Approximate Inference and Learning of State Space Models with Laplace Noise

Julian Neri, Philippe Depalle, Roland Badeau

Approximate Inference and Learning of State Space Models with Laplace Noise

Julian Neri, *Member, IEEE,* Philippe Depalle, and Roland Badeau, *Senior Member, IEEE*

***Abstract*—State space models have been extensively applied to model and control dynamical systems in disciplines including neuroscience, target tracking, and audio processing. A common modeling assumption is that both the state and data noise are Gaussian because it simplifies the estimation of the system's state and model parameters. However, in many real-world scenarios where the noise is heavy-tailed or includes outliers, this assumption does not hold, and the performance of the model degrades. In this paper, we present a new approximate inference algorithm for state space models with Laplace-distributed multivariate data that is robust to a wide range of non-Gaussian noise. Locally exact inference is combined with an expectation propagation algorithm, leading to filtering and smoothing that outperforms existing approximate inference methods for Laplace-distributed data, while retaining a fast speed similar to the Kalman filter. Further, we present a maximum posterior expectation maximization (EM) algorithm that learns the parameters of the model in an unsupervised way, automatically avoids over-fitting the data, and provides better model estimation than existing methods for the Gaussian model. The quality of the inference and learning algorithms are exemplified through a diverse set of experiments and an application to non-linear tracking of audio frequency.**

***Index Terms*—Bayesian inference, time series, heavy-tailed noise, EM algorithm, machine learning, expectation propagation**

I. Introduction

STATE space models are probabilistic representations for sequential data that have proven beneficial in a wide range of disciplines such as control systems, audio processing, and neuroscience. In state space models, a sequence of observable data is assumed to have been generated from a latent variable sequence that evolves over time according to a first-order Markov chain. These latent variable models can represent a great diversity of dynamical behavior, and are customizable with respect to whether the dynamics are linear or non-linear and the choice of transition probability distribution and output emission noise. The Kalman filter and smoother are efficient and numerically accurate algorithms for exactly inferring the latent state's posterior statistics and marginal likelihood for a model with Gaussian state and observation noise [1]. However, time-series data often exhibit non-Gaussian noise, consisting of outliers, glint noise [2], sensor failure, and extended periods of drastically increased noise levels [3]. The Kalman filter's performance severely degrades in all of these cases because its mean squared error objective is suitable for short-tailed, compactly distributed data.

Models that assume non-Gaussian, heavy-tailed observation noise are better at representing outlier data [4] and have proven beneficial in many real world applications. Such applications include target tracking, analyzing biological signals, and mechanical vibration analysis. Non-Gaussian models, such as those that assume heavy-tailed data, do not admit closed-form recursive filtering or smoothing equations, making exact inference intractable. Sequential Monte Carlo (SMC) methods like particle filtering can estimate posterior statistics for arbitrary state space models [5], however, there is no limit to the number of samples needed to attain a certain degree of estimation quality, they are subject to the curse of dimensionality, and it is hard to evaluate the reliability of their estimates.

Recently, there has been much interest in fast and reliable deterministic estimators for state space models with heavy-tailed observation noise. Deterministic methods typically exploit a recursive structure akin to the Kalman filter for speed. Such methods may use an alternative cost function to the Kalman filter such as the maximum correntropy filter [6], heuristics and optimization algorithms minimax-based filters [4], [7], or variational inference, a principled approach to deterministic approximate inference that turns inference into an optimization problem [8]–[11].

In particular, the Laplace distribution is a heavy-tailed distribution that has proven applicable to tasks such as outlier filtering, sparse regression, modeling glint noise and speech spectra [12], and differential privacy [13] [14]. State space models with Laplace-distributed observation noise have proven robust to extreme outliers and a variety of other heavy-tailed-distributed noises. Existing methods for inference of state space models with Laplace distributed noise rely on iterative optimization algorithms to approximate the posterior, including convex optimization [15], majorization-minimization [16], [17], Huber cubature filtering [18], and variational inference with Gaussian scale mixtures [9].

In this paper, we formulate comprehensive inference and learning algorithms for state space models with Laplace-distributed data. For inference, we propose a recursive filtering and smoothing algorithm based on expectation propagation (EP) that is shown to be superior in quality to existing methods like variational inference and SMC, while being of comparable speed to the Kalman filter and smoother. Its high quality is attributed, in part, to the availability of an analytic solution to the exact posterior for a univariate observation. We extend the analytic solution from [19] to multivariate data through expectation propagation. The automatic learning of model

J. Neri is with McGill University, Montréal, Canada. E-mail: julian.neri@mcgill.ca

P. Depalle is with McGill University, Montréal, Canada.

R. Badeau is with LTCI, Télécom Paris, Institut Polytechnique de Paris, France.

parameters is addressed using an expectation maximization (EM) algorithm. We provide all the equations necessary to update the model parameters to maximize the expected log likelihood of the data and log prior probabilities, *i.e.* maximize the a posteriori probability. These equations include new closed-form analytic update equations for the output noise and a Newton step for the observation matrix. The inference and learning is robust to a wide range of noise and, due to the Laplace distribution, naturally regulates the estimated precision of the output noise to avoid over-fitting the data.

This paper is organized as follows. The state space model with Laplace observation noise is defined in Section II. Section III reviews approximate inference methods for non-Gaussian state space models. Section IV presents and validates the new Bayesian filtering algorithm for the Laplace model, then Section V presents the smoothing algorithm. Section VI extends the filter to non-linear dynamical models. Section VII covers the automatic learning of the Laplace model's parameters using an EM algorithm. Extensive results that validate the methods presented in the paper are reported and discussed in Section VIII. Section IX concludes the paper and proposes avenues of future work.

Notation

This following notation is used throughout the paper:

- bold lowercase denotes a vector and bold uppercase denotes a matrix,
- $\mathbf{x}^\mathsf{T}$: transpose of $\mathbf{x}$,
- $\mathbf{x}_{1:n}$ denotes the set $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$, and $\mathbf{X}$ is the full set $\mathbf{x}_{1:N}$,
- $x_{i,n}$: the ith element of $\mathbf{x}_n$,
- $\mathbf{C}_{(i)}$: the ith row of $\mathbf{C}$,
- $C_{(i,j)}$: the (i,j)th entry of $\mathbf{C}$,
- $\langle \mathbf{x} \rangle$: expected value of $\mathbf{x}$,
- $\text{cov}[\mathbf{x}, \mathbf{y}] = \langle \mathbf{x}\mathbf{y}^\mathsf{T} \rangle - \langle \mathbf{x} \rangle \langle \mathbf{y} \rangle^\mathsf{T}$: covariance of $\mathbf{x}$ and $\mathbf{y}$,
- $\int f(\mathbf{x}) d\mathbf{x}$: short for $\int_{-\infty}^{+\infty} \ldots \int_{-\infty}^{+\infty} f(\mathbf{x}) dx_1 \ldots dx_D$,
- $\mathbf{I}$: identity matrix,
- $\text{diag}(\mathbf{a})$: diagonal matrix formed from the elements of $\mathbf{a}$,
- $|.|$: absolute value of a scalar,
- $\det(.)$: determinant of a matrix,
- $q_{\setminus i}$: cavity distribution for likelihood term i,
- $q(x; y)$: probability of x with respect to y,
- $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$: multivariate Gaussian density with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$,
- $\text{Lap}(x|a, b)$: Laplace density with mean a and scale b [13],
- $\text{Gam}(x|a, b)$: Gamma density with shape a and rate b [13].

II. Probabilistic Model

State space models assume that a sequence of observable M-dimensional data $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$ are generated from a latent variable sequence of D-dimensional states $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ whose probabilistic dynamics are governed by a first-order Markov chain. The joint probability of all the observed data and latent states is

$$p(\mathbf{Y}, \mathbf{X}) = p(\mathbf{y}_1|\mathbf{x}_1)p(\mathbf{x}_1) \prod_{n=2}^{N} p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}) \,, \quad (1)$$

where $p(\mathbf{y}_n|\mathbf{x}_n)$ is the emission probability, $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ is the transition probability, and $p(\mathbf{x}_1)$ is the initial state's prior probability.

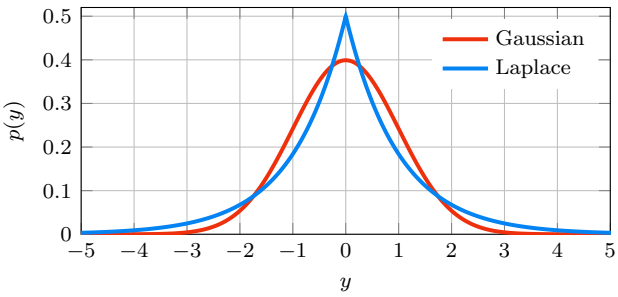


Fig. 1. Probability density functions for the Gaussian and Laplace distributions. The Gaussian's variance and Laplace's scale are set to one.

Linear time-invariant state space models can be described by the following state and observation equations,

$$\mathbf{x}_n = \mathbf{A}\mathbf{x}_{n-1} + \boldsymbol{\varepsilon}_n^x \,, \quad (2)$$

$$\mathbf{y}_n = \mathbf{C}\mathbf{x}_n + \boldsymbol{\varepsilon}_n^y \,, \quad (3)$$

where the states are transformed over adjacent times by $D \times D$ system dynamics matrix $\mathbf{A}$, and output to the observable space after being transformed through $M \times D$ output matrix $\mathbf{C}$. A common assumption is that both the state noise $\boldsymbol{\varepsilon}_n^x$ and the observation noise $\boldsymbol{\varepsilon}_n^y$ are drawn from Gaussian distributions because it enables exact inference.

