and H can be properly computed from a set of experimental data with rigorous system identification tools, the noise CMs are tuned by trial and error on a much larger set of tests. In this paper, we will propose a rigorous system identification method that, based on one set of data (where the real state is also assumed to be available), finds the process noise CM providing the best filtering performance.[‡]

3 | RELATED WORKS

In this section, we will briefly review the main feedback-free methods for the computation of the process noise covariance matrix from data. Feedback methods are not covered here as they resort to nonlinear state estimation. For further details, see[13].

The first class of methods usually employed to estimate the noise CMs are the so-called *correlation-based methods*. Such techniques are based on the analysis of the *innovation* sequence, generated by a steady-state Kalman filter starting from an arbitrary initial condition $\hat{x}(0|-1)$ and a gain K selected such that the matrix $\bar{F} = F - FKH$ is stable. The pioneer of these methods is the work presented in[7] (denoted here as Indirect Correlation Method, ICM), in which the noise CMs are estimated using a three-step procedure with a classical least squares (LS) approach. The method is based on a system of N (user-defined) linear matrix equations stemming from the auto-covariance function (ACF):

$$C_j = \mathrm{cov}[e(t), e(t-j)] = \begin{cases} HPH^T + R, & j = 0 \\ H\bar{F}^{j-1}F(PH^T - KC_0), & j > 0 \end{cases} \tag{8}$$

where P is the CM of the steady-state *state estimation error*, given by the solution to the Lyapunov equation

$$P = \bar{F}P\bar{F}^T + FKRK^TF^T + Q. \tag{9}$$

Specifically, given the estimate of the ACF computed from the innovation sequence as

$$\hat{C}_j = \frac{1}{\tau - j}\sum_{t=j}^{\tau} e_t e_{t-j},\ j = 0, \dots, N-1, \tag{10}$$

where τ denotes the number of collected data, the following three steps are performed:

1. Compute the LS estimate $\widehat{PH}^T$ of PH^T from $\hat{C}_j$, $j = 1, \dots, N-1$, with $C_0 = \hat{C}_0$, according to (8).
2. Based on $\widehat{PH}^T$ and $C_0 = \hat{C}_0$, compute the estimate $\hat{R}$ from (8).
3. Compute the LS estimate $\hat{Q}$ of Q from (9), substituting $\widehat{PH}^T$ for PH^T and multiplying both sides of (9) by H and H^T, respectively.

Remark 1. Notice that the estimation of R is decoupled from that of Q, since the sampled version of C_j, namely $\hat{C}_j$ in (10), is employed to compute the system of N equations according to (8).

A second method, denoted here Weighted Correlation Method (WCM), is introduced in[8]. This approach is still based on the processing of the ACF, similarly to[7], but employs a suitable parameterization of the noise CMs and the ACF. More precisely, the unknown CMs are expressed as linear combinations of known (user-defined) basis matrices $Q^{(i)}$ and $R^{(i)}$, $i = 1, \dots, M$:

$$Q = \sum_{i=1}^{M}\theta(i)Q^{(i)},\ R = \sum_{i=1}^{M}\theta(i)R^{(i)},$$

[‡]As already mentioned, the output noise CM R can usually be inferred from sensor characteristics. When this is not possible, one may assume a diagonal structure for R and estimate it from the knowledge of y and x as $R = I_p \cdot \|\varepsilon\|^2$, with $\|\varepsilon\|^2 = \|y - Hx\|^2$. Another possibility, in case a dataset with x is not available for CM tuning, is to roughly estimate R from (8) as in[7].

where $\theta^{(i)}$, $i = 1, \dots, M$, are unknown weights to be estimated. In a similar fashion, the ACF is now defined as a linear function of the unknown weights $\theta(i)$:

$$C_j = \text{cov}[e(t), e(t-j)] = \sum_{i=1}^{M} \mathcal{F}_i(t, j)\theta(i),$$

where $\mathcal{F}_i(t, j)$ is appropriately defined (see Equation (16) in[8]).

Remark 2. Assuming for example $n = p = 2$, the basis matrices $Q^{(i)}$ and $R^{(i)}$, $i = 1, \dots, M$ can be selected as follows:

$$Q^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, Q^{(2)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Q^{(3)} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, Q^{(4)} = Q^{(5)} = Q^{(6)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{11}$$

$$R^{(1)} = R^{(2)} = R^{(3)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, R^{(4)} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, R^{(5)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, R^{(6)} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{12}$$

This justifies the use of a single vector of weights $\theta \in \mathbb{R}^M$ for both the $Q^{(i)}$s and the $R^{(i)}$s.

Finally, the method presented in[9] (denoted here as Direct Correlation Method, DCM), and extended in[10], estimates the CMs in a single step, by reformulating the relations of[7] in such a form that the three intermediate steps can be replaced by the resolution of a single LS problem.

The three mentioned methods can estimate all the elements of R and no more than $n \cdot p$ elements of Q, and they require the full knowledge of the structure of Q (*i.e.*, the number and position of the nonzero elements). In particular, the method in[7] does not work when Q has more than $n \cdot p$ nonzero elements. Furthermore, they are based on the assumption that the sampled ACF $\hat{C}_j$ approaches its true value. However, this is not generally true for finite data sets, especially due to the dependence of the innovation sequence on the *a priori* defined gain K. On the other hand, these methods appear to be very computational efficient, since they only need a single-point estimation following a classical LS approach.

The *maximum-likelihood (ML) methods* are based on the maximization of a likelihood function over the collected data. The method in[11] (denoted Input-Output Correlation Method, IOCM) is based on the minimization of the innovation related to an input-output ARMAX model of order $\mathcal{O}$. Assuming that $\mathcal{O} = 1$, $n = p$, and $H = I_n$, the ARMAX model can be reformulated as in (1). A ML step is first required to compute the covariance and the cross-covariances of the innovation, that are needed to estimate the noise CMs.

