EURASIP Journal on Advances
in Signal Processing

RESEARCH Open Access

New graphical models for sequential data and the improved state estimations by data-conditioned driving noises

Wonjung Lee[1,2*]

*Correspondence:
wonjunglee@bnu.edu.cn

[1] Research Center for Mathematics, Beijing Normal University at Zhuhai, Zhuhai 519087, China
[2] Department of Mathematical Sciences, United International College (BNU-HKBU), Zhuhai 519087, China

Abstract

A prevalent problem in statistical signal processing, applied statistics, and time series analysis arises from the attempt to identify the hidden state of Markov process based on a set of available noisy observations. In the context of sequential data, filtering refers to the probability distribution of the underlying Markovian system given the measurements made at or before the time of the estimated state. In addition to the filtering, the smoothing distribution is obtained from incorporating measurements made after the time of the estimated state into the filtered solution. This work proposes a number of new filters and smoothers that, in contrast to the traditional schemes, systematically make use of the process noises to give rise to enhanced performances in addressing the state estimation problem. In doing so, our approaches for the resolution are characterized by the application of the graphical models; the graph-based framework not only provides a unified perspective on the existing filters and smoothers but leads us to design new algorithms in a consistent and comprehensible manner. Moreover, the graph models facilitate the implementation of the suggested algorithms through message passing on the graph.

Keywords: Probabilistic graphical model, Bayesian inference, Gaussian approximation filter and smoother, Cubature measure, Expectation propagation (EP)

1 Introduction

In a wide range of data science applications, such as tracking [1], navigation [2], and audio signal processing [3], one deals with the hidden Markov model that governs the dynamics of the latent process x_n and that establishes the relationship between the unobserved variable x_n and the observation y_n. Here $n \in \mathbb{N}$ signifies the discrete time. Let $Y_{n'} := \{y_1, \cdots, y_{n'}\}$ be the historical accumulation of data and let $x_{n|n'} := x_n|Y_{n'}$ be the conditioned random variable, then the Bayesian resolution for the unknown instance of x_n is given by the probability $P(x_{n|n'})$, which is called filtering when $n = n'$, smoothing when $n < n'$, and prediction when $n > n'$ [4]. In this paper concern is confined to the filtering and smoothing distributions.

In the simplest case of linear dynamics together with linear observations corrupted by an independent Gaussian noise, the filtering/smoothing problem can explicitly be solved by the Kalman filter/smoother which describes how the mean and covariance of the conditional probability evolve in the course of time [5–7]. For nonlinear systems where the conditioned measure cannot be characterized by a Gaussian probability, however, the target distributions do not permit closed-form expressions. In this case, various efforts have been made to obtain approximate solutions, see e.g., [5, 8–20]. Examples of traditional algorithms for filtering include the extended Kalman filter [14], the ensemble Kalman filter [15], unscented Kalman filter [21], cubature Kalman filter [22], Gaussian particle filter [23], the bootstrap filter and its variants [13, 16, 24, 25], and the Gaussian mixture filter [17, 18, 26, 27]. As for the smoothing, one can solve the problem by direct application of the Kalman filtering results because smoothing problem is a Kalman filtering problem in disguise (see, for instance, [4] and references therein). In recent years, significant attention has been directed toward the filtering/smoothing for the state estimation due to the remarkable success in various applications including medical tomography [28], geological tomography [29, 30], hydrology [31], petroleum engineering [32, 33], as well as a host of other physical, biological, or social systems [34–37].

Let w_n be the independent system noise that affects x_n, then the dependency among the relevant variables can typically be visualized through the directed graphical model in Fig. 1. This diagrammatic representation of the relationships among the random variables allows for the exact or approximate inference via turning the complex computation into a number of reduced operations in the graph [38, 39]. Notice that the graph-based approach has been already introduced to address the problem of sequential data (see, for instance, [19, 40–42] and references therein). Here it is important to point out that the classical filter and smoother never pay particular attention to the system noise w_n in estimating the hidden signal, given observations. One can verify the tendency using the graph model; in general any inference method can be associated with a suitable graphical model, and the class of data assimilation algorithms that can be associated with the graph in Fig. 1 without noises w_n encompasses most of the existing filters and smoothers [38, 39].

The goal of this work is to investigate the utilities of the previously ignored system noises in improving the state estimation skill of some well-known methods. Specifically, we build a new family of algorithms for which the driving noises play a more prominent and active role compared to the traditional schemes. The key idea for the development is to design a graphical model where, unlike the conventional graph, the process noises

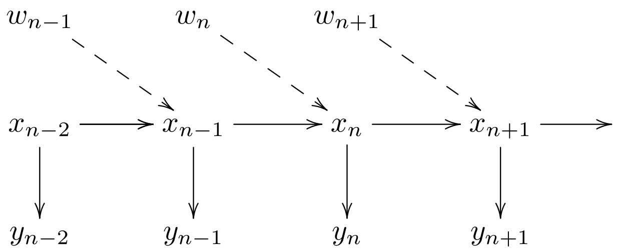


Fig. 1 A diagrammatic representation of the hidden Markov model, for which the underlying process x_n forms a Markov chain and emits a discrete time series of observation y_n. The dashed arrows are used to emphasize that, in the graph model associated with the classical algorithms, the independent system noises w_n influencing x_n are not shown

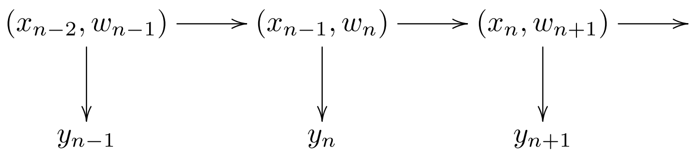


Fig. 2 The graphical model in Fig. 1 is transformed into the new graph, where the observation y_n is determined by x_{n-1} and w_n, instead of x_n

are explicitly visible and further integrated to the system variables so that they cannot be simply marginalized out but constitute an essential and integral part of the whole algorithm constructed according to the new graph. For instance, from the graph in Fig. 1, the graph model depicted in Fig. 2 can be obtained through augmenting the driving noise w_n to the system variable x_{n-1}. The resultant graph is in an equivalent form with the original graph without noises, and thus one can obtain the counterpart of a classic algorithm by using the variable $X_n := (x_{n-1}, w_n)^T$ in place of x_n. Here the upper T denotes transpose.

The move to the other graphical model appears to bring no advantage due to the invariance of the basic scenario, but on the contrary, we will show that the inference schemes derived from the new graph yield more accurate and stable estimations of the unobserved underlying system state. For this, we address two principal ways of data assimilation, and the context can be classified into two parts accordingly. Section 2 discusses the batch assimilation which involves processing the entire training set in one go, while Sect. 3 studies the data assimilation techniques in the sequential fashion. Differently from the batch setting where one has the opportunity to re-use the data points many times and to obtain an answer irrelevant to the order of data, the sequential Bayesian updating uses each data point on arrival and then discards it before receiving the next point. Section 4 contains the summary of our contributions and discussion on future works.

2 Batch data assimilation

Section 2.1 introduces the variational inference scheme known as the expectation propagation or EP for short (one who has knowledge of EP can skip this section). Building a new smoother in Sect. 2.2, we provide the theoretical arguments on why the proposed method is likely to outperform in Sect. 2.3, and present numerical simulation results supporting our demonstration in Sect. 2.4.

2.1 Inference using EP

Suppose that the Markov chain $\mathcal{X}_n$ and the associated data y_n are governed by the transition density and the observation distribution:

$$\mathcal{X}_n | \mathcal{X}_{n-1} \sim P(\cdot | \mathcal{X}_{n-1}), \tag{1a}$$

$$y_n | \mathcal{X}_n \sim P(\cdot | \mathcal{X}_n). \tag{1b}$$

Then the conditional distribution $P(\mathcal{X}_n | Y_N)$ can be approximated using the EP scheme as follows.

Together with the notations $\mathcal{X}_{p:q} := \{\mathcal{X}_p, \mathcal{X}_{p+1}, \cdots, \mathcal{X}_q\}$ and $\mathcal{X}_{0:1} := \mathcal{X}_1$, one has the factorization $P(\mathcal{X}_{1:N}|Y_N) \propto \prod_{n=1}^{N} \varphi_n(\mathcal{X}_{n-1:n})$ where the factor function $\varphi_n(\mathcal{X}_{n-1:n}) := P(\mathcal{X}_n|\mathcal{X}_{n-1})P(y_n|\mathcal{X}_n)$ is partially evaluated at the realization of y_n. The EP method then seeks an approximation of the (conditioned) joint distribution, which is in the form

$$q(\mathcal{X}_{1:N}) \propto \prod_{n=1}^{N} \widehat{\varphi}_n(\mathcal{X}_{n-1:n}) \tag{2}$$

where $\widehat{\varphi}_n$ is a Gaussian approximation of φ_n. The algorithm further assumes a Gaussian factorization of the element $\widehat{\varphi}_n = \beta_{n-1}(\mathcal{X}_{n-1})\alpha_n(\mathcal{X}_n)$ so that $q(\mathcal{X}_{1:N}) \propto \prod_{n=1:N} q_n(\mathcal{X}_n)$ is fully factorized and $q_n = \alpha_n \beta_n$.