The Laplace state space model (LSSM) assumes that the latent state $\mathbf{x}_n$ is corrupted by additive noise drawn from a Gaussian distribution with $D \times D$ covariance matrix $\mathbf{Q}$, while the observation $\mathbf{y}_n$ is corrupted by additive noise drawn from the Laplace distribution [13]. Diagonalizing the observation noise covariance through a linear operation, the emission distribution is expressed as a product of M univariate Laplace distributions, corresponding to the M dimensions of $\mathbf{y}_n$,

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n|\mathbf{A}\mathbf{x}_{n-1}, \mathbf{Q}) \,, \quad (4)$$

$$p(\mathbf{y}_n|\mathbf{x}_n) = \prod_{i=1}^{M} \text{Lap}(y_{i,n}|\mathbf{C}_{(i)}\mathbf{x}_n, R_i) \quad (5)$$

$$= \prod_{i=1}^{M} \frac{1}{2R_i} \exp\left(-\frac{|y_{i,n} - \mathbf{C}_{(i)}\mathbf{x}_n|}{2R_i} \right) . \quad (6)$$

Initial state $\mathbf{x}_1$ has a Gaussian distribution with $D \times 1$ mean $\mathbf{m}_0$ and $D \times D$ covariance $\mathbf{P}_0$,

$$p(\mathbf{x}_1) = \mathcal{N}(\mathbf{x}_1|\mathbf{m}_0, \mathbf{P}_0) \,. \quad (7)$$

Figure 1 shows that, in contrast to the Gaussian probability density function (PDF), the Laplace PDF has a heavy tail and sharp peak.

III. Approximate Inference

State estimation and model learning of state space models both rely on applying Bayes' theorem to infer the statistics of the posterior marginals $p(\mathbf{x}_n|\mathbf{Y})$ and pairwise posterior marginals $p(\mathbf{x}_n, \mathbf{x}_{n+1}|\mathbf{Y})$. Exact inference is carried out by Bayes' theorem,

$$p(\mathbf{X}|\mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X})p(\mathbf{X})}{p(\mathbf{Y})}\,. \tag{8}$$

For the linear Gaussian state space model (GSSM), exact inference of the posterior and pairwise posterior marginals is made tractable by the forward-backward algorithm, specifically the Kalman filter [1] and Rauch-Tung-Striebel (RTS) smoother [20]. Exact inference is intractable for any state space model that assumes non-Gaussian state and/or observation noise.

Approximate inference is required for the LSSM because the noise is non-Gaussian, and thus non-conjugate to the Gaussian transition probability. We denote the approximation to the posterior distribution as q,

$$q(\mathbf{X};\mathbf{Y}) \approx p(\mathbf{X}|\mathbf{Y})\,, \tag{9}$$

where semi-colon notation makes explicit the dependence on the data but distinguishes q from a true conditional density.

A. Approximate inference in state space models

Filtering algorithms compute the (approximate) posterior marginal of $\mathbf{x}_n$ given all the data up to that time, $q(\mathbf{x}_n;\mathbf{y}_{1:n})$, while smoothing algorithms incorporate future observations to provide $q(\mathbf{x}_n;\mathbf{Y})$ and $q(\mathbf{x}_n, \mathbf{x}_{n+1};\mathbf{Y})$.

Approximate inference in filtering and smoothing algorithms are carried out by either stochastic sampling methods, like particle filters, or deterministic methods, like variational Bayesian inference (VB). Deterministic methods are desirable because they are generally much faster and easier to interpret than sampling algorithms. One particularly popular method used for deterministic approximate inference in filters with non-Gaussian data is assumed density filtering (ADF) [21]. ADF approximates the marginal posterior at each time $p(\mathbf{x}_n|\mathbf{y}_{1:n})$ with a Gaussian, enabling a fast recursive structure analogous to the Kalman filter.

B. Expectation Propagation

Expectation propagation (EP) [22] is an approximate inference algorithm for minimizing the *forward* Kullback-Leibler (KL) divergence from the posterior $p(\mathbf{x}|\mathbf{y})$ to an approximate distribution $q(\mathbf{x};\mathbf{y})$,

$$\mathrm{KL}(p\|q) = -\int p(\mathbf{x}|\mathbf{y}) \ln \frac{q(\mathbf{x};\mathbf{y})}{p(\mathbf{x}|\mathbf{y})} d\mathbf{x}. \tag{10}$$

In contrast, variational Bayesian inference (VB) minimizes the *reverse* KL divergence from $q(\mathbf{x};\mathbf{y})$ to $p(\mathbf{x}|\mathbf{y})$, $\mathrm{KL}(q\|p)$ [23] [24] [25]. EP can provide better quality approximations than VB because minimizing the forward KL divergence is mean-seeking and inclusive, finding the tightest fit around the full distribution p, while minimizing the reverse KL divergence is mode-seeking and exclusive, finding a fit around a mode of p [26].

When $q(\mathbf{x};\mathbf{y})$ is in the exponential family (*e.g.* Gaussian) [27], the forward KL divergence from $p(\mathbf{x}|\mathbf{y})$ to $q(\mathbf{x};\mathbf{y})$ is minimized when the moments of $q(\mathbf{x};\mathbf{y})$ match those of $p(\mathbf{x}|\mathbf{y})$. Calculating the moments of the posterior requires the marginal probability $p(\mathbf{y})$, which involves integrating out $\mathbf{x}$ from the joint distribution and is not analytically possible for all but the simplest models. But joint distributions are commonly products of factors, including conditional distributions and a prior:

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x})\prod_{i=1}^{M} p_i(\mathbf{y}|\mathbf{x})}{p(\mathbf{y})} \tag{11}$$

For example, when the observations are independent, the joint distribution is a product of likelihood terms over each dimension of the observation, so $p_i(\mathbf{y}|\mathbf{x}) = p(y_i|\mathbf{x})$.

Expectation propagation circumvents the intractable integral involved in calculating $p(\mathbf{y})$ by approximating the posterior also as a product of approximate factors. These factors include the prior $q_0(\mathbf{x};\mathbf{y}) = p(\mathbf{x})$, and likelihood terms $q_i(\mathbf{x};\mathbf{y})$, $\forall i \in [1..M]$. The posterior is approximated as

$$q(\mathbf{x};\mathbf{y}) = \frac{\prod_{i=0}^{M} q_i(\mathbf{x};\mathbf{y})}{\int \prod_{i=0}^{M} q_i(\mathbf{x};\mathbf{y}) d\mathbf{x}} \tag{12}$$

where the approximate posterior and factors $q_i(\mathbf{x};\mathbf{y})$ are exponential family distributions (*e.g.* Gaussian).

The main difficulty in EP is computing the moments of the approximate posterior distribution $q(\mathbf{x};\mathbf{y})$. Although the factorization over likelihood terms simplifies the problem, for some models this computation will still be intractable. However, certain models admit analytic expressions for the posterior moments after factorizing the likelihood terms. In practice, VB is fast and applicable to arbitrary models because it does not require the analytic expression for the evidence $p(\mathbf{y})$. EP can be implemented efficiently depending on the model and has been proven to offer a closer approximation to the true posterior when compared to VB.

Expectation propagation provides a means of approximate inference for LSSMs because the likelihood function is a product of univariate distributions over each data dimension and the posterior moments can be computed analytically for a univariate Laplace likelihood with a Gaussian prior.

IV. Filtering

In this section, we derive the recursive filtering algorithm for the Laplace state space model. Filtering in state space models refers to inferring the marginal posterior probability of state $\mathbf{x}_n$ given every observation up to time n, $p(\mathbf{x}_n|\mathbf{y}_{1:n})$. Filtering propagates forward in time, at each time computing the predictive distribution (13), assessing the marginal likelihood (14), and updating the marginal posterior (15).

$$p(\mathbf{x}_n|\mathbf{y}_{1:n-1}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1}) p(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}) d\mathbf{x}_{n-1} \tag{13}$$

$$p(\mathbf{y}_n|\mathbf{y}_{1:n-1}) = \int p(\mathbf{y}_n|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{1:n-1}) d\mathbf{x}_n \tag{14}$$

$$p(\mathbf{x}_n|\mathbf{y}_{1:n}) = \frac{p(\mathbf{y}_n|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{1:n-1})}{p(\mathbf{y}_n|\mathbf{y}_{1:n-1})} \tag{15}$$

For linear GSSMs this is an exact inference algorithm, called the Kalman filter [1], [21], [28]. However, for any model with non-Gaussian noise, exact inference over all times is not analytically tractable and thus requires approximations. In [19], we derived the exact marginal posterior for a univariate data LSSM at time n per Equation (15) given that the posterior from time $n-1$ is approximated by a Gaussian q with mean $\boldsymbol{\mu}_{n-1}$ and covariance $\mathbf{V}_{n-1}$. After describing the prediction step, we present the expectation propagation update step that embeds exact univariate inference.

A. Prediction

First, given the previous time approximate posterior, the predictive distribution is

$$\begin{aligned} q(\mathbf{x}_n; \mathbf{y}_{1:n-1}) &= \int p(\mathbf{x}_n|\mathbf{x}_{n-1}) q(\mathbf{x}_{n-1}; \mathbf{y}_{1:n-1}) d\mathbf{x}_{n-1} \\ &= \mathcal{N}(\mathbf{x}_n|\mathbf{m}_{n-1}, \mathbf{P}_{n-1}) \,, \end{aligned} \tag{16}$$

where the predictive mean and covariance are

$$\mathbf{m}_{n-1} = \mathbf{A}\boldsymbol{\mu}_{n-1} \,, \tag{17}$$

$$\mathbf{P}_{n-1} = \mathbf{A}\mathbf{V}_{n-1}\mathbf{A}^{\mathsf{T}} + \mathbf{Q} \,. \tag{18}$$

Initially the mean and covariance are $\mathbf{m}_0$ and $\mathbf{P}_0$. This is equivalent to the Kalman filter's prediction step because the transition probability is Gaussian and the previous time posterior was approximated by a Gaussian.