A classical ML approach is instead adopted in[12] (here denoted MLM), where a negative log-likelihood function is directly maximized w.r.t. the unknown CMs. Therefore, compared to the previous method, no intermediate steps are needed to recover the optimal CMs. However, the *a priori* setting of the unknown CMs is crucial for the convergence of the method and the accuracy of the final estimate.

In general, although they generally provide more accurate solutions than correlation-based methods, ML methods involve a much more significant computational load, mainly due to the computation of the gradient descent direction from data over the parameter space.

4 | PROCESS NOISE COVARIANCE ESTIMATION VIA STOCHASTIC APPROXIMATION

we will propose an alternative solution to the process noise CM estimation, based on the *stochastic approximation* approach introduced in[18] and[14]. Specifically, we will propose a *gradient-free* policy optimizing a stochastic approximation of the real cost function, namely the mismatch between actual and estimated state, during a training phase. This is doable under the assumption that a preliminary dataset including the state measurement is available. However, this is not a strong assumption in many practical applications, see, e.g., the Kalman-filter based roll angle estimation of[19] and[20].

As ML-based methods, the proposed approach does not suffer from the limitation of the correlation-based methods regarding the maximum number of free parameters of Q that can be estimated. At the same time, the use of a gradient-free optimization approach compares quite favorably with the computationally heavy ML-based methods.

The problem reformulation involves the definition of an *ad hoc* objective function, which relates the estimation accuracy to the unknown CMs, assuming that the optimal estimates of the latter are those that guarantee maximum filtering accuracy. For the reasons explained previously, we limit the optimization to matrix Q, assuming that R is fixed. Precisely, let the objective

function be:

$$J(Q) = \frac{1}{\tau}\sum_{t=1}^{\tau}\left(x(t) - \hat{x}_Q(t)\right)^T\left(x(t) - \hat{x}_Q(t)\right) \in \mathbb{R}^+, \tag{13}$$

where $\hat{x}_Q(t)$ is the state estimate obtained using matrix Q as the process noise CM and τ is the number of collected observations. The optimization problem can then be formulated as that of finding the semidefinite positive matrix Q that optimizes the filter accuracy as described by $J(Q)$:

$$\begin{aligned} &\underset{Q}{\text{minimize}} \quad J(Q) \\ &\text{s.t. } Q \succeq 0 \end{aligned} \tag{14}$$

Figure 1 depicts the relationship among all the variables involved in the optimization problem.

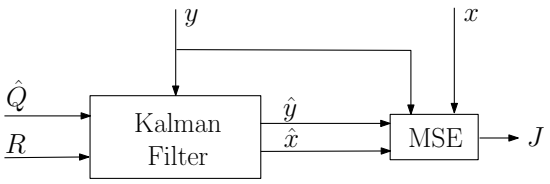

FIGURE 1 Graphical representation of the relationship between J, Q, and R.

The optimization problem (14) is subject to the constraint that $Q \succeq 0$, but can easily be reformulated as an unconstrained one by factorizing Q using a Cholesky decomposition[21] to automatically ensure its positive definiteness:

$$Q = ZZ^T, \tag{15}$$

where Z is a lower triangular $n \times n$ matrix. Accordingly, the optimization problem becomes:

$$\underset{Z}{\text{minimize}} \quad J(Z) \tag{16}$$

where with a slight abuse of notation we denote as $J(Z)$ the cost function $J(Q) = J(ZZ^T)$ defined in terms of Z.

As it is formulated, the envisaged optimization problem cannot be solved in closed form but relatively simple iterative solution algorithms can be applied to find the optimum Q in the absence of constraints. For example, Newton-type or other gradient-based algorithms could be suitable for this purpose. Here we employ a stochastic approximation method (see[14] and[22] for a review of such methods), which avoids the computation of gradients of the cost function and requires an affordable computational load.

4.1 | Stochastic approximation methods

The working principle behind stochastic approximation methods can be summarized as follows. Consider a target function $f(\theta)$, where θ is the parameter vector. The goal is to find θ° for which $f'(\theta^\circ) = 0$. Assume that only (noisy) measurements of $f(\theta)$ – but not of its derivative $f'(\theta)$ – are available for some θ's. Then the optimum point can be estimated iteratively by solving the following update equation at each iteration:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - \alpha_k \hat{g}_k(\hat{\theta}_k) \tag{17}$$

where $\hat{g}_k(\hat{\theta}_k)$ is an estimate of the gradient $g(\theta) = f'(\theta)$ at the kth iteration. For the calculation of $\hat{g}_k(\hat{\theta}_k)$ one needs only measurements of the target function at given points, specifically $f(\hat{\theta}_k + h_k)$ and $f(\hat{\theta}_k - h_k)$, where h_k is a sufficiently small perturbation. The Finite Difference Stochastic Approximation (FDSA)[18], and the Simultaneous Perturbation Stochastic Approximation (SPSA)[14] provide two alternative approximations of the gradient, by perturbing the components of the vector $\hat{\theta}_k$ in different ways: FDSA perturbs one parameter at a time, while SPSA perturbs all of them simultaneously. Specifically, with the FDSA method the ith component of the gradient is computed as:

$$\hat{g}_{k,i}(\hat{\theta}_k) = \frac{f(\hat{\theta}_k + h_k e_i) - f(\hat{\theta}_k - h_k e_i)}{2h_k}, \tag{18}$$

where e_i is vector whose elements are all equal to 0 except the ith one which is equal to 1. The main drawback of the FDSA is the large computational load at each iteration, in view of the numerous evaluations of function $f(\cdot)$ involved, which ultimately

causes a slow convergence of the algorithm. Conversely, with the SPSA method all the components of the parameter vector are simultaneously perturbed by a random amount, so that the ith component of the gradient can be computed as

$$\hat{g}_{k,i}(\hat{\theta}_k) = \frac{f(\hat{\theta}_k + h_k\Delta_k) - f(\hat{\theta}_k - h_k\Delta_k)}{2h_k\Delta_{k,i}} \tag{19}$$

where $\Delta_k = [\Delta_{k,1}\ \Delta_{k,2}\ \dots\ \Delta_{k,n}]^T$ is the perturbation vector, which contains only ± 1 in random positions. In this case, the same two function evaluations $f(\hat{\theta}_k \pm h_k\Delta_k)$ are used for each component, thus significantly reducing the computational burden.