Suppose that all of the factors in (2) are given by some Gaussian functions. Then EP recursively updates the factor $\widehat{\varphi}_n$ to a new Gaussian function $\widehat{\varphi}'_n$ (hereafter the prime will be used to denote the revised one). In order to do this, one first removes the relevant factor from approximate distribution (2) and then multiplies the exact factor φ_n to obtain $\widehat{q} := \varphi_n \prod_{j \neq n} \widehat{\varphi}_j$. One next evaluates the new posterior q' by minimizing the KL divergence of $\widehat{q}$ against q', given by $\int \widehat{q} \ln \left(\widehat{q}/q'\right)$. The result is that q' comprises the product of factors in which each factor is given by the corresponding marginal of $\widehat{q}$. To obtain the refined factor $\widehat{\varphi}'_n (= \beta'_{n-1}\alpha'_n)$, one simply divides q' by $\prod_{n \neq j} \widehat{\varphi}_n$.

Let $p'_n(\mathcal{X}_{n-1:n}) := \alpha_{n-1}\varphi_n\beta_n$ then the procedure involves an approximate marginalization denoted by

$$q'_m(\mathcal{X}_m) = \text{collapse} \int p'_n(\mathcal{X}_{n-1:n})\, \mathrm{d}\mathcal{X}_{\mathrm{n-1:n}} \backslash \mathcal{X}_{\mathrm{m}} \tag{3}$$

where \ denotes the set difference. Here m is either $m = n-1$ or $m = n$. The "collapse $\int$ "-operator performs projection to a Gaussian and marginalization over the states $\mathcal{X}_n$ or $\mathcal{X}_{n-1}$. From $q'_m = \alpha'_m \beta'_m$, the knowledge of (3) allows one to update β'_{n-1} and α'_n.

The difficulty encountered here is that, when p'_n is a non-Gaussian function, computation (3) is usually intractable. A particular way of resolving this issue leads to one instance of the EP algorithm. Here we introduce two state-of-the-art techniques for the Gaussian approximation of q'_m. One method is via the use of Gaussian cubature [41, 43]. Let $Q_n(\mathcal{X}_{n-1:n}) := \alpha_{n-1}\beta_{n-1}\alpha_n\beta_n$ be the proposal distribution and let $\mu_{Q_n} = \sum_j \lambda_j \delta_{\mathcal{X}^j_{n-1:n}}$ be a cubature rule for this Gaussian function. By virtue of the reweighting $\int g p'_n \, \mathrm{d}\mathcal{X}_{\mathrm{n-1:n}} = \int \mathrm{g} \frac{\mathrm{p}'_\mathrm{n}}{\mathrm{Q}_\mathrm{n}} \mathrm{Q}_\mathrm{n} \, \mathrm{d}\mathcal{X}_{\mathrm{n-1:n}}$, the discrete measure

$$\mu_{p'_n} = \sum_j \lambda'_j \delta_{\mathcal{X}^j_{n-1:n}} \quad \text{where} \quad \lambda'_j \propto \frac{p'_n(\mathcal{X}^j_{n-1:n})}{Q_n(\mathcal{X}^j_{n-1:n})} \lambda_j \tag{4}$$

is an approximation of the distribution p'_n. For the approximation of q'_m, we use the appropriate marginalization of the joint Gaussian distribution function whose mean and covariance are given by the ones from (4). We call this scheme EP–GC (expectation propagation–Gaussian cubature). The other method is to collapse the non-Gaussian two-slice posterior belief p'_n to a Gaussian form by Laplace approximation [40, 42, 44].

This can be achieved by the truncation of the Taylor expansion of p'_n at the second-order term [38]. We call the resulting scheme EP-Laplace.

Importantly, the update procedure can be cast in terms of local message passing on the graphical model. To be more precise, one can regard α_n and β_{n-1} as the messages between the factor function and the variable in the graph illustrated in Fig. 3. Though there is no required order, the usual EP implementation iteratively performs the message update via the forward and backward pass. During the forward pass, the α is updated, while the β remains fixed. During the backward pass, the β is updated, while the α remains fixed. The resulting algorithm reads as follows;

1. One initializes $\alpha_n(\mathcal{X}_n)$ and $\beta_n(\mathcal{X}_n)$ by suitable Gaussian functions ($\beta_N \equiv 1$);,
2. Until possible convergence, one continues to perform multiple forward–backward passes;,
   - Forward pass: update α_n as $\alpha'_n \propto q'_n/\beta_n$ in the ascending order of $n \in [1, N]$,
   - Backward pass: update β_{n-1} as $\beta'_{n-1} \propto q'_{n-1}/\alpha_{n-1}$ in the descending order of $n \in [2, N]$.

No convergence guarantees can be given for EP. It is however known that, in case of being convergent, the solution minimizes the Bethe free energy that takes into account two-point correlations between neighboring variables in the chain [38, 39, 45]. Eventually, the distribution function $P(\mathcal{X}_n|Y_N)$ is approximated by $q_n = \alpha_n\beta_n$, that is the product of two incoming messages into the circle node associated with the variable $\mathcal{X}_n$.

2.2 Proposed algorithm

Let the forward map for the hidden process and the likelihood function be given by

$$x_n = \phi^n(x_{n-1}, w_n), \quad w_n \sim \mathcal{N}(0, Q_n) \tag{5a}$$

$$P(y_n|x_n) = \mathcal{L}^{y_n}(x_n), \tag{5b}$$

and let the law of $x_1 := x_{1|0}$ be known. Here we intend to make use of EP in calculating the smoothing distribution $P(x_{n|N})$ for n ranging from 1 to N.

Notice that, in order to solve the estimation problem arised from Eq. (5), one typically applies EP to the graph in Fig. 3, for which $\mathcal{X}_n$ is given by x_n. It deserves to mention that this particular graph model can be viewed as the one converted from the graph in Fig. 1 without driving noises. Motivated by the relationship between Figs. 1 and 2, we develop a new version of EP from converting the graph in Fig. 2 into the one

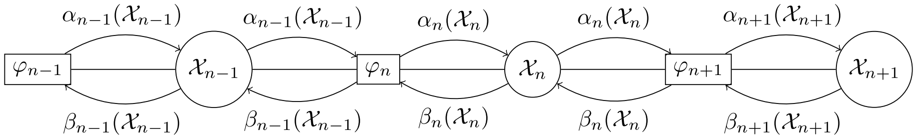

$\alpha_{n-1}(\mathcal{X}_{n-1})$
$\alpha_{n-1}(\mathcal{X}_{n-1})$
$\alpha_n(\mathcal{X}_n)$
$\alpha_n(\mathcal{X}_n)$
$\alpha_{n+1}(\mathcal{X}_{n+1})$
φ_{n-1}
$\mathcal{X}_{n-1}$
φ_n
$\mathcal{X}_n$
φ_{n+1}
$\mathcal{X}_{n+1}$
$\beta_{n-1}(\mathcal{X}_{n-1})$
$\beta_{n-1}(\mathcal{X}_{n-1})$
$\beta_n(\mathcal{X}_n)$
$\beta_n(\mathcal{X}_n)$
$\beta_{n+1}(\mathcal{X}_{n+1})$

Fig. 3 The factor graph for state space model (1) is shown. The circle node is occupied with the variable, whereas the square node is occupied with the factor function

in Fig. 3, for which $\mathcal{X}_n$ is given by $X_n = (x_{n-1}, w_n)^T$. More precisely, using the notation $x_n = \phi^n(X_n)$ instead of Eq. (5a), state space model (5) is reformulated into

$$X_n = \left(\phi^n(X_{n-1}), w_n \right)^T, \tag{6a}$$

$$P(y_n|X_n) = \mathcal{L}^{y_n} \circ \phi^n(X_n), \tag{6b}$$

and our proposed algorithm uses the transition kernel and likelihood governing state space model (6) in applying the EP method described in the preceding section.

One difference between the two methods is that, while the naive EP directly approximates $P(x_n|Y_N)$, our suggested scheme yields an approximation of $P(x_n, w_{n+1}|Y_N)$. Hence, in order to obtain the smoothing distribution of the system variable, additional marginalization needs to be performed. Because the task involves the integration against the noises conditioned on data, we refer to the new method as conditioned-noise EP or CNEP for short.

2.3 Discussion on the prospective performances

Based on the analytic approximation to the distribution function of interest by assuming that it factorizes into a particular way, the EP method carries out a variational inference through the iterated local optimization of a Kullback–Leibler (KL) divergence [19, 40]. For the problem of data assimilation, the original EP scheme seeks the approximate factorization of the probability distribution of the variables x_n, given observations. By contrast, our CNEP explicitly takes into account the noise w_n and regards it as the part of the system variable so that the new method corresponds to seeking the approximate factorization of the joint distribution function expressed in terms of the augmented variable $X_n = (x_{n-1}, w_n)^T$, conditioned on data. The critical effect by CNEP is that the reference measure in the KL divergence is replaced by the one containing the more information on the whole system state, and that the space of the candidate functions for the optimization using the KL divergence has been extended from the one for EP. It is therefore our belief that the CNEP enables a closer approach to the true distribution function, leading to the accuracy enhancement.