B. Approximate inference from multivariate data

An expectation propagation (EP) algorithm can be applied elegantly to update the filter from multivariate Laplace distributed time series data because it factors multivariate inference into a collection of univariate inferences that can be computed exactly through analytic expressions. After performing the prediction step at time n, EP iterates over each likelihood term p_i to find an approximation to the marginal posterior $q(\mathbf{x}_n|\mathbf{y}_{1:n})$. Considering the LSSM, likelihood term p_i is defined as a univariate Laplace distribution over dimension i of the data,

$$p_i(\mathbf{y}_n|\mathbf{x}_n) \triangleq \text{Lap}(y_{i,n}|\mathbf{C}_{(i)}\mathbf{x}_n, R_i) \tag{19}$$

In practice, it is convenient to use natural parameters for combining and marginalizing the Gaussian likelihood terms in EP[1]. Natural parameters of the Gaussian include $D \times 1$ location parameter $\boldsymbol{\ell}$ and $D \times D$ precision matrix $\boldsymbol{\Lambda}$ [27].

First, the natural parameters of each likelihoood term are initialized: the location is $\boldsymbol{\ell}_i = \mathbf{0}$ and the precision is $\boldsymbol{\Lambda}_i = \mathbf{I}$, $\forall i \in [1..M]$. The prior predictive location is $\boldsymbol{\ell}_0 = \mathbf{P}_{n-1}^{-1}\mathbf{m}_{n-1}$ and the precision is $\boldsymbol{\Lambda}_0 = \mathbf{P}_{n-1}^{-1}$. The global location is $\widehat{\boldsymbol{\ell}} = \sum_{i=0}^{M} \boldsymbol{\ell}_i$ and the precision is $\widehat{\boldsymbol{\Lambda}} = \sum_{i=0}^{M} \boldsymbol{\Lambda}_i$.

Next, a cavity distribution $q_{\setminus i}(\mathbf{x}_n; \mathbf{y}_{1:n})$ is created by removing $q_i(\mathbf{x}; \mathbf{y}_{1:n})$ from the global $q(\mathbf{x}; \mathbf{y}_{1:n})$ and normalizing,

$$q_{\setminus i}(\mathbf{x}_n; \mathbf{y}_{1:n}) \propto \prod_{j \neq i} q_j(\mathbf{x}_n; \mathbf{y}_{1:n}) = \frac{q(\mathbf{x}_n; \mathbf{y}_{1:n})}{q_i(\mathbf{x}_n; \mathbf{y}_{1:n})} \,. \tag{20}$$

[1]For more information about natural parameters of the exponential family as it relates to this paper, see the supplementary material [29].

In terms of natural parameters, this step is completed by subtraction:

$$\boldsymbol{\ell}_{\setminus i} = \widehat{\boldsymbol{\ell}} - \boldsymbol{\ell}_i \,, \tag{21}$$

$$\boldsymbol{\Lambda}_{\setminus i} = \widehat{\boldsymbol{\Lambda}} - \boldsymbol{\Lambda}_i \,, \tag{22}$$

where $\boldsymbol{\ell}_{\setminus i}$ and $\boldsymbol{\Lambda}_{\setminus i}$ denote the natural parameters of $q_{\setminus i}(\mathbf{x}_n; \mathbf{y}_{1:n})$, and are converted into mean $\mathbf{m}_{\setminus i} = \boldsymbol{\Lambda}_{\setminus i}^{-1}\boldsymbol{\ell}_{\setminus i}$ and covariance $\mathbf{P}_{\setminus i} = \boldsymbol{\Lambda}_{\setminus i}^{-1}$. Parameters of the cavity distribution encode information from all the likelihood terms (including the prior) other than term p_i.

Likelihood term $p_i(\mathbf{y}_n|\mathbf{x}_n)$ is combined with the cavity distribution to form a hybrid joint distribution,

$$p(\mathbf{y}_n, \mathbf{x}_n) = \text{Lap}(y_{i,n}|\mathbf{C}_{(i)}\mathbf{x}_n, R_i)\mathcal{N}(\mathbf{x}_n|\mathbf{m}_{\setminus i}, \mathbf{P}_{\setminus i}) \,. \tag{23}$$

Therefore, this hybrid joint distribution now encodes information about all the terms and the prior. Using Bayes' theorem to express the hybrid posterior distribution gives

$$p(\mathbf{x}_n|\mathbf{y}_n) = \frac{p(\mathbf{y}_n, \mathbf{x}_n)}{p(\mathbf{y}_n)} = \frac{p(\mathbf{y}_n, \mathbf{x}_n)}{\int p(\mathbf{y}_n, \mathbf{x}_n) d\mathbf{x}_n} \,. \tag{24}$$

Global approximate posterior $q(\mathbf{x}_n; \mathbf{y}_{1:n})$ is found by minimizing the forward KL divergence $\text{KL}\left(p(\mathbf{x}_n|\mathbf{y}_n) \| q(\mathbf{x}_n; \mathbf{y}_{1:n})\right)$. For any approximating distribution $q(\mathbf{x}_n; \mathbf{y}_{1:n})$ in the exponential family, the forward KL divergence is minimized when the moments of $q(\mathbf{x}_n; \mathbf{y}_{1:n})$ match those of $p(\mathbf{x}_n|\mathbf{y}_n)$. Therefore, we compute the moments $\boldsymbol{\mu}_n$ and $\mathbf{V}_n$ of the posterior $p(\mathbf{x}_n|\mathbf{y}_n)$ and match the moments of $q(\mathbf{x}_n; \mathbf{y}_{1:n})$ to them. Section IV-C details the analytic solutions to the exact posterior moments.

Finally, the approximate posterior corresponding to likelihood term i conditioned on the prior p_0 is updated as

$$q_i(\mathbf{x}_n; \mathbf{y}_{1:n}) = Z_i \frac{q(\mathbf{x}_n; \mathbf{y}_{1:n})}{q_{\setminus i}(\mathbf{x}_n; \mathbf{y}_{1:n})} \,, \tag{25}$$

where Z_i normalizes $q_i(\mathbf{x}_n; \mathbf{y}_{1:n})$ such that $\int_{-\infty}^{+\infty} q_i(\mathbf{x}_n; \mathbf{y}_{1:n}) d\mathbf{x}_n = 1$. In practice, Equation (25) is computed by converting the moments back to natural parameters,

$$\widehat{\boldsymbol{\ell}} = \mathbf{V}_n^{-1}\boldsymbol{\mu}_n \,, \tag{26}$$

$$\widehat{\boldsymbol{\Lambda}} = \mathbf{V}_n^{-1} \,, \tag{27}$$

then updating the parameters of approximate term q_i,

$$\boldsymbol{\ell}_i = \widehat{\boldsymbol{\ell}} - \boldsymbol{\ell}_{\setminus i} \,, \tag{28}$$

$$\boldsymbol{\Lambda}_i = \widehat{\boldsymbol{\Lambda}} - \boldsymbol{\Lambda}_{\setminus i} \,. \tag{29}$$

This completes one update to the approximate posterior term q_i and the global approximate posterior q. One complete iteration involves repeating this for each approximate term q_i. EP scales linearly with the number of likelihood terms, $\mathcal{O}(M)$, and is non-iterative for $M = 1$.

Optionally, the marginal model evidence quantifies the model's fit to the given data and may be approximated by[2]

$$p(\mathbf{y}_n|\mathbf{y}_{1:n-1}) \approx \int \prod_{i=0}^{M} q_i(\mathbf{x}_n; \mathbf{y}_{1:n}) d\mathbf{x_n} \,. \tag{30}$$

[2]Equations for computing the filter's approximate model evidence are provided in the supplementary material [29].

Although EP is not guaranteed to converge if the initialization is too far away from the algorithm's fixed point [22], for our well-defined problem EP tends to converge within M iterations over each term.

C. Exact inference for a univariate Laplace likelihood

In this section, we derive locally exact Bayesian inference for a univariate Laplace likelihood conditioned on a Gaussian prior. This procedure is embedded into the expectation propagation to infer the latent state statistics from multivariate observations, as required by Equations (23) and (24).

The hybrid marginal likelihood is found by integrating out $\mathbf{x}_n$ from the numerator of Bayes' theorem[3].

$$\begin{aligned} p(y_{i,n}) &= \int p(y_{i,n}|\mathbf{x}_n) q_{\setminus i}(\mathbf{x}_n; \mathbf{y}_{1:n}) d\mathbf{x}_n \\ &= \int \text{Lap}(y_{i,n}|\mathbf{C}_{(i)}\mathbf{x}_n, R_i) \mathcal{N}(\mathbf{x}_n|\mathbf{m}_{\setminus i}, \mathbf{P}_{\setminus i}) d\mathbf{x}_n \\ &= \frac{\Phi_{i,n}^{(-)} + \Phi_{i,n}^{(+)}}{4R_i} \exp\left(-\frac{\widetilde{y}_{i,n}^2}{2S_{i,n}}\right) \end{aligned} \tag{31}$$

where we have defined

$$\widehat{y}_{i,n} = \mathbf{C}_{(i)}\mathbf{m}_{\setminus i}\,, \tag{32}$$

$$\widetilde{y}_{i,n} = y_{i,n} - \widehat{y}_{i,n}\,, \tag{33}$$

$$S_{i,n} = \mathbf{C}_{(i)}\mathbf{P}_{\setminus i}\mathbf{C}_{(i)}^{\mathsf{T}}\,, \tag{34}$$

$$\Phi_{i,n}^{(\pm)} = \text{erfcx}\left(\frac{\sqrt{S_{i,n}}}{\sqrt{2R_i^2}} \pm \frac{\widetilde{y}_{i,n}}{\sqrt{2S_{i,n}}}\right). \tag{35}$$

The scaled complementary error function $\text{erfcx}(x) \triangleq e^{x^2}\text{erfc}(x)$ is available in many programming languages. It avoids underflow and overflow errors associated with directly computing the product of e^{x^2} and the complementary error function $\text{erfc}(x)$ as defined in [30].