4.2 | The SAPI method: the scalar case

The proposed method for solving problem (14) or (16), denoted SAPI (Stochastic Approximation with Parabolic Interpolation), belongs to the category of stochastic approximation methods briefly described in the previous subsection. To convey more effectively the idea behind the SAPI, we first describe it in the simple case where Q is a scalar. The more general case will be addressed in subsection 4.3.

In the scalar case, there is only one free parameter and there is no need to resort to the reformulation (16), so we will make reference to the original formulation (14). The error surface is a 2-dimensional curve and for a given starting value of Q there are only two possible directions of exploration, as depicted in Figure 2.

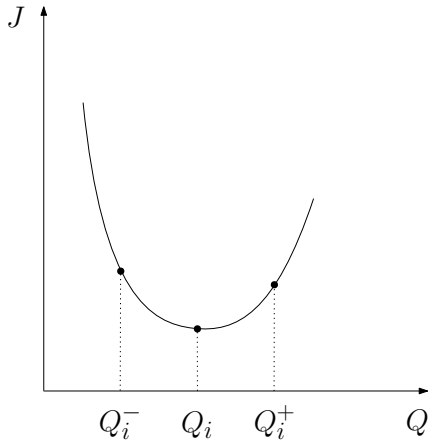

FIGURE 2 Graphical representation of the J curve exploration in the scalar case.

Starting from an initial point Q_0 and an initial step size α_0, the following steps are carried out at each iteration i:

1. Let $Q_i^+ = Q_i + \alpha_i$ and $Q_i^- = Q_i - \alpha_i$.
2. Compute $J_i = J(Q_i)$, $J_i^+ = J(Q_i^+)$, and $J_i^- = J(Q_i^-)$.
3. If $J_i \geq \frac{J_i^- + J_i^+}{2}$ then choose Q_{i+1} as:
$$Q_{i+1} = \underset{\{Q_i, Q_i^+, Q_i^-\}}{\arg\min}\ J(Q), \tag{20}$$
set α_{i+1} as:
$$\alpha_{i+1} = \alpha_i + \mu\alpha_i, \tag{21}$$
and go to Step 8.
4. Interpolate a parabola over the three points (Q_i, J_i), (Q_i^+, J_i^+), and (Q_i^-, J_i^-).
5. Find the minimum point Q_i° of the interpolating parabola, and compute the corresponding $J_i^\circ = J(Q_i^\circ)$.
6. Choose Q_{i+1} as:
$$Q_{i+1} = \underset{\{Q_i, Q_i^+, Q_i^-, Q_i^\circ\}}{\arg\min}\ J(Q) \tag{22}$$

7. Compute α_{i+1} as follows:

$$\alpha_{i+1} = \begin{cases} \alpha_i + \mu\alpha_i, & \text{if } Q_{i+1} \neq Q_i^\circ \\ \alpha_i - \beta\alpha_i, & \text{if } |J_i - J_{i-1}| \leq \texttt{TOL} \\ \alpha_i, & \text{otherwise} \end{cases} \tag{23}$$

8. If $|J_i - J_{i-1}| \leq \texttt{TOL}$ then $C := C + 1$, else $C = 0$. If $C = \texttt{ITER\_OBS}$ then exit, else $i := i + 1$ and go to Step 1.

After Step 2, one could evaluate which of the three points gives the best J and simply elect it as the new starting point for the next iteration (*basic selection* (20)). However, to further speed up the minimization of J, a local *parabolic interpolation*[15] is applied on the cost function when possible (*i.e.*, when the interpolating parabola is opening to the top). The parabolic interpolation method is a well known technique to find the minimum of a unimodal function, its rationale being illustrated in Figure 3. Assume that the function (solid line) has been evaluated at three points p_1, p_2 and p_3 and that the parabola (dashed line) passing through them has been computed. Its minimum, *i.e.* point p_4, is closer to the minimum of the function (point p_5) than point p_2 (which is the lower of the three computed points and would have been chosen according to the basic selection (20)).

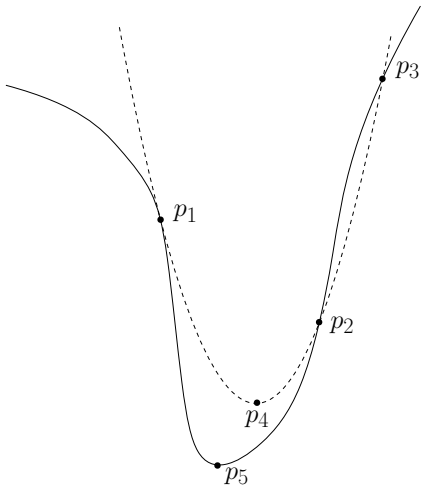

FIGURE 3 The rationale behind parabolic interpolation.

The application of (20) or, alternatively, (22) ensures that the sequence of J_i values is decreasing (see the next subsection for a more detailed discussion about convergence). Notice that J_i° is the value of J at the minimum point of the parabola Q_i°, and it is not guaranteed that it will be lower than J_i, J_i^+, and J_i^-. This is why the new point must be calculated by means of (22).