Furthermore, unlike the conventional EP where the driving noises are simply integrated out at early stage before conditioning on data, our CNEP produces solutions through averaging with respect to the conditioned noises; the procedure effectively weakens the potential information loss caused by the marginalization, giving rise to a similar effect with the fully Bayesian approach in machine learning where all of the involved variables are modeled and estimated according to Bayes' rule to yield a conditional-averaged result that is robust to the particular set of data [38, 39]. We thus anticipate the improvement of the batch assimilation performance by CNEP, compared to the use of EP, in the aspect of stability.

2.4 Numerical experiment

Let us consider the Poisson tracking model [41, 46]. The dynamic equation governs neural activity unfolding over time, and the spike counts within short time-bins are observed. The state space model is given by

$$x_n|x_{n-1} \sim \mathcal{N}(\Phi(x_{n-1}), Q),\ \ \Phi(x) = \alpha x + \beta\,\mathrm{erf}(x) \tag{7a}$$

$$y_n|x_n \sim \mathrm{Poisson}(\exp(x_n)). \tag{7b}$$

where erf(·) represents the error function. Here the transition kernel can be obtained from the forward map $\phi^n(x, w) = \Phi(x) + w$. Note that the data $y_n \in \{0, 1, 2, \cdots\}$ assume a non-negative integer according to a Poisson distribution, and that the likelihood is given by

$$\mathcal{L}^{y_n=k}(x) \propto \exp\left(kx - e^x\right). \tag{8}$$

Because Eq. (8) is a non-Gaussian function, rather than many other algorithms, it is plausible to apply the EP method to calculate the smoothing distribution [41, 46]. Equation (7) thus serves as a good example in comparing the performances of EP and CNEP

For the parameter values $\alpha = 0.9$, $\beta = 0.2$, $Q_n = 0.1$, $x_1 \sim \mathcal{N}(0.1, 0.1)$, and $N = 60$, we implement the EP-GC and the corresponding CNEP method, which we call CNEP-GC. To do this, we use the standard Gaussian cubature formula of degree 3 and 5 that can be found in [47], whose support size is $2k$ and $2k^2 + 2$ in case of k dimension. For the same problem data, we also implement EP-Laplace and the corresponding CNEP method, which we call CNEP-Laplace. We perform a total of 40 independent simulations, and we use the root-mean-square error (RMSE) to compare the various reconstructions of the evolving system state. Note the RMSE between $A = \{A_i\}_{i=1}^L$ and $B = \{B_i\}_{i=1}^L$ is defined by

$$\mathrm{RMSE}(A, B) = \sqrt{\tfrac{1}{L}\textstyle\sum_{i=1}^L |A_i - B_i|^2}$$

where A_i and B_i are vectors.

We first calculate RMSE between the true trajectories and the mean of the smoothing solutions for each time n and depict the resulting RMSE distances as the function of time in the top panels of Fig. 4. This result shows that CNEP yields more accurate estimates than EP. While one can see notable improvement in the case of EP–GC and CNEP–GC, the difference between the accuracies of EP-Laplace and CNEP-Laplace is not so much. It also shows that the use of higher-order cubature is advantageous in case of classical EP–GC, but the benefit in case of CNEP–GC is not significant.

We next calculate the averaged RMSEs over the entire time interval, denoted by aRMSE, for independent simulations. The mean and variance of aRMSE are presented in the bottom panels of Fig. 4. The reduced mean further ensures the accuracy improvement of CNEP against EP. We use the variance to address the stability issue; the variance reduction due to CNEP implies that the state estimation results are more robust compared to the ones by EP.

Because the simulation shows that the outperformance of CNEP-Laplace over EP-Laplace is not significant in both aspects of accuracy and stability, our noise-conditioned framework appears not so advantageous for the EP-Laplace. By contrast, it is indeed useful to apply CNEP–GC, rather than EP–GC, for the performance improvement. Considering the increased computational complexity in the implementation of CNEP–GC using Gaussian cubature degree 5, we conclude that CNEP–GC using Gaussian cubature

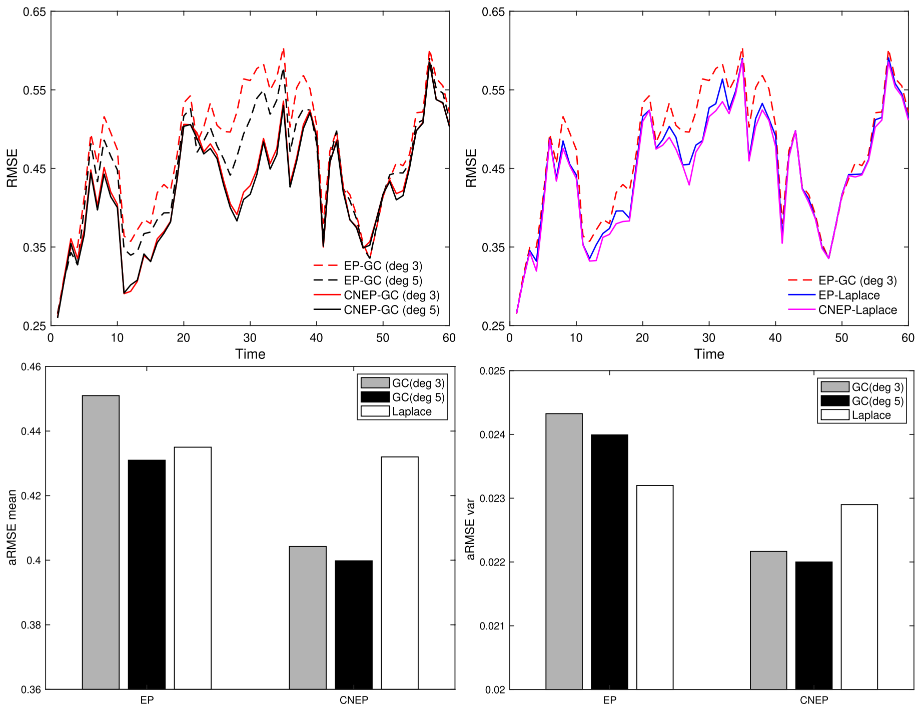


Fig. 4 A realization of the hidden signal x_n over time is generated from (7a) and compared with the smoothing approximations using observations from (7b). Top panel shows the RMSE distance between these two trajectories, and the bottom panel shows the mean and variance of averaged RMSE (aRMSE)

degree 3 is the optimal choice among the smoothing algorithms under consideration in the example of state space model (7).

3 Recursive Bayesian estimation

Section 3.1 describes the representative recursive data assimilation schemes called the (nonlinear) Kalman filter and smoother. Section 3.2 discusses a number of graph models and the associated filtering/smoothing algorithms. Section 3.3 provides new techniques for the sequential filter and smoother. Specifying how to implement the concerned algorithms in Sect. 3.4, their performances will be compared in Sect. 3.5.

3.1 Sequential filter and smoother

In what follows, our presentation is in the context of the state space model given by

$$x_n = \phi^n(x_{n-1}, w_n), \quad w_n \sim \mathcal{N}(0, Q_n) \tag{9a}$$

$$y_n = g^n(x_n) + \eta_n, \quad \eta_n \sim \mathcal{N}(0, R_n) \tag{9b}$$

where w_n and η_n are independently distributed centered Gaussians. Given the law of $x_1 := x_{1|0}$, the goal is to estimate the distribution of the conditioned variable $x_{n|n'}$. Here n ranges from 1 to N, and the case of either $n' = n$ or $n' = N$ will be considered depending on the problem of filtering and smoothing, respectively.

3.1.1 Forward filter

The typical approach adopted in most filters for a sequential estimation of the underlying system state is to recursively alternate the time update and the measurement update of the probability distribution [4, 8, 9]. Here and after, the notation $\nearrow/\searrow$ will be used to increase/decrease the first index of the conditioned variable by one, and the notation $\Rightarrow$ to increase the second index by one. For instance, $x_{n|m} \nearrow x_{n+1|m}$ and $x_{n|m} \Rightarrow x_{n|m+1}$. Then the pseudo-code for the filter is as follows.

```
For $n = 1$ to $N$, do
  If $n > 1$,
      • $Forward\ time\ update$: perform $x_{n-1|n-1} \nearrow x_{n|n-1}$ using (9a)
  end
      • $Measurement\ update$: perform $x_{n|n-1} \Rightarrow x_{n|n}$ using (9b)
end
```

To achieve a Gaussian approximation of the target distribution, the forward time update is implemented in a way that one obtains a Gaussian approximation of $(x_{n-1|n-1}, x_{n|n-1})$ and performs a suitable marginalization. As for the measurement update, one approximates the joint distribution of $(x_{n|n-1}, y_n)$ by Gaussian and applies Bayes' rule (A1) described in Appendix A in order to perform the conditioning $x_{n|n-1}|y_n := x_{n|n}$.