Observation $\mathbf{y}_n$ has the following mean and covariance with respect to the marginal likelihood $p(y_{i,n})$:

$$\langle y_{i,n} \rangle = \widehat{y}_{i,n}\,, \tag{36}$$

$$\text{cov}[y_{i,n}, y_{i,n}] = S_{i,n} + 2R_i^2. \tag{37}$$

Finally, state $\mathbf{x}_n$ has the following mean and covariance with respect to the locally exact marginal posterior $p(\mathbf{x}_n|\mathbf{y}_{1:n})$:

$$\langle \mathbf{x}_n \rangle = \mathbf{m}_{\setminus i} + \mathbf{k}_n \delta_{i,n}\,, \tag{38}$$

$$\text{cov}[\mathbf{x}_n, \mathbf{x}_n] = \mathbf{P}_{\setminus i} + \mathbf{k}_n \mathbf{k}_n^{\mathsf{T}} \Delta_{i,n}\,, \tag{39}$$

where we have defined the $D \times 1$ gain $\mathbf{k}_n$, scalar $\delta_{i,n}$, and scalar $\Delta_{i,n}$ as

$$\mathbf{k}_n = \mathbf{P}_{\setminus i}\mathbf{C}_{(i)}^{\mathsf{T}} R_i^{-1}\,, \tag{40}$$

$$\delta_{i,n} = \frac{\Phi_{i,n}^{(-)} - \Phi_{i,n}^{(+)}}{\Phi_{i,n}^{(-)} + \Phi_{i,n}^{(+)}}\,, \tag{41}$$

$$\Delta_{i,n} = \frac{1}{\Phi_{i,n}^{(-)} + \Phi_{i,n}^{(+)}} \left(\frac{4\Phi_{i,n}^{(-)}\Phi_{i,n}^{(+)}}{\Phi_{i,n}^{(-)} + \Phi_{i,n}^{(+)}} - \sqrt{\frac{8R_i^2}{\pi S_{i,n}}}\right). \tag{42}$$

[3]Properties of Gaussian integrals related to this derivation are detailed in the supplementary material [29].

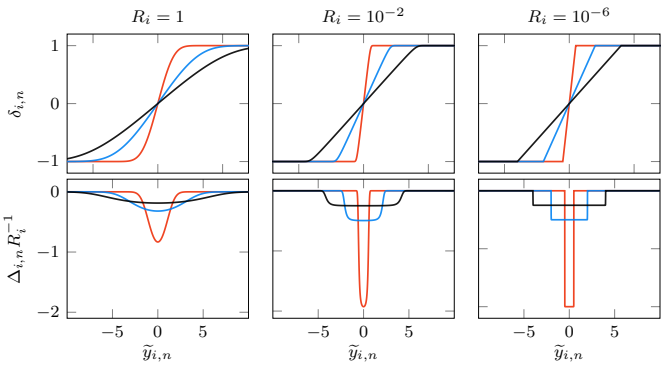


Fig. 2. Plots of $\delta_{i,n}$ and $\Delta_{i,n}R_i^{-1}$. The red, blue, and black lines correspond to the ratios $\mathbf{P}_{\setminus i}R_i^{-1} = \frac{1}{2}$, 2, and 4, respectively, and $\mathbf{C}_{(i)} = 1$. Here, the prior is univariate Gaussian so $\mathbf{P}_{\setminus i}$ and $\mathbf{C}_{(i)}$ are scalar.

To gain intuition into the mechanisms that render the LSSM robust to outliers and heavy-tailed noise, it is beneficial to investigate the behavior $\delta_{i,n}$ and $\Delta_{i,n}$. Figure 2 shows the values of $\delta_{i,n}$ and $\Delta_{i,n}$ as functions of the residual $\widetilde{y}_{i,n}$ in Equation (33), and for different ratios of latent noise variance $\mathbf{P}_{\setminus i}$ to observed noise scale R_i. Given that $\Delta_{i,n}$ is proportional to scale R_i, the normalized value $\Delta_{i,n}R_i^{-1}$ is plotted.

Both functions are symmetric about $\widetilde{y}_{i,n} = 0$, and nonlinear. The shape of $\delta_{i,n}$ and $\Delta_{i,n}$ depends on $\mathbf{C}_{(i)}$, $\mathbf{P}_{\setminus i}$, and R_i, and generally controls the model's sensitivity to outliers. The function $\delta_{i,n}$ is a sigmoid, or "S"-shaped, as it is approximately linear near $\widetilde{y}_{i,n} = 0$ and approaches negative or positive one as the residual tends to positive or negative infinity. The function $\Delta_{i,n}$ is bell-shaped when the variance is large (a loose model), and is approximately a box function when the variances are close to zero (a tight model).

Taking the limit of these functions as the residual approaches negative or positive infinity yields

$$\lim_{\widetilde{y}_{i,n}\to\pm\infty} \delta_{i,n} = \pm 1\,, \qquad \lim_{\widetilde{y}_{i,n}\to\pm\infty} \Delta_{i,n} = 0\,. \tag{43}$$

Considering Equations (38), (39) and (43), when the difference between the data and the prediction (the residual) tends to infinity, the expected latent state covariance remains as the prior value $\mathbf{P}_{\setminus i}$, while the mean is updated to $\mathbf{m}_{\setminus i} + \mathbf{k}_n$ (when $\widetilde{y}_{i,n} = +\infty$) or $\mathbf{m}_{\setminus i} - \mathbf{k}_n$ (when $\widetilde{y}_{i,n} = -\infty$).

For the Kalman filter, $\delta_{i,n}$ is simply equal to the residual $\widetilde{y}_{i,n}$. Therefore, its state update equation describes a line with a slope altered by the Kalman gain. Since outliers tend to create large residual values, they severely affect the Kalman filter because they pull the expected value of the state far from the prediction. With the Laplace state space model, there is a limit to an observation's influence on the state update, $\mathbf{m}_{\setminus i} \pm \mathbf{k}_n$.

D. Moment matching

The mean and covariance of the approximating Gaussian posterior $q(\mathbf{x}_n; \mathbf{y}_{1:n})$ are set equal to the mean and covariance of the locally exact posterior $p(\mathbf{x}_n|\mathbf{y}_n)$ from Equations (38) and (39):

$$\boldsymbol{\mu}_n = \langle \mathbf{x}_n \rangle\,, \tag{44}$$

$$\mathbf{V}_n = \text{cov}[\mathbf{x}_n, \mathbf{x}_n]\,. \tag{45}$$

This completes the update of the approximate global posterior q. Per Section IV-B, locally exact inference is followed by updating the likelihood term q_i, then proceeding to iterate over the other likelihood terms in the same manner with expectation propagation.

In accordance with the Kalman filtering paradigm, at each time n the filter completes a prediction step followed by an update step. At time $n+1$, the filter begins again at the prediction step with Equations (17) and (18).

Pseudo-code for the complete multivariate data filtering algorithm is presented in Table I.

E. Validation

For the LSSM, the proposed EP-based method provides higher quality approximations to the posterior than variational Bayesian inference (VB)[4]. Figure 3 demonstrates this quality for both a univariate and multivariate observation for a snapshot at time n. Given a univariate observation y_n, the VB approximation centers closely to the mode of the posterior as expected, and severely underestimates the true variance. For a multivariate observation $\mathbf{y}_n$, VB has a large bias and variance when the data has outliers. Our EP algorithm, on the other hand, provides the tightest fit possible around the true posterior, regardless of the dimensionality of observation. For univariate observations, the mean and variance are exactly matched to the posterior's. For multivariate observations, EP provides a close fit to the true mean and variance. When embedded into a Bayesian filter and smoother, the proposed EP-based approximation significantly outperforms existing methods when given higher dimensional data. This is particularly important for sensor fusion, one of the primary applications of the Kalman filter.

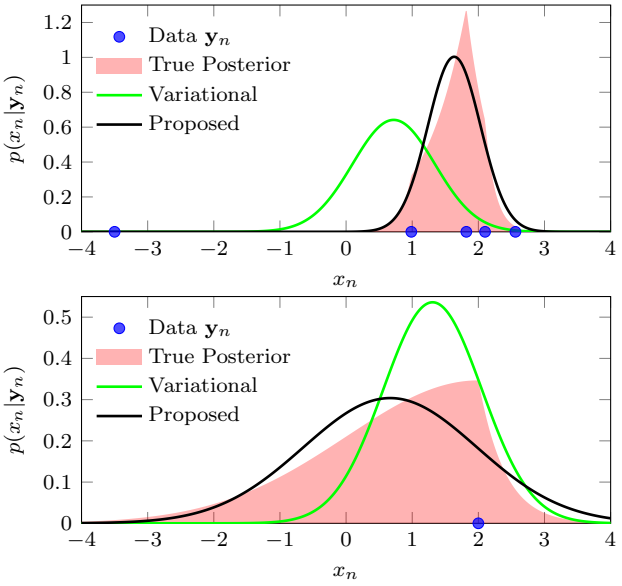


Fig. 3. The true posterior distribution $p(x_n|\mathbf{y}_n)$ and approximating Gaussian posteriors provided by variational inference and the proposed expectation propagation (moment matching) given multivariate data (top) and univariate data (bottom). Note that the quality of the variational approximation is degraded by outliers, centering more closely to the sample mean and under-estimating the variance. Moment matching provides a superior quality Gaussian approximation to the true posterior.