The role of the step size α is critical in the functioning of the algorithm. In the early stages of the algorithm, when Q_i might be far from the optimum, a large α value allows the algorithm to exploit the parabolic interpolation mechanism to quickly run away from that region. As the algorithm proceeds, α should be gradually reduced to increase the algorithm resolution in the vicinity of the (local) optimum point. Notice that for small values of α the three points (Q_i, J_i), (Q_i^+, J_i^+), and (Q_i^-, J_i^-) may turn out to be close to collinearity, causing the minimum point (Q_i°, J_i°) of the parabola to be far apart. While this may occasionally allow the algorithm to exit from a local minimum, it generally reduces the utility of the parabolic interpolation mechanism. In such cases, increasing α and thus picking points at larger distances may produce more meaningful parabolic interpolations, avoiding a repetition of the problem in the subsequent iterations. These observations are condensed in the update rule (23), where parameters $\mu > 0$ and $\beta > 0$ govern respectively the increase and decrease of α.

The stopping criterion is met when J does not improve by more than $\texttt{TOL}$ for $\texttt{ITER\_OBS}$ subsequent iterations, $\texttt{TOL} > 0$ and $\texttt{ITER\_OBS} \in \mathbb{N}^+$ being user-defined thresholds. These two parameters have to be chosen carefully in order to avoid the early stopping of the algorithm, thus allowing it to converge.

4.3 | The SAPI method: the general case

When the state x is an n-dimensional vector, with $n > 1$, the number of possible directions of exploration increases in view of the larger size of Q. Indeed, the free parameters are $n_\theta = \frac{n(n+1)}{2}$ due to the symmetry of matrix Q. As mentioned previously, the process noise CM matrix is factorized as $Q = ZZ^T$, Z being a lower triangular $n \times n$ matrix, and the optimization is carried out with reference to problem (16). The optimization parameters are the elements z_{jk} of matrix Z with $j \geq k$. Accordingly, the parameter vector is defined as $\theta = [z_{11}\ z_{21}\ z_{22}\ \dots\ z_{nn}]^T$. To simplify the notation, we will indicate as $J(\theta)$ the cost function $J(Z) = J(Z(\theta))$. The exploration of J according to the SAPI mechanism is carried out by simultaneously perturbing all the components of θ.

Specifically, the parameter perturbation is applied along a random direction defined by the perturbation vector Δ_i, both in the positive and negative directions, its amplitude being defined (on each axis) by the step size parameter α_i. According to[23], a simple (and theoretically valid) choice for each component of the perturbation vector Δ_i in (19) is to use a Bernoulli ± 1 distribution with probability of 0.5, denoted here as $\mathtt{Be}(0.5)$, for each ± 1 outcome. Starting from θ_i, a perturbation Δ_i vector is extracted according to the mentioned Bernoulli distribution, and the following two points are computed in the parameter space:

$$\theta_i^{\pm} = \theta_i \pm \alpha_i \Delta_i = \begin{bmatrix} \theta_{i,1} \pm \alpha_i \Delta_{i,1} \\ \theta_{i,2} \pm \alpha_i \Delta_{i,2} \\ \vdots \\ \theta_{i,N} \pm \alpha_i \Delta_{i,N} \end{bmatrix}. \tag{24}$$

As in the scalar case, the value of J can be calculated in correspondence to the three parameterizations $Z(\theta_i^-)$, $Z(\theta_i)$, and $Z(\theta_i^+)$, obtaining respectively the values J_i^-, J_i, and J_i^+. Notice that although J is a $n_\theta + 1$-dimensional surface, the three points θ_i^-, θ_i, and θ_i^+ all stand on a straight line, defined as:

$$\theta_i^*(a) = (1 - a)\theta_i^- + a\theta_i^+. \tag{25}$$

Indeed, $\theta_i^- = \theta_i^*(0)$, $\theta_i = \theta_i^*(0.5)$, and $\theta_i^+ = \theta_i^*(1)$. Thanks to this property, the parabolic interpolation can be applied as in the scalar case in the plane defined by a and J. The procedure requires to find the minimum a° of the parabola interpolating the three points $(0, J_i^-)$, $(0.5, J_i)$, and $(1, J_i^+)$ in the (a, J) space and then to compute the corresponding θ_i° as follows:

$$\theta_i^\circ = (1 - a^\circ)\theta_i^- + a^\circ \theta_i^+. \tag{26}$$

Algorithm 1 (see the next page) depicts the overall procedure for the general case and its convergence to a local minimum of the considered cost can be proven as follows.

Theorem 1. Algorithm 1 converges to a local minimum of J.

Proof. The convergence of Algorithm 1 is proven upon observing that the application of (20) or (22) (when applicable) ensures that the sequence $\{J_i\}$ is non-increasing, *i.e.,* $J_{i-1} \geq J_i$. Furthermore, J is by definition bounded from below ($J \geq 0$), and hence, for the Monotone Convergence Theorem[24], it converges to a limit value. Such a limit is associated to a local minimum point of the surface $J(Q)$. In fact, at any new iteration, either the optimum point Q_i changes in favor of a better one or it remains unchanged. If it stops changing, or the difference between J_i and J_{i-1} becomes negligible, the value of α between consequent iterations also decreases and the distance between the points $Q_i^{\pm}$ shrinks. □

5 | SIMULATION RESULTS

In this section two simulation examples are discussed to show the effectiveness of the proposed method w.r.t. the approaches briefly reviewed in Section 3. First, a comparative analysis is carried out (Section 5.1) on the example reported in[13], which represents an ideal case in which the applicability and the identifiability conditions of all the considered methods are met. Then, a second example is discussed (Section 5.2), in which some identifiability issues arise.

All tests are performed within a MATLAB 2017a environment[25], on a i7-4702MQ CPU @2.20 GHz with 8GB of RAM. The code implementation of the methods appearing in the comparative analysis is part of the package made publicly available by[13].