3.1.2 Backward smoother

While the sequential filter is in progress, the approximate distributions of $x_{n|n}$ are recursively obtained from $n = 1$ to $n = N$ in the increasing order. Once the process for filtering is over, the nonlinear Kalman smoother given in Appendix B can be applied to yield Gaussian approximations of $x_{n|N}$ in the order of decreasing index, from $n = N$ to $n = 1$. The pseudo-code reads as follows.

```
For $n = N - 1 : -1 : 1$, do
    • $Backward\ time\ update$: perform $x_{n+1|N} \searrow x_{n|N}$ using (B2)
end
```

Note the implementation requires the knowledge of the joint Gaussian distribution of $(x_{n|n}, x_{n+1|n})$, which was obtained and stored during the previous filtering procedure.

3.2 One-step ahead filter and smoother

Notice that the forward filter and backward smoother introduced in the preceding section are directly relevant to the graph in Fig. 1 without the noises w_n. More precisely, the forward–backward algorithm is the general graph-based method for the statistical inference via the message passing and reduces to the standard filter and smoother when it complies with the designated graph [38, 39].

In view of the same form of two directed graph models in Fig. 1 without noises and in Fig. 2, one is naturally interested in the forward–backward algorithm with respect to the graph in Fig. 2. Let us first describe the corresponding methods. In order to do that, state space model (9) governing (x_n, y_n) is posed as the one in terms of the variables (X_n, y_n):

$$X_n = \big(\phi^n(X_{n-1}), w_n \big)^T, \tag{10a}$$

$$y_n = g^n \circ \phi^n(X_n) + \eta_n. \tag{10b}$$

Denoting $X_{n|n'} := X_n|Y_{n'}$, one makes use of state space model (10) to perform the iteration as follows.

```
For $n = 1$ to $N$, do
  If $n = 1$, find $x_{1|1}$ and find $X_{2|1} = (x_{1|1}, w_2)$
  else
    • Measurement update: perform $X_{n|n-1} \Rightarrow X_{n|n}$ using (10b)
    • Forward time update: perform $X_{n|n} \nearrow X_{n+1|n}$ using (10a) and obtain
      Gaussian approximation of $(X_{n|n}, X_{n+1|n})$
  end
end
```

From suitable marginalizations of the outcomes $P(X_{n+1|n})$, the filtering distributions $P(x_{n|n})$ are obtained for $1 \le n \le N$. Likewise, the smoothing distributions $P(x_{n|N})$ can be calculated according to the following backward iteration.

```
For $n = N : -1 : 2$, do
    • Backward time update: perform $X_{n+1|N} \searrow X_{n|N}$ using (B2) and the
      covariance of $(X_{n|n}, X_{n+1|n})$ obtained via the forward time update
end
```

We next remark that the forward filter making use of reformulation (10) has been proposed in the author's prior work [48]. In the present paper the algorithm will be called the one-step ahead filter because, unlike the standard filter conducting the step $x_{n|n-1} \Rightarrow x_{n|n}$, its variant using Eq. (10) produces the filtering law of $x_{n|n}$ from the one-step ahead smoothing law of $x_{n-1|n}$ (similarly, the smoother using Eq. (10) will be called the one-step ahead smoother). There is a body of work that has demonstrated the advantage of this alternative path toward the filtering distribution in addressing the estimation problem arised in geophysical sciences [49–52]. Inspired by the practical relevance of the smoothing-based filter, we would like to improve the current version of our one-step look-ahead algorithms. This can be achieved by repeating essentially the same procedure with the one we did when Fig. 1 is converted to Fig. 2, and here we can take advantage of the graph-model framework.

Specifically one can read from the graph in Fig. 2 that, in carrying out the time update of the one-step ahead filter to quantify the uncertainty propagated by the statistical model, the prediction $x_{n-1|n} \nearrow x_{n|n}$ is pushed forward according to the law of $w_n|y_n$. This driving noise conditioned on future observation possesses a nonzero value as the mean, giving rise to an effective form of importance sampling so that the filtered solution tends to be nudged to the true system trajectory [48]. Now the transition from Figs. 1 to 2 creates a momentum for us to proceed to a new graphical model

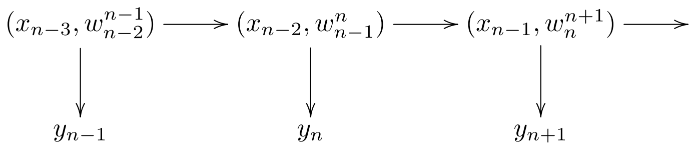


Fig. 5 The probabilistic graphical model that gives rise to the two-step ahead conditioned-noise filter and smoother. Here the notation $w_m^{m+1} = (w_m, w_{m+1})$ is used

presented in Fig. 5, and the philosophy lying behind this attempt is to strengthen the nudging effect; unlike the one-step ahead filter, the time update in the new graph drives the evolving estimate together with not only the conditioned noises $w_n|y_n$ but $w_{n-1}|y_n$ as well. Our concern is whether the noises conditioned on further future observations can give rise to a more accurate estimation of the hidden state. We address the issue after building this insight into the practical algorithm in the next section.

3.3 Proposed algorithm: Two-step ahead filter and smoother

Here we formulate the forward filter and backward smoother associated with the graph in Fig. 5. Let $\mathbb{X}_n := (X_{n-1}, w_n)^T = (x_{n-2}, w_{n-1}, w_n)^T$ and let Eq. (10a) be denoted by $X_n := \Phi^n(\mathbb{X}_n)$ then state space models (9), (10) can be posed as the one governing the joint variable $(\mathbb{X}_n, y_n)$:

$$\mathbb{X}_n = \left(\Phi^n(\mathbb{X}_{n-1}), w_n \right)^T, \tag{11a}$$

$$y_n = g^n \circ \phi^n \circ \Phi^n(\mathbb{X}_n) + \eta_n. \tag{11b}$$

Using the notation $\mathbb{X}_{n|n'} := \mathbb{X}_n|Y_{n'}$, the filtering of $x_{n|n}$ can be obtained via the following two-layer procedure. On the one hand, the recursive estimation $\mathbb{X}_{n|n-1} \Rightarrow \mathbb{X}_{n|n} \nearrow \mathbb{X}_{n+1|n}$ is performed. Meanwhile, on the other hand, the time update $\mathbb{X}_{n+1|n} \nearrow \mathbb{X}_{n+2|n}$ is carried out for filtering. This additional step is necessary because the law of $x_{n|n'}$ comes from the marginalization of $\mathbb{X}_{n+2|n'}$. Putting it together, the pseudo-code reads as follows.

```
For $n = 1$ to $N$, do
  If $n = 1$, find $\mathbb{X}_{3|1} = (x_{1|1}, w_2, w_3)$
  else if $n = 2$, find $\mathbb{X}_{3|2} = (x_{1|2}, w_{2|2}, w_3) = (X_{2|2}, w_3)$, and find $\mathbb{X}_{4|2}$
  else,
    • Measurement update: perform $\mathbb{X}_{n|n-1} \Rightarrow \mathbb{X}_{n|n}$ using (11b)
    • First time update: perform $\mathbb{X}_{n|n} \nearrow \mathbb{X}_{n+1|n}$ using (11a) and save the
      correlation of $(\mathbb{X}_{n|n}, \mathbb{X}_{n+1|n})$
    • Second time update: perform $\mathbb{X}_{n+1|n} \nearrow \mathbb{X}_{n+2|n}$ using (11a) and save the
      correlation of $(\mathbb{X}_{n+1|n}, \mathbb{X}_{n+2|n})$ when $n = N$
  end
end
```

The fixed-interval smoothing distributions $P(x_{n|N})$ can be obtained from the following iteration and from the marginalization of $\mathbb{X}_{n+2|N}$.

```
For $n = N+1 : -1 : 3$, do
    • Backward time update: perform $\mathbb{X}_{n+1|N} \searrow \mathbb{X}_{n|N}$ using (B2) and the correlation of $(\mathbb{X}_{n|n}, \mathbb{X}_{n+1|n})$ (when $n = N+1$, use the correlation of $(\mathbb{X}_{N+1|N}, \mathbb{X}_{N+2|N})$) obtained via the forward time update
end
```

The proposed algorithms are called the two-step ahead filter and smoother. In Fig. 6, the conditioned variables $x_{n|n'}$ resulting from our one-step and two-step ahead filters are illustrated to emphasize the algorithmic difference. As in [53], the two-step ahead filter produces the law of $x_{n|n}$ through the smoothing distributions of $x_{n-2|n}$ and $x_{n-1|n}$. However, the presence of conditioned system noises distinguishes our method from the prior works and the framework can account for the prospective out-performance of the derived algorithms in a more comprehensible fashion. Though the approach can straightforwardly be generalized to higher-order procedures, we do not proceed further due to increasing complexity.

3.4 Implementation using cubature measure

Notice that the equations in state space models (9) and (10), (11) can be cast into the common form of $\mathsf{Y} = \Phi(\mathsf{X})$ for an appropriate function $\Phi(\cdot)$ and a Gaussian random variable X. For instance, one can regard $\mathsf{X} = (X_{n-1}, w_n)$ and $\mathsf{Y} = X_n$ in case of Eq. (10a), and $\mathsf{X} = (X_n, \eta_n)$ and $\mathsf{Y} = y_n$ in case of Eq. (10b). Therefore, for the implementation of the algorithms discussed so far, it is sufficient to define how to obtain a Gaussian approximation of $\mathsf{Z} = (\mathsf{X}, \mathsf{Y})$.