V. Smoothing

Now we turn to estimating the state posterior given all the data, which is completed by a *smoother*. Smoothing refers to inferring the pairwise posterior 46 and marginal posterior 47.

$$p(\mathbf{x}_n, \mathbf{x}_{n+1}|\mathbf{Y}) = \frac{p(\mathbf{x}_n|\mathbf{y}_{1:n})p(\mathbf{x}_{n+1}|\mathbf{x}_n)p(\mathbf{x}_{n+1}|\mathbf{Y})}{p(\mathbf{x}_{n+1}|\mathbf{y}_{1:n})}, \quad (46)$$

$$p(\mathbf{x}_n|\mathbf{Y}) = p(\mathbf{x}_n|\mathbf{y}_{1:n}) \int \frac{p(\mathbf{x}_{n+1}|\mathbf{x}_n)p(\mathbf{x}_{n+1}|\mathbf{Y})}{p(\mathbf{x}_{n+1}|\mathbf{y}_{1:n})} d\mathbf{x}_{n+1}. \quad (47)$$

Smoothing is required for model learning because the parameter update equations depend on the statistics of these posterior distributions.

Smoothing propagates backwards in time, computing an approximation to the marginal posterior given the entire data sequence,

$$p(\mathbf{x}_n|\mathbf{Y}) \approx q(\mathbf{x}_n; \mathbf{Y}) = \mathcal{N}(\mathbf{x}_n|\widehat{\boldsymbol{\mu}}_n, \widehat{\mathbf{V}}_n). \quad (48)$$

All the probabilities involved in smoothing are now Gaussian because the filter assumed a Gaussian density. Moreover, the marginal distribution of a Gaussian conditional and prior is another Gaussian. As a result, the smoothing step is equivalent to the Kalman smoother. Initially $\widehat{\boldsymbol{\mu}}_N = \boldsymbol{\mu}_N$ and $\widehat{\mathbf{V}}_N = \mathbf{V}_N$. Then, from $n = N-1$ back to $n = 1$,

$$\mathbf{J}_n = \mathbf{V}_n \mathbf{A}^\mathsf{T} \mathbf{P}_n^{-1}, \quad (49)$$

$$\widehat{\boldsymbol{\mu}}_n = \boldsymbol{\mu}_n + \mathbf{J}\left(\widehat{\boldsymbol{\mu}}_{n+1} - \mathbf{m}_n\right), \quad (50)$$

$$\widehat{\mathbf{V}}_n = \mathbf{V}_n + \mathbf{J}\left(\widehat{\mathbf{V}}_{n+1} - \mathbf{P}_n\right)\mathbf{J}^\mathsf{T}. \quad (51)$$

Since the filtering involved an approximation, the smoothed pairwise and marginal posteriors are also approximate. However, the smoothing algorithm itself is exact.

VI. Extension to Non-Linear Models

While linear dynamics are able to model much real word data, it is also with great interest to be able to apply Bayesian filters to applications that involve non-linear dynamics, such as tracking the position, velocity, and acceleration of moving targets.

The proposed expectation propagation filter elegantly extends to non-linear models. We call this the extended expectation propagation (EEP) filtering algorithm. In this case, a state is transformed by a non-linear function $h(\mathbf{x}_{n-1})$ then output to the observable space by a non-linear function $g(\mathbf{x}_n)$. Adopting a local linearization approach akin to the extended Kalman filter (EKF) [21], we linearize $h(.)$ around the previous state estimate $\boldsymbol{\mu}_{n-1}$ and $g(.)$ around the predicted mean $\mathbf{m}_{n-1}$.

Considering Section IV, the predicted state, predicted observation, system dynamics matrix, and output matrix are

[4]A description of the variational Bayesian inference algorithm for the LSSM is provided in Appendix A.

TABLE I
IMPLEMENTATION PSEUDO-CODE FOR THE PROPOSED FILTERING ALGORITHM FOR ONE TIME STEP

procedure UPDATE
  Initialize:
  $\boldsymbol{\ell}_i = \mathbf{0}$, $\boldsymbol{\Lambda}_i = \mathbf{I}$, $\forall i \in [1..M]$
  $\boldsymbol{\ell}_0 = \mathbf{P}_{n-1}^{-1}\mathbf{m}_{n-1}$, $\boldsymbol{\Lambda}_0 = \mathbf{P}_{n-1}^{-1}$
  $\widehat{\boldsymbol{\ell}} = \sum_{i=0}^{M} \boldsymbol{\ell}_i$, $\widehat{\boldsymbol{\Lambda}} = \sum_{i=0}^{M} \boldsymbol{\Lambda}_i$
  while not converged **do**
    for $i = 1 \ldots M$ **do**
      $\boldsymbol{\ell}_{\setminus i} = \widehat{\boldsymbol{\ell}} - \boldsymbol{\ell}_i$
      $\boldsymbol{\Lambda}_{\setminus i} = \widehat{\boldsymbol{\Lambda}} - \boldsymbol{\Lambda}_i$
      $\mathbf{m}_{\setminus i} = \boldsymbol{\Lambda}_{\setminus i}^{-1}\boldsymbol{\ell}_{\setminus i}$
      $\mathbf{P}_{\setminus i} = \boldsymbol{\Lambda}_{\setminus i}^{-1}$
      $\widetilde{y}_{i,n} \leftarrow y_{i,n} - \mathbf{C}_{(i)}\mathbf{m}_{\setminus i}$
      $S_{i,n} \leftarrow \mathbf{C}_{(i)}\mathbf{P}_{\setminus i}\mathbf{C}_{(i)}^{\mathsf{T}}$
      $\Phi_{i,n}^{(\pm)} \leftarrow \operatorname{erfcx}\left(\frac{\sqrt{S_{i,n}}}{\sqrt{2R_i^2}} \pm \frac{\widetilde{y}_{i,n}}{\sqrt{2S_{i,n}}}\right)$
      $\mathbf{k}_n \leftarrow \mathbf{P}_{\setminus i}\mathbf{C}_{(i)}^{\mathsf{T}} R_i^{-1}$
      $\delta_{i,n} \leftarrow \frac{\Phi_{i,n}^{(-)} - \Phi_{i,n}^{(+)}}{\Phi_{i,n}^{(-)} + \Phi_{i,n}^{(+)}}$
      $\Delta_{i,n} \leftarrow \frac{1}{\Phi_{i,n}^{(-)} + \Phi_{i,n}^{(+)}}\left(\frac{4\Phi_{i,n}^{(-)}\Phi_{i,n}^{(+)}}{\Phi_{i,n}^{(-)} + \Phi_{i,n}^{(+)}} - \sqrt{\frac{8R_i^2}{\pi S_{i,n}}}\right)$
      $\boldsymbol{\mu}_n \leftarrow \mathbf{m}_{\setminus i} + \mathbf{k}_n\delta_{i,n}$
      $\mathbf{V}_n \leftarrow \mathbf{P}_{\setminus i} + \mathbf{k}_n\mathbf{k}_n^{\mathsf{T}}\Delta_{i,n}$
      $\widehat{\boldsymbol{\ell}} = \mathbf{V}_n^{-1}\boldsymbol{\mu}_n$
      $\widehat{\boldsymbol{\Lambda}} = \mathbf{V}_n^{-1}$
      $\boldsymbol{\ell}_i = \widehat{\boldsymbol{\ell}} - \boldsymbol{\ell}_{\setminus i}$
      $\boldsymbol{\Lambda}_i = \widehat{\boldsymbol{\Lambda}} - \boldsymbol{\Lambda}_{\setminus i}$
    end for
  end while
end procedure
procedure PREDICT
  $\mathbf{m}_n \leftarrow \mathbf{A}\boldsymbol{\mu}_n$
  $\mathbf{P}_n \leftarrow \mathbf{A}\mathbf{V}_n\mathbf{A}^{\mathsf{T}} + \mathbf{Q}$
end procedure

approximated by, respectively,

$$\mathbf{m}_{n-1} = h(\boldsymbol{\mu}_{n-1}), \qquad \widehat{y}_{i,n} = g_i(\mathbf{m}_{n-1}), \tag{52}$$

$$\mathbf{A}_n = \left.\frac{dh}{d\mathbf{x}_{n-1}}\right|_{\boldsymbol{\mu}_{n-1}}, \qquad \mathbf{C}_{(i),n} = \left.\frac{dg_i}{d\mathbf{x}_n}\right|_{\mathbf{m}_{n-1}}, \tag{53}$$

where $\mathbf{A}_n$ and $\mathbf{C}_{(i),n}$ are Jacobian matrices. While this is the simplest extension, one could instead employ an unscented transform for non-linear estimation [31]. In contrast, variational inference-based approaches are violated by this kind of local linearization and require the monitoring of convergence or iterative optimization of the variational lower bound [32].

VII. LEARNING

Thus far we have considered the approximate inference problem for the Laplace state space model (LSSM), assuming that the model parameters $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{Q}, \mathbf{C}, \mathbf{R}, \mathbf{m}_0, \mathbf{P}_0\}$ are known. Next, we consider the automatic learning of these parameters using maximum a posteriori probability (MAP) estimation. Since the model has latent variables, learning can be addressed using the expectation maximization (EM) algorithm [33] [34]. The expectation (E) step consists of filtering and smoothing, using the current estimates of the parameters $\hat{\boldsymbol{\theta}}$. The maximization (M) step maximizes the expected log-likelihood function under the latent variable's distribution with respect to the model parameters, plus the log-prior probabilities of the parameters.

$$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \left\{ \langle \ln p(\mathbf{Y}, \mathbf{X}|\boldsymbol{\theta}) \rangle_{q(\mathbf{X}|\boldsymbol{\theta}^{\text{old}})} + \ln p(\boldsymbol{\theta}) \right\} \tag{54}$$

Prior probabilities regulate maximum likelihood estimates.