Algorithm 1 SAPI

Require: $\{(x(t), y(t)), t = 1, \cdots, \tau\}$, F, H, R, θ_0, α_0, μ, β, TOL, ITER_OBS
Ensure: Q

$C \leftarrow 0$; ▷ Convergence counter
$n_\theta \leftarrow |\theta_0|$;
$Q_i \leftarrow$ compute_Q(θ_0); ▷ Retrieve Q from θ
$J_i \leftarrow$ compute_J(Q_i, F, H, R, x, y); ▷ Compute J according to (13)
$\gamma \leftarrow 0$; ▷ Needed boolean variable
repeat
  for $l = 1$ **to** n_θ **do** ▷ Generate a random ± 1 perturbation vector
    $\Delta_{i,l} \sim$ Be(0.5);
  end for
  $\theta_i^{\pm} \leftarrow \theta_i \pm \alpha_i \Delta_i$;
  $Q_i^{\pm} \leftarrow$ compute_Q$(\theta_i^{\pm})$;
  $J_i^{\pm} \leftarrow$ compute_J$(Q_i^{\pm}, F, H, R, x, y)$;
  if $J_i \geq \frac{J_i^- + J_i^+}{2}$ **then**
    $Q_{i+1} \leftarrow \underset{\{Q_i, Q_i^+, Q_i^-\}}{\arg\min}\ J(Q)$;
    $\alpha_{i+1} \leftarrow \alpha_i + \mu\alpha_i$;
    $\gamma \leftarrow 1$;
  else
    $K \leftarrow$ parabolic_interpolation $\left((0, J_i^-), (0.5, J_i), (1, J_i^+)\right)$; ▷ Parabola in the form $b = K_1 a^2 + K_2 a + K_3$
    $a^\circ \leftarrow -K_2/2K_1$; ▷ Compute the minimum
    $\theta_i^\circ \leftarrow [\theta_i^+,\ \theta_i^-] \cdot [a^\circ,\ (1 - a^\circ)]^T$;
    $Q_i^\circ \leftarrow$ compute_Q(θ_i°);
    $J_i^\circ \leftarrow$ compute_J$(Q_i^\circ, F, H, R, x, y)$;
    $Q_{i+1} \leftarrow \underset{\{Q_i, Q_i^+, Q_i^-, Q_i^\circ\}}{\arg\min}\ J(Q)$;
    if $Q_{i+1} \neq Q_i^\circ$ **then**
      $\alpha_{i+1} \leftarrow \alpha_i + \mu\alpha_i$;
      $\gamma \leftarrow 1$;
    end if
  end if
  if $|J_i - J_{i-1}| \leq$ TOL **then** ▷ Check for algorithm convergence
    $C \leftarrow C + 1$
    if $\gamma = 0$ **then**
      $\alpha_{i+1} \leftarrow \alpha_i - \beta\alpha_i$;
    end if
  else
    $C \leftarrow 0$
  end if
until $C \leq$ ITER_OBS ▷ Stopping criterion

5.1 | Example 1

Consider the system in the form (1), with $F = \begin{bmatrix} 0.9 & 0 \\ -0.3 & 0.8 \end{bmatrix}$, $H = I_2$, $Q = \begin{bmatrix} 2 & -0.5 \\ -0.5 & 1 \end{bmatrix}$ and $R = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$[13]. The following settings[13] have been adopted for the different algorithms considered in the comparison:

- ICM, DCM:
  - $N = 2$

  - filter gain $K_0 = 0.8 I_2$

- WCM:
  - $N = 2$
  - filter gain $K_0 = 0.8 I_2$
  - basis matrices: $Q^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, $Q^{(2)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $Q^{(3)} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$, $Q^{(4)} = Q^{(5)} = Q^{(6)} = 0_2$

- IOCM:
  - initial condition for the estimation of B, $B = 0_2$

- MLM:
  - number of EM steps: 50
  - initial value of Q, $Q_0^0 = \begin{bmatrix} 3.43 & 0.75 \\ 0.75 & 2.73 \end{bmatrix}$
  - initial value of R, $R_0 = R$, *i.e.,* the initial estimate equals the true value
  - initial state estimation error CM set to $P_{0|-1} = 0_2$

- SAPI:
  - $\texttt{ITER\_OBS} = 10$
  - $\texttt{TOL} = 10^{-7}$
  - initial step size: $\alpha_0 = 0.5$
  - increasing step factor: $\mu = 0.2$
  - decreasing step factor: $\beta = 0.05$
  - initial θ vector: $\theta_0 = [1, \dots, 1]^T$

Tables 1 and 2 report the aggregated results obtained from 10^3 different realizations of the noises $v(t) \sim WGN(0, Q)$ and $w(t) \sim WGN(0, R)$, considering respectively two datasets of different size, namely $\tau = 10^4$ and $\tau = 10^5$. For each of the considered methods, these tables report the average and variance of the estimates of the individual q_{ij}, $i, j = 1, 2$, values, the elapsed time (ET) expressed in seconds, and the relative error:

$$\mathcal{L} = \frac{|J^\circ - \hat{J}|}{J^\circ},$$

where J° and $\hat{J}$ are respectively the J values computed according to (13) when $\hat{x}$ has been obtained starting from the true Q and $\hat{Q}$.

Apparently, all the methods provide comparable results in terms of the accuracy of the q_{ij} estimates. As for the computational burden, the correlation-based methods (ICM, DCM, and WCM) outperform the SAPI, but this is to be expected since they rely on a single-point estimate, whereas the SAPI is iterative by nature. Instead, there appears to be an evident advantage in using the SAPI compared to the ML-based methods from a computational point of view. This gap increases significantly with τ. Indeed, the ET values corresponding to the IOCM and MLM methods are one order of magnitude bigger than those of the SAPI for $\tau = 10^5$.