The method adopted here is to pass a set of weighted points known as cubature points through the function and fit a Gaussian to the resulting transformed points. To be precise, let $\mu_{\mathsf{X}} = \sum_{\mathsf{i}} \lambda_{\mathsf{i}} \delta_{\mathsf{X}^{\mathsf{i}}}$ be the cubature with respect to X, that refers to a discrete measure in possession of the same moments with the distribution of X up to a certain degree. Let $\mathsf{Z}^{\mathsf{i}} = (\mathsf{X}^{\mathsf{i}}, \Phi(\mathsf{X}^{\mathsf{i}}))$ then the mean and covariance of $\mu_{\mathsf{Z}} = \sum_{\mathsf{i}} \lambda_{\mathsf{i}} \delta_{\mathsf{Z}^{\mathsf{i}}}$ are given by

$$\mathsf{M} = \sum_{\mathsf{i}} \lambda_{\mathsf{i}} Z^{i},$$

$$\Sigma = \sum_{\mathsf{i}} \lambda_{\mathsf{i}} \left(\mathsf{Z}^{\mathsf{i}} - \mathsf{M}\right)\left(\mathsf{Z}^{\mathsf{i}} - \mathsf{M}\right)^{\mathsf{T}}.$$

The probability of Z is approximated by the normal distribution $\mathcal{N}(\mathsf{M}, \Sigma)$. The method is easy to implement, giving rise to computational advantage, and further ensures a good degree of accuracy in filtering and smoothing applications [22, 47, 54, 55].

$$\begin{tikzcd}
{} \arrow[r] & x_{n-2|n-1} \arrow[r, Rightarrow] \arrow[d] & x_{n-2|n} \arrow[r] & x_{n-1|n} \arrow[r, Rightarrow] \arrow[d] & {} \\
 & x_{n-1|n-1} \arrow[urr, Rightarrow] & & x_{n|n} &
\end{tikzcd}$$

Fig. 6 Two different paths to the filtered solution are shown: (i) the one-step ahead filter ($x_{n-2|n-1} \rightarrow x_{n-1|n-1} \Rightarrow x_{n-1|n} \rightarrow x_{n|n}$) and (ii) the two-step ahead filter ($x_{n-2|n-1} \Rightarrow x_{n-2|n} \rightarrow x_{n-1|n} \rightarrow x_{n|n}$)

In the present paper, we call the filter and smoother that are introduced in Sect. 3.1 and that use the cubature measure, the *cubature Kalman filter* (CKF) and the *cubature Kalman smoother* (CKS). Similarly, we call the filters that are introduced in Sects. 3.2 and 3.3 and that use the cubature measure, the *conditioned-noise cubature Kalman filter of the first order* (CNCKF) and the *conditioned-noise cubature Kalman filter of the second order* (CN2CKF). The corresponding smoothers are named by CNCKS and CN2CKS.

3.5 Numerical simulation

Here we numerically perform a comparison analysis of the algorithms defined in the preceding section. Specifically, we are interested in the accuracy and stability of the cubature-based Gaussian approximation filters (CKF, CNCKF, CN2CKF) and smoothers (CKS, CNCKS, CN2CKS). Our test-bed is two benchmark examples in the context of sequential filtering and smoothing.

3.5.1 Target tracking

Consider a model air-traffic monitoring scenario, where an aircraft executes a maneuvering turn in a horizontal plane at an unknown turn rate Ω_n at time n [22, 48, 55]. The dynamical system is given by

$$x_{n+1} = \begin{pmatrix} 1 & \frac{\sin(\Omega_n \Delta t)}{\Omega_n} & 0 & \frac{\cos(\Omega_n \Delta t)-1}{\Omega_n} & 0 \\ 0 & \cos(\Omega_n \Delta t) & 0 & -\sin(\Omega_n \Delta t) & 0 \\ 0 & \frac{1-\cos(\Omega_n \Delta t)}{\Omega_n} & 1 & \frac{\sin(\Omega_n \Delta t)}{\Omega_n} & 0 \\ 0 & \sin(\Omega_n \Delta t) & 0 & \cos(\Omega_n \Delta t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} x_n + w_{n+1} \tag{12}$$

where $x_n = (\mathrm{x}_n, \dot{\mathrm{x}}_n, \mathrm{y}_n, \dot{\mathrm{y}}_n, \Omega_n)^T$; $(\mathrm{x}_n, \mathrm{y}_n)$ and $(\dot{\mathrm{x}}_n, \dot{\mathrm{y}}_n)$ are the position and velocity at time n, respectively. The system noise w_n is distributed according to a centered Gaussian with covariance

$$Q_n = \begin{pmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 & 0 & 0 \\ \frac{\Delta t^2}{2} & \Delta t & 0 & 0 & 0 \\ 0 & 0 & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 \\ 0 & 0 & \frac{\Delta t^2}{2} & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 1.75 \times 10^{-3} \Delta t \end{pmatrix}.$$

The measurement equation is given by

$$y_n = \begin{pmatrix} \sqrt{\mathrm{x}_n^2 + \mathrm{y}_n^2} \\ \tan^{-1}(\mathrm{y}_n / \mathrm{x}_n) \end{pmatrix} + \eta_n \tag{13}$$

and the noise covariance is $R_n = \mathrm{diag}(10^2, 10^{-5})$. The inter-observation time is $\Delta t = 1$ (hereafter the units of physical quantities are omitted for brevity).

The simulation results are based on 200 independent Monte Carlo runs. In each case, the initial state of the true signal is a draw from the normal distribution $x_1 \sim \mathcal{N}\Big((10^3, 3 \cdot 10^2, 10^3, 0, -\frac{3\pi}{180})^T, \mathrm{diag}([10^2, 10, 10^2, 10, 10^{-4}])\Big)$, and the sequential observations over the time interval $1 \leq n \leq 200$ are randomly generated.

We obtain a single trajectory of x_n over time by solving (12) and calculate the filtered/smoothed estimations using the observational data from (13). Figure 7 shows the RMSEs in each time step, with respect to position, velocity and turn rate. We see that (i) the performance of CNCKF/CNCKS is less sensitive to the cubature order compared to the case of CKF/CKS, and that (ii) the algorithms obtained by applying the conditioned-noise framework uniformly outperform the naive methods. Figure 8 depicts the mean and variance of aRMSE. From the reduced means in case of our algorithms, compared to the classical schemes, we argue that there are improvements in accuracy. Similarly, the reduced variances imply that the new algorithms are more stable than the classical schemes. Because the performances of CNCKS and CN2CKS are similar to each other, taking into account the computational burden, CNCKS is preferred to the other suggested algorithms.

3.5.2 Ballistic target

Let us consider the problem of tracking a ballistic target under the influence of drag and gravity acting on the target [54, 56, 57]. Let $x_n = (x_n^1, x_n^2, x_n^3)^T$ be the state vector, where x_n^1 and x_n^2 are altitude and velocity, respectively, and x_n^3 is a constant ballistic coefficient. The equation of motion is given by

$$x_{n+1} = \begin{pmatrix} x_n^1 - \delta x_n^2 \\ x_n^2 - \delta e^{-\gamma x_n^1}(x_n^2)^2 x_n^3 + g \\ x_n^3 \end{pmatrix}. \tag{14}$$