Consider the complete data log-likelihood given by

$$\ln p(\mathbf{Y}, \mathbf{X}|\boldsymbol{\theta}) = \ln p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) + \ln p(\mathbf{X}|\boldsymbol{\theta}). \tag{55}$$

The log probability of the latent state sequence is

$$\begin{aligned} \ln p(\mathbf{X}|\mathbf{A}, \mathbf{Q}) = &-\frac{N-1}{2}\ln\det(2\pi\mathbf{Q}) \\ &- \frac{1}{2}\sum_{n=2}^{N}(\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})^{\mathsf{T}}\mathbf{Q}^{-1}(\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1}) \\ &- \frac{1}{2}\ln\det(2\pi\mathbf{P}_0) - \frac{1}{2}(\mathbf{x}_1 - \mathbf{m}_0)^{\mathsf{T}}\mathbf{P}_0^{-1}(\mathbf{x}_1 - \mathbf{m}_0), \end{aligned} \tag{56}$$

and the log likelihood of the data sequence is

$$\ln p(\mathbf{Y}|\mathbf{X}, \mathbf{C}, \mathbf{R}) = \sum_{n=1}^{N}\sum_{i=1}^{M}\left\{ -\ln 2R_i - \frac{\left|y_{i,n} - \mathbf{C}_{(i)}\mathbf{x}_n\right|}{R_i} \right\}. \tag{57}$$

We now take the expectation of the complete-data log likelihood with respect to the Gaussian approximate posterior distribution $q(\mathbf{X}; \mathbf{Y})$ from the E step, which defines the function

$$\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{\text{old}}) = \langle \ln p(\mathbf{Y}, \mathbf{X}|\boldsymbol{\theta}) \rangle_{q(\mathbf{X}|\boldsymbol{\theta}^{\text{old}})}. \tag{58}$$

In the M step, this function and the log priors are maximized with respect to the components of $\boldsymbol{\theta}$.

We define zero-mean Gaussian priors over the columns of $\mathbf{C}$:

$$p(\mathbf{C}|\mathbf{R}) = \textstyle\prod_{i=1}^{M}\mathcal{N}(\mathbf{C}_{(i)}|\mathbf{0}, \operatorname{diag}(\boldsymbol{\tau})^{-1}R_i), \tag{59}$$

where $D \times 1$ vector $\boldsymbol{\tau}$ are hyperparameters. The dependence of the precision of $\mathbf{C}$ on the noise variance (scale) $\mathbf{R}$ is motivated by conjugacy and links the scale (amplitude) of the signal to the noise [24]. The use of maximum likelihood estimation of the hyperparameters has the effect of penalizing complex models and gives rise to a sparse model parameterization.

For the noise variances, we define inverse Gamma priors over the Laplace scales.

$$p(\mathbf{R}) = \textstyle\prod_{i=1}^{M}\text{Gam}(R_i^{-1}|a_0, b_0) \tag{60}$$

Defining these priors over the noise parameters helps to prevent negative estimates when there is a small number of observations. Also, the estimates from the EM algorithm can be guided through the choice of these hyperparameters, as they indicate prior beliefs about the noise. For example, setting a_0 and b_0 such that the expected value of the distribution is very large promotes smaller variances. This is useful to fit the latent space tightly to a dynamical model, and thus enforce more interpretable results. Alternatively, an uninformative prior has hyperparameters that are approximately zero.

We choose to make maximum likelihood estimates of the latent state parameters to simplify their update equations and retain focus on the estimation of the Laplace PDF parameters. Placing similar priors over the state parameters is straightforward and has been well-studied [24] [35].

A. Laplace output probability parameters

Consider first the parameters $\mathbf{C}$ and $\mathbf{R}$ of the Laplace output probability function. Taking the expectation of the log likelihood in Equation (57) with respect to $q(\mathbf{X})$ and absorbing terms that do not depend on $\mathbf{C}$ or $\mathbf{R}$ into a constant yields

$$\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{\text{old}}) \stackrel{c}{=} \sum_{i=1}^{M}\sum_{n=1}^{N}\left\{-\ln 2R_i - \frac{2\rho_{i,n}S_{i,n}}{R_i} - \frac{\widetilde{y}_{i,n}\text{erf}\left(l_{i,n}\right)}{R_i}\right\} \tag{61}$$

where we have defined

$$\widetilde{y}_{i,n} = y_{i,n} - \mathbf{C}_{(i)}\widehat{\boldsymbol{\mu}}_n\,, \tag{62}$$
$$S_{i,n} = \mathbf{C}_{(i)}\widehat{\mathbf{V}}_n\mathbf{C}_{(i)}^{\mathsf{T}}\,, \tag{63}$$
$$l_{i,n} = \frac{\widetilde{y}_{i,n}}{\sqrt{2S_{i,n}}}\,, \tag{64}$$
$$\rho_{i,n} = \frac{\exp(-l_{i,n}^2)}{\sqrt{2\pi S_{i,n}}} = \mathcal{N}(\widetilde{y}_{i,n}|0, S_{i,n})\,. \tag{65}$$

Maximization of this objective function and the log prior with respect to $\mathbf{C}$ and $\mathbf{R}$ is addressed with gradient root-finding [36].

1) Output matrix: Given that $\mathcal{Q}$ involves a sum over M, each row of $\mathbf{C}$ is independent of the other. Thus, we describe the process for estimating a single row $\mathbf{C}_{(i)}$, which generalizes to all i. Since the function is not differentiable at all points, we describe the objective subgradients as prescribed in [36]. When $\sum_j|\mathbf{C}_{(i,j)}| \neq 0$, the $1 \times D$ subgradient is

$$\boldsymbol{\nabla}_{\mathcal{Q}}(\mathbf{C}_{(i)}) = -\frac{1}{R_i}\sum_{n=1}^{N}\left\{2\rho_{i,n}\mathbf{C}_{(i)}\widehat{\mathbf{V}}_n - \text{erf}\left(l_{i,n}\right)\widehat{\boldsymbol{\mu}}_n^{\mathsf{T}}\right\}. \tag{66}$$

When $\sum_j|\mathbf{C}_{(i,j)}| = 0$, the subgradient is undefined. However, it should be a rare case that every element in $\mathbf{C}_{(i)}$ is zero, as it would mean that the observation $y_{i,n}$ has no discernible underlying dynamics. Still, we can find the value of the gradient as all of the elements in $\mathbf{C}_{(i)}$ approach zero,

$$\lim_{\mathbf{C}_{(i)}\to\mathbf{0}}\left(\boldsymbol{\nabla}_{\mathcal{Q}}(\mathbf{C}_{(i)})\right) = -\frac{1}{R_i}\sum_{n=1}^{N}\text{sign}(y_{i,n})\widehat{\boldsymbol{\mu}}_n^{\mathsf{T}}\,. \tag{67}$$

Thus, the gradient may be set to any value between zero and the one defined in Equation (67). Alternatively, the maximum may indeed be located at $\mathbf{C}_{(i)} = \mathbf{0}$. This can be verified by searching for larger values of $\mathcal{Q}$ in the vicinity of $\mathbf{C}_{(i)} = \mathbf{0}$.

Because an analytic solution to $\boldsymbol{\nabla}_{\mathcal{Q}}(\mathbf{C}_{(i)}) = 0$ is not available, we use an iterative root-finding algorithm. Specifically, as we are dealing with an unconstrained convex optimization problem and can derive the first and second derivatives of the objective function, we elect to use Newton's method [36].

The $D \times D$ Hessian matrix is given by

$$\mathbf{H}_{\mathcal{Q}}(\mathbf{C}_{(i)}) = \frac{2}{R_i}\sum_{n=1}^{N}\left\{\frac{\rho_{i,n}}{S_{i,n}}\widehat{\mathbf{V}}_n\mathbf{C}_{(i)}^{\mathsf{T}}\mathbf{C}_{(i)}\widehat{\mathbf{V}}_n - \rho_{i,n}(\boldsymbol{\xi}_n^{\mathsf{T}}\boldsymbol{\xi}_n + \widehat{\mathbf{V}}_n)\right\}, \tag{68}$$

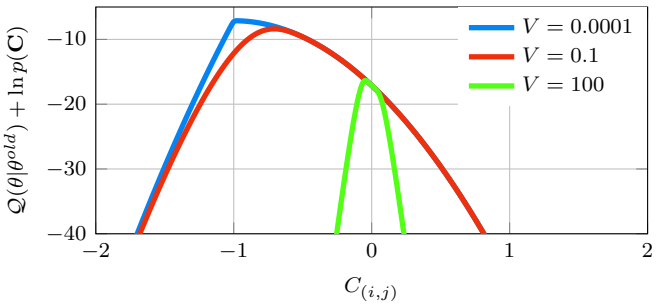


Fig. 4. Objective to maximize with respect to $C_{(i,j)}$ for different values of $V_{(j,j)}$. The values of the data and expected state were $y = 1$ and $\mu = -1$, while the scale was set to $R_i = .05$. When the expected state variance is small, the maximum of the objective is at $C_{(i,j)} = -1$, which minimizes the residual $\widetilde{y}_i = y_i - C_{(i,j)}\mu_j$. For less certain expected state values, where $V_{(j,j)}$ is large, the estimate of $C_{(i,j)}$ is closer to zero.

where we have defined the $1 \times D$ vector

$$\boldsymbol{\xi}_n = \widehat{\boldsymbol{\mu}}_n^{\mathsf{T}} + \frac{\widetilde{y}_{i,n}}{S_{i,n}}\mathbf{C}_{(i)}\widehat{\mathbf{V}}_n\,. \tag{69}$$

For a MAP estimate, the gradient and Hessian with respect to the log prior probability are

$$\boldsymbol{\nabla}_p(\mathbf{C}_{(i)}) = -\mathbf{C}_{(i)}\text{diag}(\boldsymbol{\tau})R_i^{-1}\,, \tag{70}$$
$$\mathbf{H}_p(\mathbf{C}_{(i)}) = -\text{diag}(\boldsymbol{\tau})R_i^{-1}\,. \tag{71}$$

Hyperparameter $\boldsymbol{\tau}$ scales the variance of the normal distribution. Updating the hyperparameter can create sparse solutions by guiding the estimate of $\mathbf{C}$ towards zero.