An important issue with correlation-based methods is that they depend on the arbitrary initial choice of the filter gain K_0. It is therefore interesting to assess the robustness of such methods for different values of K_0. Figures 4 and 5 report the results of a sensitivity analysis to K_0 that has been carried out for the DCM method (but the results can be generalized to the ICM and WCM algorithms, as well), by considering $K_0' = 0.8 I_2$ and $K_0'' = 1.5 I_2$. The figures show a strong sensitivity of the method to this design parameter in terms of the variance of the estimates and also of the relative error. Indeed, for K_0'' the mean relative error $\mathcal{L}$ is equal to $7.26 \cdot 10^{-3}$, which is significantly bigger than for K_0' (see Table 1), and also compared to the SAPI.

method		$\hat{q}_{11}$	$\hat{q}_{12}$	$\hat{q}_{22}$	ET [s]	$\mathcal{L}$
ICM	avg.	2.0039	-0.4999	0.9979	0.0368	$6.8712 \cdot 10^{-4}$
	var.	0.0087	0.0030	0.0041		
DCM	avg.	1.9992	-0.5045	1.0004	0.0365	$6.9102 \cdot 10^{-4}$
	var.	0.0086	0.0031	0.0040		
WCM	avg.	2.0003	-0.5028	0.9987	0.0382	$7.0559 \cdot 10^{-4}$
	var.	0.0095	0.0027	0.0041		
IOCM	avg.	2.0014	-0.4980	0.9980	13.0027	$3.0771 \cdot 10^{-4}$
	var.	0.0053	0.0017	0.0022		
MLM	avg.	1.9979	-0.4920	1.0096	29.8488	$2.8290 \cdot 10^{-4}$
	var.	0.0046	0.0016	0.0020		
SAPI	avg.	2.0019	-0.5003	1.0030	2.9206	$2.2410 \cdot 10^{-4}$
	var.	0.0037	0.0013	0.0012		

TABLE 1 Estimates of the elements of Q with $\tau = 10^4$ for 10^3 MC runs.

method		$\hat{q}_{11}$	$\hat{q}_{12}$	$\hat{q}_{22}$	ET [s]	$\mathcal{L}$
ICM	avg.	1.9987	-0.5000	0.9999	0.3630	$7.2808 \cdot 10^{-5}$
	var.	$0.9330 \cdot 10^{-3}$	$0.2942 \cdot 10^{-3}$	$0.4153 \cdot 10^{-3}$		
DCM	avg.	2.0009	-0.4995	1.0001	0.3595	$6.7294 \cdot 10^{-5}$
	var.	$0.8490 \cdot 10^{-3}$	$0.3025 \cdot 10^{-3}$	$0.4025 \cdot 10^{-3}$		
WCM	avg.	2.0008	-0.5006	1.0003	0.3598	$7.0442 \cdot 10^{-5}$
	var.	$0.9215 \cdot 10^{-3}$	$0.2869 \cdot 10^{-3}$	$0.4140 \cdot 10^{-3}$		
IOCM	avg.	2.0002	-0.5001	0.9998	115.7523	$2.9439 \cdot 10^{-5}$
	var.	$0.5056 \cdot 10^{-3}$	$0.17021 \cdot 10^{-3}$	$0.2300 \cdot 10^{-3}$		
MLM	avg.	1.9988	-0.4993	1.0014	249.1815	$2.4228 \cdot 10^{-5}$
	var.	$0.3724 \cdot 10^{-3}$	$0.1348 \cdot 10^{-3}$	$0.1293 \cdot 10^{-3}$		
SAPI	avg.	2.0013	-0.4999	1.0012	13.029	$4.1392 \cdot 10^{-5}$
	var.	$0.5677 \cdot 10^{-3}$	$0.3004 \cdot 10^{-3}$	$0.1918 \cdot 10^{-3}$		

TABLE 2 Estimates of the elements of Q with $\tau = 10^5$ for 10^3 MC runs.

5.2 | Example 2

Consider a system in the form (1), where

$$F = \begin{bmatrix} 0.0218 & 0.9243 & -0.2750 \\ 0.4645 & -0.2466 & -0.8076 \\ 0.8451 & 0.1167 & 0.4530 \end{bmatrix}, H = [0.9936\ 0\ 0.6539], Q = \begin{bmatrix} 6.3557 & 6.2921 & -0.7910 \\ 6.2921 & 6.5128 & -0.0420 \\ -0.7910 & -0.0420 & 6.7963 \end{bmatrix}, \text{and } R = \frac{\text{var}(y)}{10}.$$

Prior to analyzing the results stemming from a Monte Carlo simulation performed similarly to the previous example, let us point out the following aspects:

1. As discussed in Section 3, the correlation-based methods can typically estimate no more than $n \cdot p$ elements of Q. The proposed example emphasizes this identifiability issue, since $n = 3$ and $p = 1$, implying that only 3 elements of Q can be estimated, while Q has actually 6 elements. Furthermore, the ICM method does not work at all.

2. The IOCM cannot deal with this example, since it is based on the state-space reformulation of an ARMAX model by assuming that $n = p$ and $H = I_n$.

3. The SAPI, as well as the MLM, can address this case, but the identified solution is not unique. Accordingly, we will compare the methods only in terms of computational efficiency and accuracy.