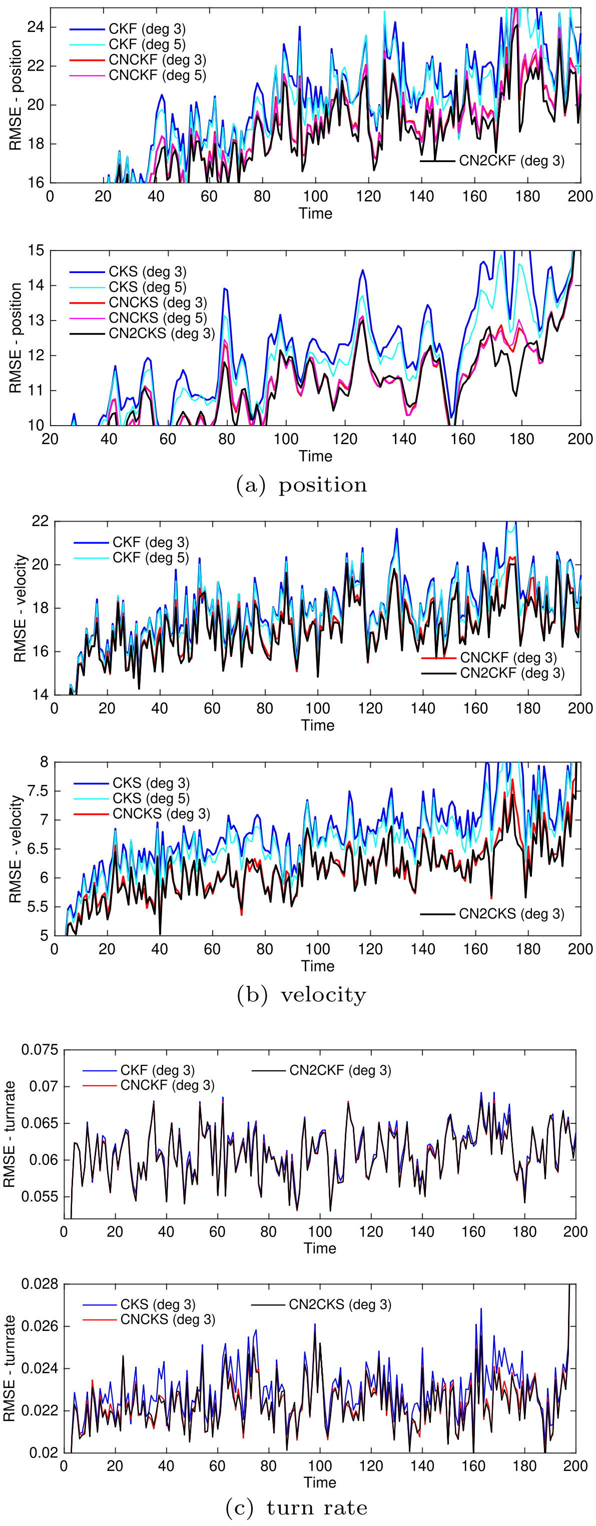
here $\delta = 0.5$ is the integration time, $\gamma = 1.49 \times 10^{-4}$ and $g = 9.81$ serve as the drag and gravity constants.

The measurement equation is given by

$$y_n = \sqrt{M^2 + (x_n^1 - H)^2} + \eta_n \tag{15}$$

where M is the horizontal distance, and H determines the radar location. The system is characterized by the parameters $H = 10^3$, $M = 10^4$ and $R_n = 30^2$. The true initial state is $x_1^* = (61 \cdot 10^3, 3048, 4.49 \cdot 10^{-4})^T$, and the initial state density is $\mathcal{N}\big((62 \cdot 10^3, 3400, 10^{-5})^T, \mathrm{diag}([10^6, 10^4, 10^{-4}])\big)$.

We generate a single trajectory of x_n over time by solving (14) and obtain the filtered/smoothed estimations using the observational data from (15). We perform 1800 independent simulations. The illustrations in Figs. 9, 10 allow us to demonstrate the out-performance of the conditioned-noise framework over the classical methods. Figure 9 depicts the RMSE values as the function of time, showing that (i) CN2CKF is always better than CNCKF, and (ii) CN2CKS is in general better than CNCKS but occasionally does not lead to an improved performance. This numerical simulation reveals that, quite interestingly, the outperformance of our framework holds even for the system without explicit driving noise. This result accords with the prior work provided in [53].

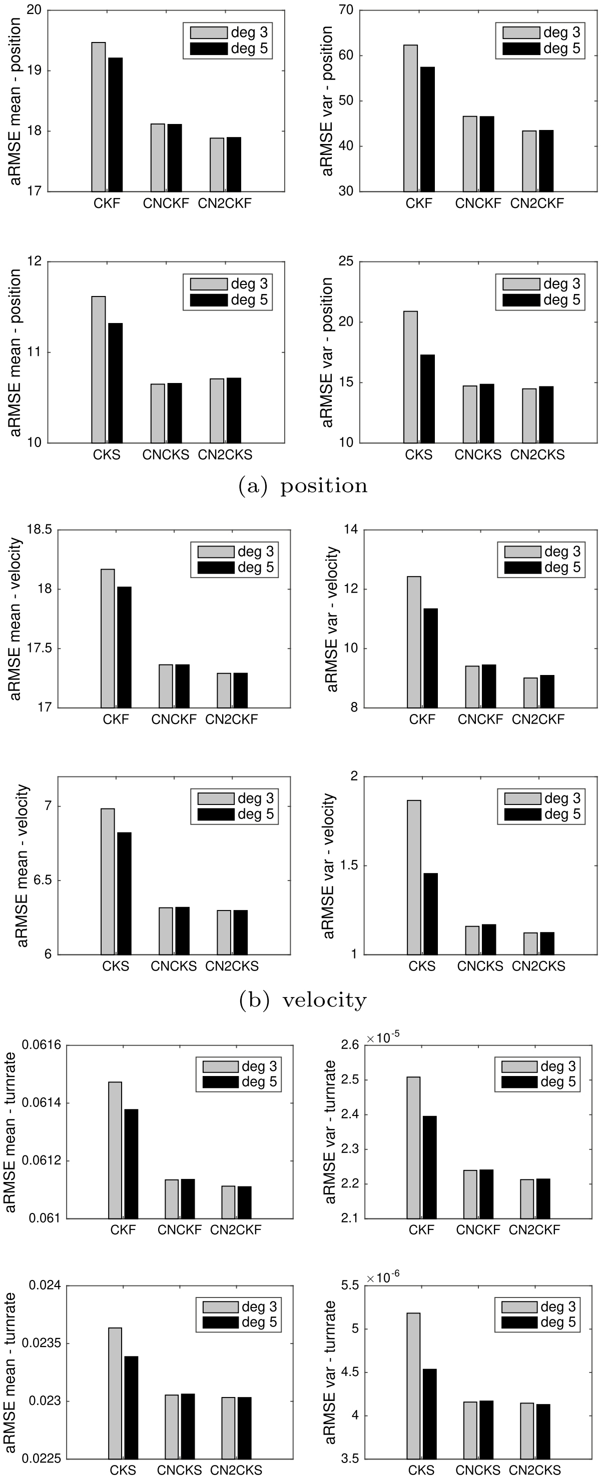


(a) position

(b) velocity

(c) turn rate

Fig. 7 The RMSEs between the target trajectory and the filtering/smoothing estimates obtained from the average over 200 independent simulations; **a** position; **b** velocity; and **c** turn rate

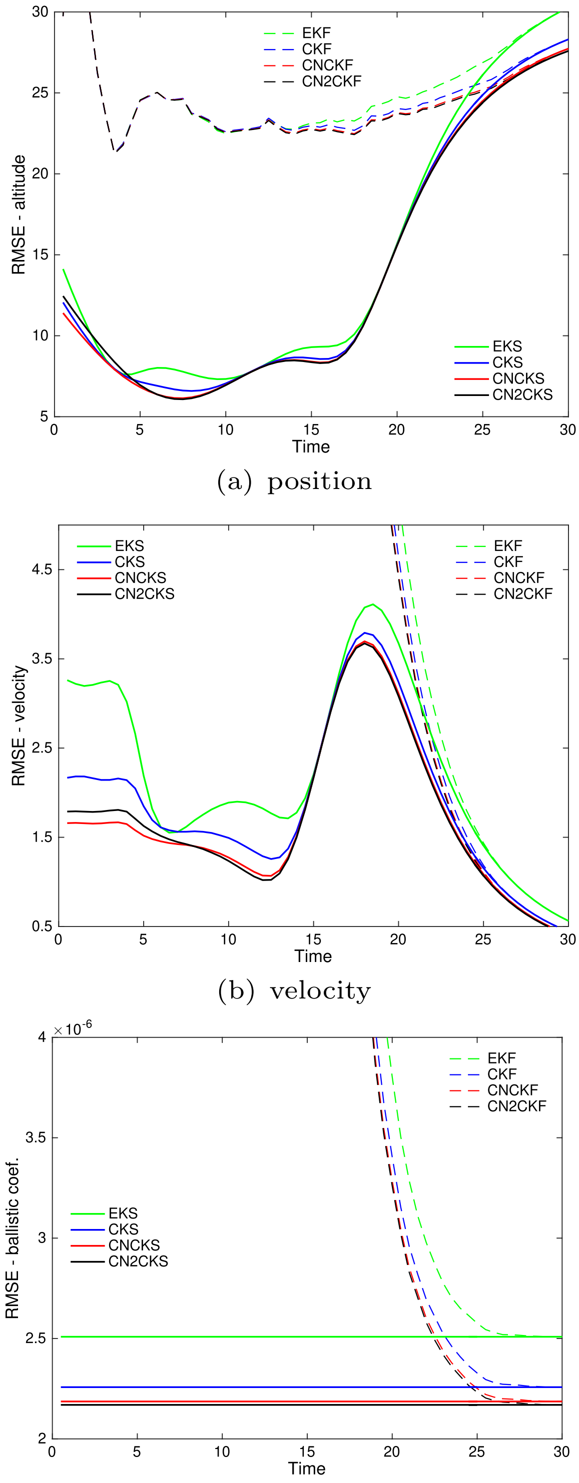


(a) position

(b) velocity

(c) turn rate

Fig. 8 The mean and variance of time-averaged RMSEs between the target trajectory and filtered/smoothed solutions; **a** position; **b** velocity; and **c** turn rate