Finally, the Newton method update is

$$\boldsymbol{\nabla}(\mathbf{C}_{(i)}) = \boldsymbol{\nabla}_{\mathcal{Q}}(\mathbf{C}_{(i)}) + \boldsymbol{\nabla}_p(\mathbf{C}_{(i)})\,, \tag{72}$$
$$\mathbf{H}(\mathbf{C}_{(i)}) = \mathbf{H}_{\mathcal{Q}}(\mathbf{C}_{(i)}) + \mathbf{H}_p(\mathbf{C}_{(i)})\,, \tag{73}$$
$$\mathbf{C}_{(i)} = \mathbf{C}_{(i)} - \boldsymbol{\nabla}(\mathbf{C}_{(i)})\mathbf{H}(\mathbf{C}_{(i)})^{-1}\,. \tag{74}$$

Convergence of the algorithm can be monitored with the *Newton decrement*

$$\epsilon_i = \left(\boldsymbol{\nabla}(\mathbf{C}_{(i)})\mathbf{H}(\mathbf{C}_{(i)})^{-1}\boldsymbol{\nabla}(\mathbf{C}_{(i)})^{\mathsf{T}}\right)^{1/2}. \tag{75}$$

The algorithm is terminated after ϵ_i falls below a pre-defined threshold. Figure 4 shows the objective function with respect to $C_{(i,j)}$ for a variety of (scalar) state covariance estimates V.

Maximizing the log prior probability with respect to $\boldsymbol{\tau}$, we get the analytic estimate

$$\tau_j = \frac{M}{\sum_{i=1}^{M}C_{(i,j)}^2R_i^{-1}}\,. \tag{76}$$

As $C_{(i,j)}$ or R_i^{-1} approach zero, τ_j extends to infinity. In turn, the prior over $C_{(i,j)}$ tends towards a delta function centered at zero. Considering the update to $\mathbf{C}$, large values of τ_j drive the values of column vector $\mathbf{C}_{(j)}$ towards the prior mean of zero. To illustrate this, Figure 5 shows the objective function with respect to $C_{(i,j)}$ for a variety of τ_j values.

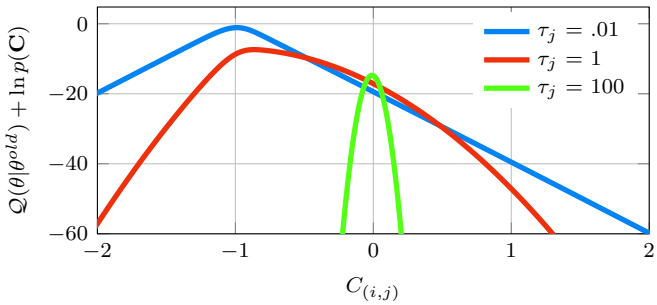


Fig. 5. Objective to maximize with respect to $C_{(i,j)}$ for three different values of τ_j. The maximum of the objective function tends to center on $C_{(i,j)} = 0$ as τ_j goes to infinity.

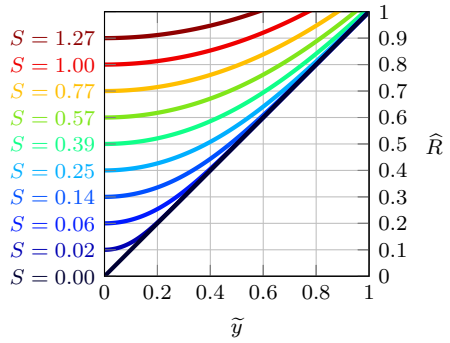


Fig. 6. Estimate of R for different residual values $\widetilde{y}$ and transformed latent variance S. For this graph $a_0 = b_0 = 0$ and $\tau = \infty$, which nulls their influence. As the function for $\widehat{R}$ is even (centered on $\widetilde{y} = 0$), only one half (for positive $\widetilde{y}$) is shown for clarity.

2) Output noise variance (Laplace scale): The Laplace scale parameter R_i that maximizes $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{\text{old}})$ and the log prior probability is given by

$$R_i = \frac{\widehat{b}_i + b_0/2}{\widehat{a}_i + a_0/2}\,, \tag{77}$$

where

$$\widehat{b}_i = \sum_{n=1}^{N} \{2\rho_{i,n} S_{i,n} + \widetilde{y}_{i,n} \text{erf}\,(l_{i,n})\} + \frac{1}{2}\sum_{j=1}^{D} \tau_j C_{(i,j)}^2\,, \tag{78}$$

$$\widehat{a}_i = N + D/2\,. \tag{79}$$

The newly estimated value of $\mathbf{C}$ is used to compute $\widetilde{y}_{i,n}$, $l_{i,n}$, $S_{i,n}$, and $\rho_{i,n}$, per Equations (62) to (65).

Figure 6 shows the estimated value of R_i as a function of the residual $\widetilde{y}_{i,n}$ and for a range of output variances $S_{i,n}$. This figure illustrates how the estimation of R_i reflects the model's robustness to outliers and ability to avoid over-fitting data. Specifically, the lower bound of the scale estimate is linear with respect to $\widetilde{y}_{i,n}$. In contrast, Gaussian noise covariance estimates have a quadratic lower bound and thus promote values closer to zero than Laplace noise estimates. As a result, GSSMs have a tendency to over-fit the data. Data over-fitting is automatically avoided by assuming Laplace noise and estimating its scale with the proposed method.

B. State parameters

Now consider the parameters $\mathbf{A}$ and $\mathbf{Q}$. Taking the expectation of Equation (56) with respect to $q(\mathbf{X})$ and absorbing terms that do not depend on $\mathbf{A}$ or $\mathbf{Q}$ into a constant, we get

$$\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{\text{old}}) \stackrel{c}{=} -\frac{N-1}{2}\ln\det(\mathbf{Q}) - \frac{1}{2}\text{Tr}\Bigg(\mathbf{Q}^{-1}\sum_{n=2}^{N}\Big\{\boldsymbol{\mathcal{X}}_n + \mathbf{A}\boldsymbol{\mathcal{X}}_{n-1}\mathbf{A}^{\mathsf{T}} - \Big(\mathbf{A}\boldsymbol{\mathcal{X}}_{n-1,n} + \boldsymbol{\mathcal{X}}_{n-1,n}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\Big)\Big\}\Bigg)\,, \tag{80}$$

where we have defined the following latent state statistics,

$$\boldsymbol{\mathcal{X}}_n \triangleq \langle \mathbf{x}_n \mathbf{x}_n^{\mathsf{T}} \rangle_{q(\mathbf{X})} = \widehat{\boldsymbol{\mu}}_n \widehat{\boldsymbol{\mu}}_n^{\mathsf{T}} + \widehat{\mathbf{V}}_n\,, \tag{81}$$

$$\boldsymbol{\mathcal{X}}_{n-1,n} \triangleq \langle \mathbf{x}_{n-1} \mathbf{x}_n^{\mathsf{T}} \rangle_{q(\mathbf{X})} = \widehat{\boldsymbol{\mu}}_{n-1} \widehat{\boldsymbol{\mu}}_n^{\mathsf{T}} + \mathbf{J}_{n-1}\widehat{\mathbf{V}}_n\,. \tag{82}$$

Since the transition probability is Gaussian and the approximating marginal posterior is Gaussian, the estimation of these parameters is equivalent to the ones used in the EM algorithm for linear GSSMs, as in [28], [34].

1) System dynamics matrix: The value of $\mathbf{A}$ that maximizes $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{\text{old}})$ is

$$\mathbf{A} = \left(\sum_{n=2}^{N} \mathbf{y}_n \widehat{\boldsymbol{\mu}}_n^{\mathsf{T}}\right)\left(\sum_{n=2}^{N} \boldsymbol{\mathcal{X}}_{n-1}\right)^{-1}. \tag{83}$$

2) Latent noise covariance matrix: The value of $\mathbf{Q}$ that maximizes $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{\text{old}})$ is

$$\mathbf{Q} = \frac{1}{N-1}\sum_{n=2}^{N}\Big\{\boldsymbol{\mathcal{X}}_n - \mathbf{A}\boldsymbol{\mathcal{X}}_{n-1,n}^{\mathsf{T}}\Big\}\,, \tag{84}$$

where $\mathbf{A}$ is newly estimated from Equation (83).

3) Initial state mean and covariance: Maximizing the expected log probability in Equation (56) with respect to the initial mean and covariance we get the estimates $\mathbf{m}_0 = \widehat{\boldsymbol{\mu}}_1$ and $\mathbf{P}_0 = \widehat{\mathbf{V}}_1$.

VIII. Experiments and Results

In order to evaluate the proposed method, we tested its performance along with existing approaches by a series of inference and learning experiments. For the inference experiments, the proposed expectation propagation-based filter (EP) was compared with a Kalman filter (KF), a bootstrap particle filter (PF) [37] with systematic resampling, and a variational Bayesian (VB) inference-based filter that uses an auxiliary latent variable to model the Laplace distribution as a scale mixture of Gaussians (see Appendix A for details), PF used 200 samples and was approximately 200 times slower than the other methods. While more samples would improve estimation quality, it would further increase computation time. Performance was evaluated according to the root mean square error (RMSE), given by

$$RMSE_n = \frac{1}{J}\sum_{j=1}^{J}\sqrt{\sum_{i=1}^{D}\left(\langle x_{i,n}^{(j)}\rangle - x_{i,n}^{(j)}\right)^2}\,, \tag{85}$$

where $\mathbf{x}_n^{(j)}$ was the true state and $\langle \mathbf{x}_n^{(j)} \rangle$ was the state estimate of the jth Monte Carlo run. Inference experiments consisted of $J = 1000$ Monte Carlo runs.