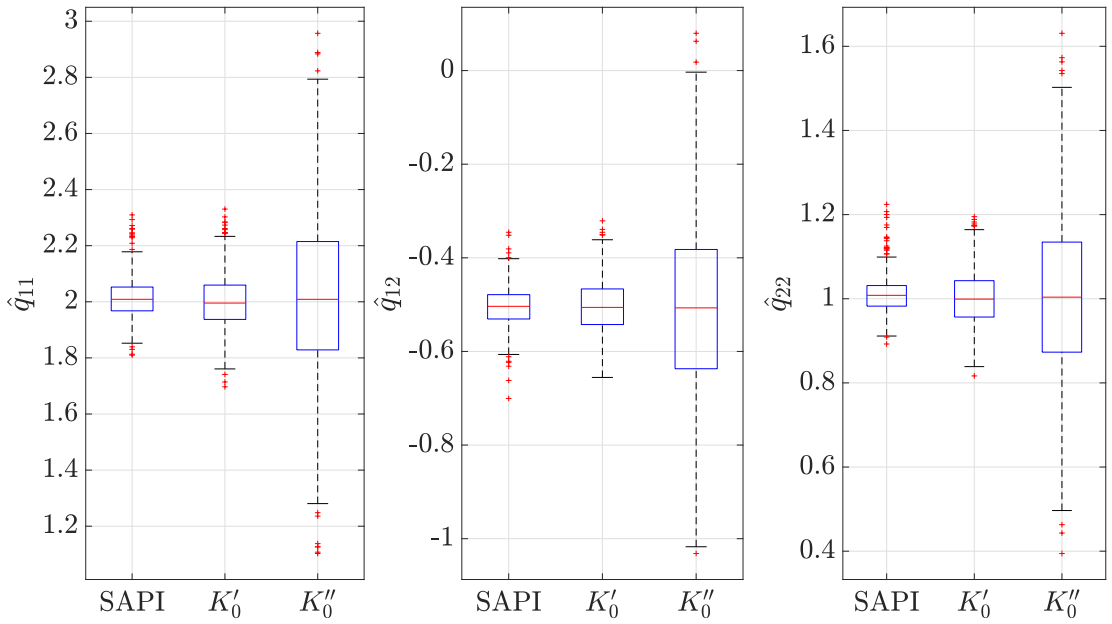

FIGURE 4 DCM method: Sensitivity to the design parameter K - estimation accuracy.

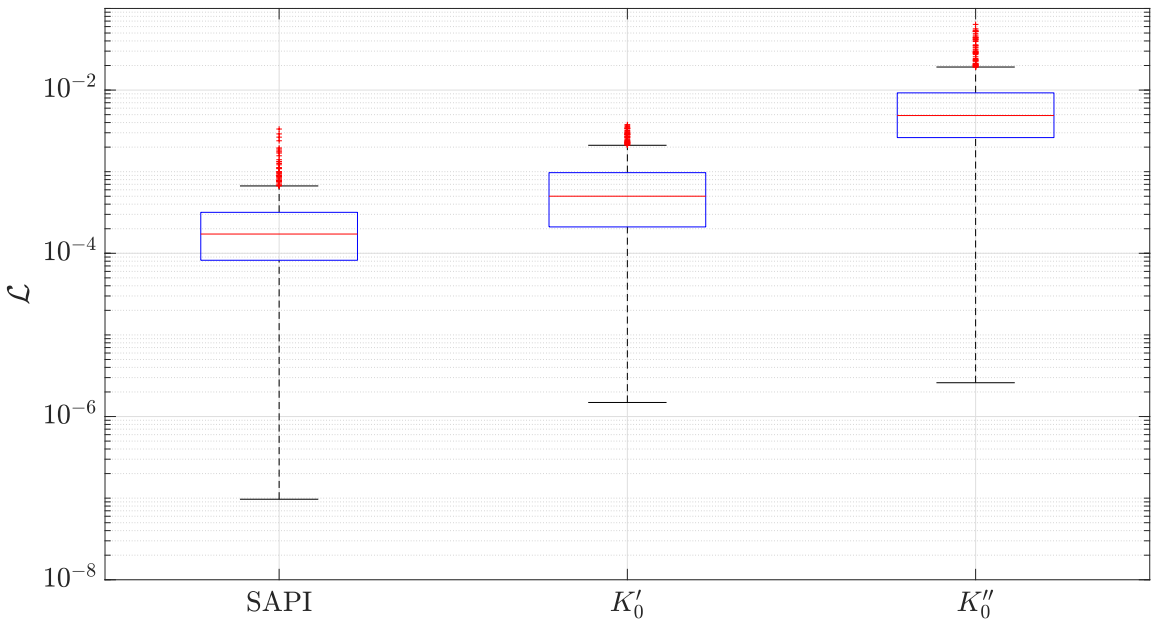

FIGURE 5 DCM method: Sensitivity to the design parameter K - relative error.

The studied methods are applied with the following settings [§]:

[§]The filter gain K_0 used for the DCM and WCM methods, is a fair approximation of the optimal predictor gain coming from the SAPI, which ultimately represents at least an optimal initialization.

- DCM:
  - $N = 2$
  - filter gain $K_0 = [0.1 \ \ -0.03 \ 0.5]^T$
- WCM:
  - $N = 2$
  - filter gain $K_0 = [0.1 \ \ -0.03 \ 0.5]^T$
  - basis matrices: $Q^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $Q^{(2)} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $\cdots$, $Q^{(10)} = 0_3$
- MLM:
  - number of EM steps: 30
  - initial value of Q, $Q_0 = 1.2Q$, *i.e.* the initial estimate is a scaled version of the true value
  - initial value of R, $R_0 = R$, *i.e.* the initial estimate equals the true value
  - initial state estimation error CM set to $P_{0|-1} = 0_3$

The SAPI has been set up as in the previous example.

The results are summarized in the plot of Figure 6. The correlation methods (DCM and WCM) are those requiring the least computational time but provide scarce accuracy. Conversely, the MLM is the most accurate method but requires a significantly large computational time. The SAPI approach reveals to be the solution providing the best trade-off between accuracy and computational burden.