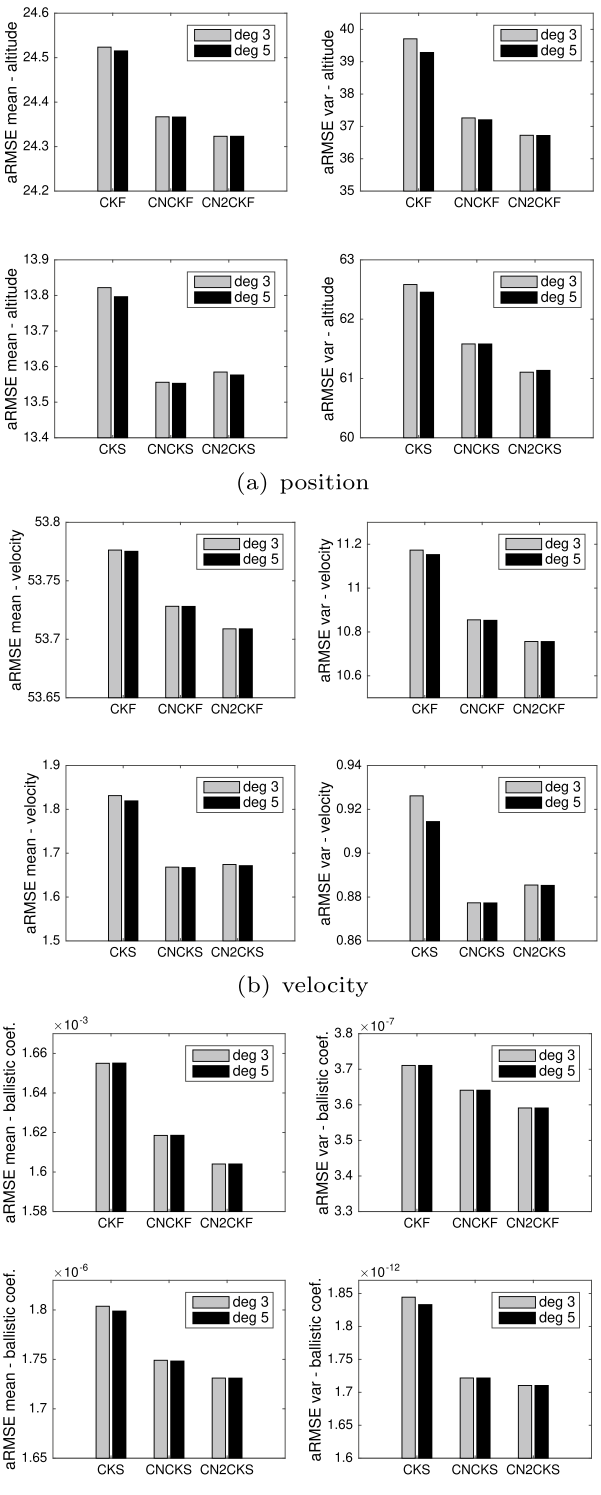


(a) position

(b) velocity

(c) turn rate

Fig. 9 The RMSEs between the target trajectory and the filtering/smoothing estimates obtained from the average over 1800 independent simulations; **a** position; **b** velocity; and **c** turn rate. Here EKF/EKS represents the extended Kalman filter/smoother



(a) position

(b) velocity

(c) turn rate

Fig. 10 The mean and variance of time-averaged RMSEs between the target trajectory and filtered/smoothed solutions; **a** position; **b** velocity; and **c** turn rate

4 Conclusion

This paper considers the design of batch smoother and sequential filter/smoother for discrete-time nonlinear systems with Gaussian noise. The new family of algorithms for the state estimation are proposed as follows;

1. For state space model (5),
   - we develop the conditioned-noise version of expectation propagation in Sect. 2.2,
2. For state space model (9),
   - we develop the conditioned-noise smoother of the first order in Sect. 3.2,
   - we develop the conditioned-noise filter and smoother of the second order in Sect. 3.3.

Note the development is achieved by the reformulation of the original state space model into the one governing the augmented variable comprising the system variable and driving noise. The implementation of the proposed algorithm is basically the same as the existing method, and the difference is that the conditioned-noise expectation propagation makes use of (6), in place of (5), and the conditioned-noise filter/smoother makes use of (10) and (11), in place of (9). The numerical simulations performed in Sects. 2.4, 3.5 confirm that, in any of the benchmark examples studied in the areas of batch and sequential data assimilation, the filters and smoothers developed according to the conditioned-noise framework uniformly outperform the corresponding classical methods in both aspects of accuracy and stability. We emphasize that this result from the numerical analysis is in accordance with the theoretical reasoning by considering the role of the conditioned noise in Sects. 2.3, 3.2.

Throughout the text, we investigate Gaussian approximation of the target probability distribution. It is our belief that the conditioned-noise framework remains to be competitive even when the distribution function is parametrized in a different way. In future works, therefore, we plan to develop Gaussian mixture and sequential Monte Carlo algorithms for filtering and smoothing, which are similar in spirit to the conditioned-noise Gaussian filters and smoothers proposed in this work. With a goal to extend the applicability of the graph-based method beyond the scope that explored in this paper, we also plan to pursue the direction where an improved performance of data assimilation is sought via introducing a new graphical model with particular characteristics and considering an inference scheme in the graph.

We finally discuss the pathway to the impact of this work to the academia and industry. Our effort to shed a new light on the driving noises can bring advantages to the academic community by creating new research momentum in devising a data assimilation technique on the basis of the graphical model. Enriching the library of the filters/smoothers directly applicable to solving real-world problems, our research result will be beneficial for industrial progress where the need for accurate and stable filtering/smoothing schemes is paramount.

Appendix A: Bayes' rule for conditional Gaussian

Let $Z = \begin{bmatrix} X \\ Y \end{bmatrix}$ be distributed as a Gaussian with the mean $\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}$ and covariance $\begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}$. Then Bayes' rule asserts that the law of $X|Y$ with $Y = y$ is Gaussian with the mean and covariance given by

$$\begin{aligned} \bar{x}' &= \bar{x} + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \bar{y}), \\ \Sigma'_{xx} &= \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}, \end{aligned} \tag{A1}$$

respectively [4].

Appendix B: Nonlinear Kalman smoother

Let $\mathsf{x}_{n|n'} := \mathsf{x}_n|Y_{n'}$ then, based on the Gaussian assumption $\mathsf{x}_{n|n'} \sim \mathcal{N}(\bar{\mathsf{x}}_{n|n'}, C_{n|n'})$, one can derive

$$\begin{aligned} \bar{\mathsf{x}}_{n|N} &= \bar{\mathsf{x}}_{n|n} + G_n(\bar{\mathsf{x}}_{n+1|N} - \bar{\mathsf{x}}_{n+1|n}), \\ C_{n|N} &= C_{n|n} + G_n(C_{n+1|N} - C_{n+1|n})G_n^T, \end{aligned} \tag{B2}$$

where $G_n = D_{n+1}C_{n+1|n}^{-1}$ and $D_{n+1} = \mathrm{cov}(\mathsf{x}_{n|n}, \mathsf{x}_{n+1|n})$ [4, 7]. Recurrence relation (B2) enables the backward time update $\mathsf{x}_{n+1|N} \searrow \mathsf{x}_{n|N}$, provided the joint Gaussian law of $(\mathsf{x}_{n|n}, \mathsf{x}_{n+1|n})$ is known.

Acknowledgements
The author thanks the anonymous referees for their helpful comments and suggestions, which indeed contributed to improving the quality of the publication.

Author contributions
Not applicable.

Funding
This work is supported by the Start-up Research Grant of Beijing Normal University (No. 28704-310432107), the United International College Start-up Research Fund (No. UICR0700040-22) and the National Natural Science Fund of China (NSFC) Research Fund for International Excellent Young Scientists (No. 12250610190).

Availability of data and materials
Not applicable.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
Not applicable.

Received: 29 November 2023 Accepted: 1 April 2024
Published online: 10 April 2024

References

1. S. Särkkä, V.V. Viikari, M. Huusko, K. Jaakkola, Phase-based uhf rfid tracking with nonlinear kalman filtering and smoothing. Sens. J. IEEE **12**(5), 904–910 (2012)
2. D. Titterton, J.L. Weston, Strapdown Inertial Navigation Technology vol. 17. IET (2004)
3. S. Godsill, P. Rayner, O. Cappé, *Digital Audio Restoration* (Springer, 2002)

4. B.D.O. Anderson, J.B. Moore, *Optimal Filtering*, vol. 11 (Prentice-hall Englewood Cliffs, NJ, 1979)
5. R.E. Kalman et al., A new approach to linear filtering and prediction problems. J. Basic Eng. **82**(1), 35–45 (1960)
6. R.E. Kalman, R.S. Bucy, New results in linear filtering and prediction theory. J. Basic Eng. **83**(3), 95–108 (1961)
7. S. Särkkä, *Bayesian Filtering and Smoothing*, vol. 3 (Cambridge University Press, 2013)
8. H. Kushner, Approximations to optimal nonlinear filters. IEEE Trans. Autom. Control **12**(5), 546–556 (1967)
9. A. Jazwinski, *Stochastic Processes and Filtering Theory*, Mathematics in science and engineering, vol. 64. (Academic Press, San Diego, California, 1970)
10. A.F. Bennett, *Inverse Methods in Physical Oceanography* (Cambridge University Press, 1992)
11. E. Kalnay, Atmospheric Modeling, Data Assimilation, and Predictability (2003)
12. D.S. Oliver, A.C. Reynolds, N. Liu, *Inverse Theory for Petroleum Reservoir Characterization and History Matching* (Cambridge University Press, 2008)
13. A. Doucet, N. De Freitas, N. Gordon, *Sequential Monte Carlo Methods in Practice* (Springer Verlag, 2001)
14. A. Gelb, *Applied Optimal Estimation* (MIT Press, 1974)
15. G. Evensen, *Data Assimilation: The Ensemble Kalman Filter* (Springer Verlag, 2009)
16. N.J. Gordon, D.J. Salmond, A.F.M. Smith, in *Novel Approach to Nonlinear/Non-gaussian Bayesian State Estimation*, IEE Proceedings F Radar and Signal Processing, vol. 140 (IET, 1993), pp. 107–113
17. R. Chen, J.S. Liu, Mixture Kalman filters. J. R. Stat. Soc. Ser. B (Statistical Methodology) **62**(3), 493–508 (2000)
18. A.S. Stordal, H.A. Karlsen, G. Nævdal, H.J. Skaug, B. Vallès, Bridging the ensemble Kalman filter and particle filters: the adaptive gaussian mixture filter. Comput. Geosci. **15**(2), 293–305 (2011)
19. T.P. Minka, in *Expectation Propagation for Approximate Bayesian inference*, Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, (Morgan Kaufmann Publishers Inc, 2001), pp. 362–369
20. S. Razali, K. Watanabe, S. Maeyama, K. Izumi, in *An Unscented Rauch-Tung-Striebel Smoother for a Bearing only Tracking Problem*, ICCAS 2010, (IEEE, 2010), pp. 1281–1286
21. S.J. Julier, J.K. Uhlmann, Unscented filtering and nonlinear estimation. Proc. IEEE **92**(3), 401–422 (2004)
22. I. Arasaratnam, S. Haykin, Cubature Kalman filters. IEEE Trans. Autom. Control **54**(6), 1254–1269 (2009)
23. J.H. Kotecha, P.M. Djuric, Gaussian sum particle filtering. IEEE Trans. Signal Process. **51**(10), 2602–2612 (2003)
24. M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. Signal Process. **50**(2), 174–188 (2002)
25. O. Cappé, E. Moulines, T. Rydén, *Inference in Hidden Markov Models* (Springer, 2005)
26. I. Hoteit, D. Pham, G. Triantafyllou, G. Korres, A new approximate solution of the optimal nonlinear filter for data assimilation in meteorology and oceanography. Mon. Weather Rev. **136**(1), 317–334 (2008)
27. I. Hoteit, X. Luo, D.-T. Pham, Particle Kalman filtering: a nonlinear Bayesian framework for ensemble Kalman filters*. Mon. Weather Rev. **140**, 528–542 (2012)
28. I.S. Weir, Fully Bayesian reconstructions from single-photon emission computed tomography data. J. Am. Stat. Assoc. **92**(437), 49–60 (1997)
29. R.P. Barry, M. Jay, V. Hoef, Blackbox kriging: spatial prediction without specifying variogram models. J. Agric. Biol. Environ. Stat. 297–322 (1996)
30. R. Glaser, G. Johannesson, S. Sengupta, B. Kosovic, S. Carle, G. Franz, R. Aines, J. Nitao, W. Hanley, A. Ramirez, et al, Stochastic engine final report: Applying markov chain monte carlo methods with importance sampling to large-scale data-driven simulation, Technical report, (Lawrence Livermore National Lab., Livermore, CA, 2004)
31. H.K. Lee, D.M. Higdon, Z. Bi, M.A. Ferreira, M. West, Markov random field models for high-dimensional parameters in simulations of fluid flow in porous media. Technometrics **44**(3), 230–241 (2002)
32. P.S. Craig, M. Goldstein, J.C. Rougier, A.H. Seheult, Bayesian forecasting for complex systems using computer simulators. J. Am. Stat. Assoc. **96**(454), 717–729 (2001)
33. B.K. Hegstad, O. Henning et al., Uncertainty in production forecasts based on well observations, seismic data, and production history. Spe J. **6**(04), 409–424 (2001)
34. P.K. Kitanidis, Parameter uncertainty in estimation of spatial functions: Bayesian analysis. Water Resour. Res. **22**(4), 499–507 (1986)
35. F. Liu, M. Bayarri, J. Berger, R. Paulo, J. Sacks, A Bayesian analysis of the thermal challenge problem. Comput. Methods Appl. Mech. Eng. **197**(29–32), 2457–2466 (2008)
36. D.M. Schmidt, J.S. George, C.C. Wood, Bayesian inference applied to the electromagnetic inverse problem. Hum. Brain Mapp. **7**(3), 195–212 (1999)
37. J. Wang, N. Zabaras, Hierarchical Bayesian models for inverse problems in heat conduction. Inverse Probl. **21**(1), 183 (2004)
38. C.M. Bishop et al., *Pattern Recognition and Machine Learning*, vol. 4 (Springer, New York, 2006)
39. K.P. Murphy, *Machine Learning: A Probabilistic Perspective* (MIT press, 2012)
40. A. Ypma, T. Heskes, Novel approximations for inference in nonlinear dynamical systems using expectation propagation. Neurocomputing **69**(1), 85–99 (2005)
41. B.M. Yu, K.V. Shenoy, M. Sahani, in *Expectation Propagation for Inference in Non-linear Dynamical Models with Poisson Observations*, Nonlinear Statistical Signal Processing Workshop, (IEEE, 2006) pp. 83–86
42. M. Deisenroth, S. Mohamed, Expectation propagation in gaussian process dynamical systems. Adv. Neural Inf. Process. Syst. **25** (2012)
43. O. Zoeter, A. Ypma, T. Heskes, in *Improved Unscented Kalman Smoothing for Stock Volatility Estimation*, Proceedings of the 2004 14th IEEE Signal Processing Society Workshop Machine Learning for Signal Processing, (IEEE, 2004) pp. 143–152
44. M.P. Deisenroth, R.D. Turner, M.F. Huber, U.D. Hanebeck, C.E. Rasmussen, Robust filtering and smoothing with gaussian processes. IEEE Trans. Autom. Control **57**(7), 1865–1871 (2011)
45. T.P. Minka, From hidden markov models to linear dynamical systems. Technical report, Tech. Rep. 531, (Vision and Modeling Group of Media Lab, MIT, 1999)
46. B. Yu, A. Afshar, G. Santhanam, S.I. Ryu, K. Shenoy, M. Sahani, Extracting dynamical structure embedded in neural activity. Adv. Neural Inf. Process. Syst. **18**, 1545 (2006)

47. B. Jia, M. Xin, Y. Cheng, High-degree cubature Kalman filter. Automatica **49**(2), 510–518 (2013)
48. W. Lee, C. Farmer, Data assimilation by conditioning of driving noise on future observations. IEEE Trans. Signal Process. **62**(15), 3887–3896 (2014)
49. M.E. Gharamti, B. Ait-El-Fquih, I. Hoteit, in *A One-Step-Ahead Smoothing-Based Joint Ensemble Kalman Filter for State-Parameter Estimation of Hydrological Models*, Dynamic Data-Driven Environmental Systems Science, (Springer, 2015), pp. 207–214
50. M. Gharamti, B. Ait-El-Fquih, I. Hoteit, An iterative ensemble kalman filter with one-step-ahead smoothing for state-parameters estimation of contaminant transport models. J. Hydrol. **527**, 442–457 (2015)
51. B. Ait-El-Fquih, M. El Gharamti, I. Hoteit, A bayesian consistent dual ensemble Kalman filter for state-parameter estimation in subsurface hydrology. Hydrol. Earth Syst. Sci **20**, 3289–3307 (2016)
52. N.F. Raboudi, B. Ait-El-Fquih, I. Hoteit, Ensemble Kalman filtering with one-step-ahead smoothing. Mon. Weather Rev. **146**(2), 561–581 (2018)
53. F. Desbouvries, Y. Petetin, B. Ait-El-Fquih, Direct, prediction-and smoothing-based Kalman and particle filter algorithms. Signal Process. **91**(8), 2064–2077 (2011)
54. I. Arasaratnam, S. Haykin, Cubature Kalman smoothers. Automatica **47**(10), 2245–2250 (2011)
55. B. Jia, M. Xin, *Rauch-Tung-Striebel High-Degree Cubature Kalman Smoother*, American Control Conference (ACC), (IEEE, 2013), pp. 2472–2477
56. S.J. Julier, J.K. Uhlmann, A general method for approximating nonlinear transformations of probability distributions. Technical report, Technical report, Robotics Research Group, Department of Engineering Science, (University of Oxford, 1996)
57. B. Ristic, S. Arulampalam, N. Gordon, Beyond the Kalman filter. IEEE Aerosp. Electron. Syst. Mag. **19**(7), 37–38 (2004)

Publisher's Note