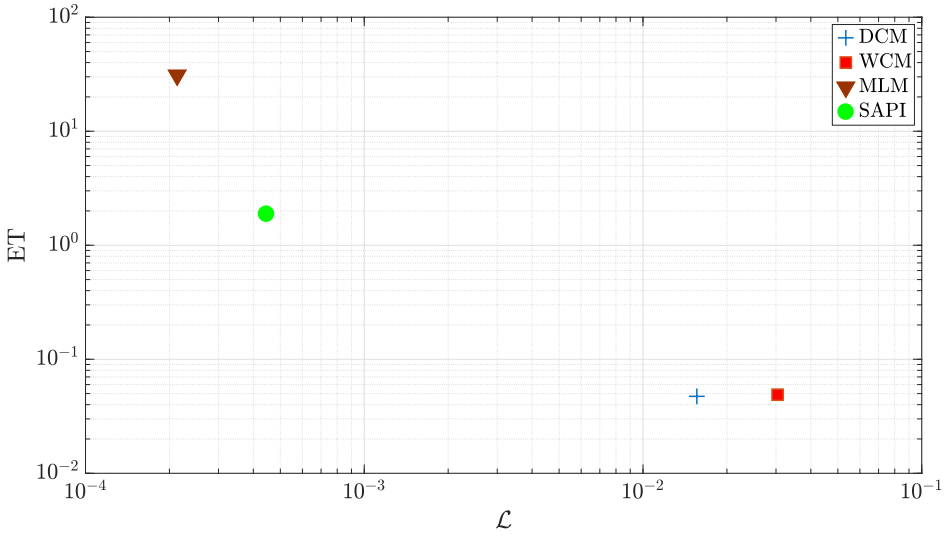

FIGURE 6 ET vs. $\mathcal{L}$ curve with $\tau = 10^4$ for 10^3 MC runs.

6 | CONCLUSIONS

In this paper, we have addressed the estimation of the process noise covariance matrix in a Kalman filter setting, proposing a novel method based on an efficient stochastic approximation method employing a local parabolic interpolation of the cost

function. The proposed method appears to be effective on the considered benchmark examples as compared to the state of the art.

Specifically, with respect to correlation-based methods, SAPI provides more accurate results, albeit at a slightly higher computational cost, and has an extended range of applicability, since it does not suffer from structural identifiability issues. Furthermore, SAPI significantly outperforms maximum-likelihood based methods in terms of computational load at the cost of a negligible accuracy degradation, thus providing a convenient trade-off between accuracy and computational burden. Such a trade-off can be suitably tuned using the parameters `TOL` and `ITER_OBS`. We stress here that SAPI can easily embed also constraints on the matrix structure, by simply constraining the exploration of the cost function surface. This observation is the basis of our ongoing work.

References

1. Kalman RE. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering* 1960; 82(1): 35–45.
2. Kalman RE, Bucy RS. New results in linear filtering and prediction theory. *Journal of Basic Engineering* 1961; 83(1): 95–108.
3. Pozzi G, Formentin S, Lluka P, Bittanti S. A Kalman filtering approach to traffic flow estimation in computer networks. *IFAC-PapersOnLine* 2018; 51(15): 37–42.
4. Verhaegen M, Verdult V. *Filtering and system identification: a least squares approach*. Cambridge University Press . 2007.
5. Ljung L. *System identification: theory for the user*. Prentice Hall . 1999.
6. Formentin S, Bittanti S. An insight into noise covariance estimation for Kalman filter design. *IFAC Proceedings Volumes* 2014; 47(3): 2358–2363.
7. Mehra R. On the identification of variances and adaptive Kalman filtering. *IEEE Transactions on Automatic Control* 1970; 15(2): 175–184.
8. Bélanger PR. Estimation of noise covariance matrices for a linear time-varying stochastic process. *Automatica* 1974; 10(3): 267–275.
9. Odelson BJ, Rajamani MR, Rawlings JB. A new autocovariance least-squares method for estimating noise covariances. *Automatica* 2006; 42: 303-308.
10. Rajamani MR, Rawlings JB. Estimation of the disturbance structure from data using semidefinite programming and optimal weighting. *Automatica* 2009; 45(1): 142–148.
11. Kashyap R. Maximum likelihood identification of stochastic linear systems. *IEEE Transactions on Automatic Control* 1970; 15(1): 25–34.
12. Shumway RH, Stoffer DS. Time series regression and exploratory data analysis. In: Time series analysis and its applications, Springer. 2011 (pp. 47–82).
13. Duník J, Straka O, Kost O, Havlík J. Noise covariance matrices in state-space models: A survey and comparison of estimation methods - Part I. *International Journal of Adaptive Control and Signal Processing* 2017; 31(11): 1505–1543.
14. Spall JC. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest* 1998; 19(4): 482–492.
15. Heath MT. *Scientific computing: An Introductory Survey*. McGraw-Hill New York . 2002.
16. Zorzi M. Robust Kalman filtering under model perturbations. *IEEE Transactions on Automatic Control* 2017; 62(6): 2902–2907.

17. Xie L, Soh YC, De Souza CE. Robust Kalman filtering for uncertain discrete-time systems. *IEEE Transactions on Automatic Control* 1994; 39(6): 1310–1314.

18. Kiefer J, Wolfowitz J. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics* 1952; 23: 462–466.

19. Boniolo I, Savaresi S, Tanelli M. Roll angle estimation in two-wheeled vehicles. *IET Control Theory & Applications* 2009; 3(1): 20–32.

20. Formentin S, Boniolo I, Lisanti P, Spelta C, Savaresi SM. Fault detection in roll angle estimation for two-wheeled vehicles. *IFAC Proceedings Volumes* 2012; 45(24): 54–59.

21. Björck Å. *Numerical methods in matrix computations*. 59. Springer . 2015.

22. Chin DC. Comparative study of stochastic algorithms for system optimization based on gradient approximations. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 1997; 27(2): 244–249.

23. Spall JC. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on Aerospace and Electronic Systems* 1998; 34(3).

24. Yeh J. *Real Analysis: Theory of Measure and Integration Second Edition*. World Scientific Publishing Company . 2006.

25. MATLAB . *Version 2017b*. Natick (MA), USA: The MathWorks Inc. . 2017.

□