A. Inference results for linear dynamics

Data was generated according to first-order linear state space dynamics. The true latent state $\mathbf{x}_n$ was two-dimensional ($D = 2$) and oscillated according to system dynamics matrix

$$\mathbf{A} = \begin{bmatrix} \cos(2\pi fT) & -\sin(2\pi fT) \\ \sin(2\pi fT) & \cos(2\pi fT) \end{bmatrix}, \tag{86}$$

where the frequency was randomly set in the range $f \in [0, \frac{Fs}{10}]$ Hz, the sampling rate was $Fs = 44.1$ kHz, and the sampling period was $T = 1/Fs$ seconds. Each row of the output matrix was sampled from a zero-mean, unit-variance Gaussian distribution, $\mathbf{C}_{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and normalized, $|\sum_j C_{(i,j)}| = 1$, for $i \in [1..M]$. The latent noise covariance matrix was diagonal $\mathbf{Q} = \mathbf{I}\sigma^2$ where $\sigma^2 = 10^{-4}$.

The first experiment was designed to test the efficacy of each method in estimating the hidden state sequence for a LSSM, given all the correct model parameters and a one-dimensional observation. The initial state prior was set to zero-mean with a relatively large variance of $\mathbf{P}_0 = \mathbf{I}$ to test each filter's response. Figure 7a shows the RMSE over time for each method. EP quickly settled to the lowest RMSE.

Next, an experiment was designed to test how each filter reacts to an abrupt change in noise variance and mean for an extended period of time. The observed signal was corrupted by Gaussian noise with variance that was abruptly increased from .01 to 1 at time $n = 60$, which lasted for 60 samples before reducing back to .01. During this time period, the mean of the Gaussian output probability was set to zero instead of $\mathbf{C}\mathbf{x}_n$. Each filter was given the correct system dynamics, with observed noise variance equal to .1, *i.e.* the average of the two noise levels. Resulting RMSE plots for each filter are shown in Figure 7b. The RSME for KF was smallest until $n = 60$, as it is the optimal estimator for Gaussian models, but grew rapidly at $n = 60$ because as KF did not adapt to changes in noise variance. The three filters that assumed Laplace noise were only lightly affected by the period of increased noise. VB was less affected than EP because the auxiliary variable of the Gaussian scale mixture distribution acted as a time-varying weight on $\mathbf{R}$. PF's lagged response attributed to its low RMSE during the increased noise, but was larger than PF and VB afterwards. The RMSE for EP was smaller than VB and PF except during the interval $60 \leq n \leq 120$. More generally, the RMSE for the Laplace-based filters increased linearly while the RMSE for KF increased along an inverted exponential curve. This behavior follows from Section IV-C on the proposed filter's update equations.

The third experiment used observations sampled from the same model as the first experiment, except the filters were provided with an incorrect system matrix $\mathbf{A} = \mathbf{I}$. Filtering with incorrect model parameters is common in real-world applications where the true underlying dynamics are not available and must be inferred. State covariance $\mathbf{Q}$ was set to larger values than the previous experiment to reflect the incorrect dynamics matrix. Figure 8 shows the results for this experiment, which was completed for $M = 1$ and again for $M = 4$ data dimensions.

Compared to the other methods, EP consistently had the lowest RMSE. Drastic improvements over existing methods

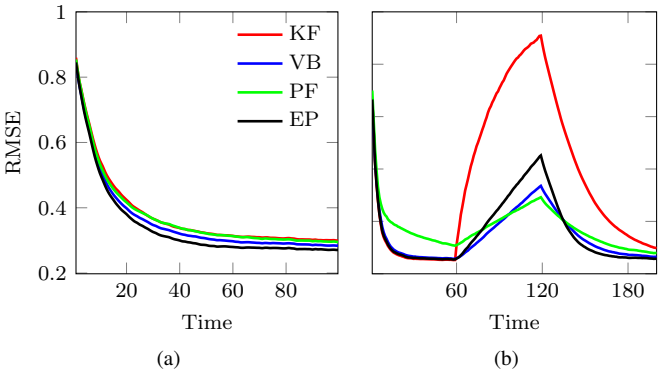


Fig. 7. Inference results for a signal with: (a) Laplace-distributed observation noise, given true model parameters and vague initial conditions; (b) Gaussian noise with a variance of 1 for $60 \leq n \leq 120$, and .01 otherwise.

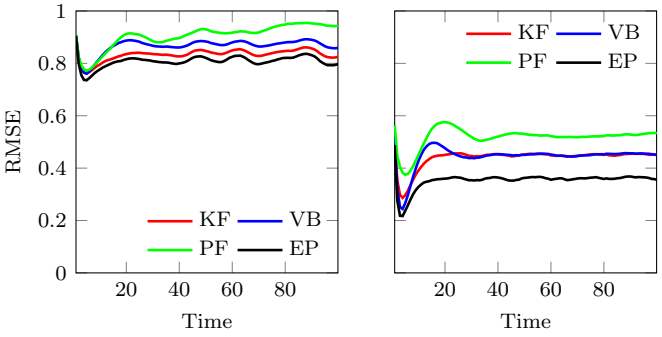
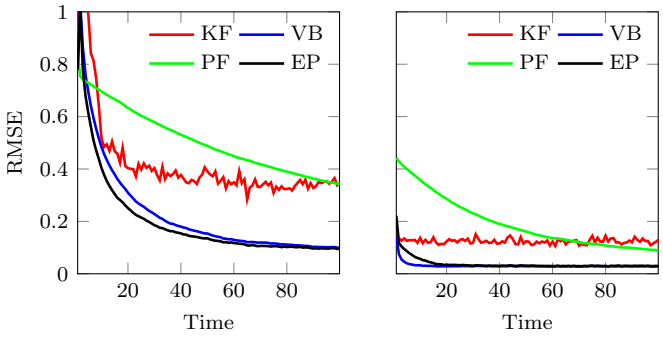


Fig. 8. Inference results for test with incorrect system dynamics matrix $\mathbf{A}$.

are clear in the 4-dimensional data results. EP provided a superior approximation to the marginal posterior and remained a tight fit even in high dimensions. In contrast, the variational approach performed poorly in higher dimensions, seeking some mode that was far from the mean of the posterior. Being able to approximate the posterior well given multidimensional data is crucial for sensor fusion, a primary application of Bayesian filters.

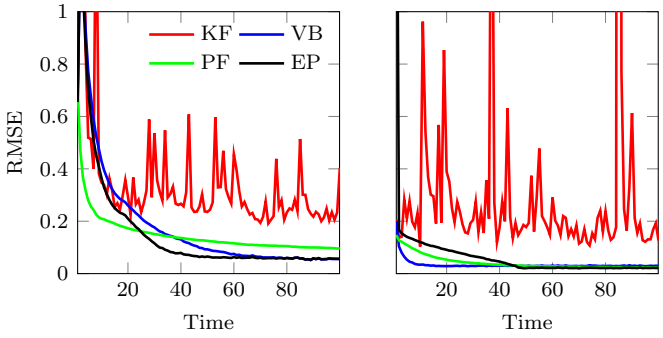
The fourth experiment involved a signal with severe outliers. The outliers were sampled randomly from a Gaussian distribution with a variance of 1, spaced in time at an average rate of 1 in 10 samples. Otherwise, the signal had light Gaussian noise, with a variance of 10^{-3}. Figure 9 shows the results for a test involving a one-dimensional signal and a 4-dimensional signal. Both EP and VB filtered the outliers successfully and outperformed PF and KF. As expected, KF was severely affected by the data outliers.

We tested how each filter responded to data generated from the same dynamical system as in Experiment 4, but instead corrupted by noise drawn from the Cauchy distribution [13]. Using a scale of .01 for the Cauchy distribution created data that had moderate noise and occasional extreme outliers. In this experiment, PF assumed Cauchy noise with the same parameters as generative model. Figure 10 shows that even though EP and VB assumed Laplace noise, they still performed excellently. Moreover, EP settled to the lowest RMSE after



(a) $M = 1$. (b) $M = 8$.

Fig. 9. Inference results for data corrupted by Gaussian outliers occurring at an average rate of 1 in 10 samples.



(a) $M = 1$. (b) $M = 4$.

Fig. 10. Inference results for Cauchy noise-corrupted data. For this experiment, the particle filter (PF) assumed Cauchy noise.

about 50 samples. KF's performance was poor due to the significant outliers in the data.

B. Inference results for non-linear dynamics

A non-linear Bayesian filtering experiment was conducted to test the ability of each method to track abrupt changes in the data that correspond to adaptations in the latent state's instantaneous variables. A test signal was synthesized by adding Gaussian-noise to a sinusoid whose frequency started at 200 Hz and doubled every 2000 samples, concluding at 1600 Hz. Figure 11 shows that the proposed extended EP (EEP) filter quickly adapted to the discrete change in frequency, even for the largest difference of 800 Hz. In contrast, VB responded slower and did not resolve the fourth frequency step. PF was able to track the first two frequency steps because the state transition probability's large variance enabled a sufficient sampling of the latent space. Still, PF did not track the larger frequency changes of the third and fourth steps. EKF's frequency and amplitude estimates undesirably modulated at the sample rate to account for the difference in frequency between the second and third step. Enhancing the non-linear filtering algorithms, for example by implementing changepoint detection to reset the latent noise covariance over time, could help in this scenario. Even without such modifications, the proposed EEP was successful.

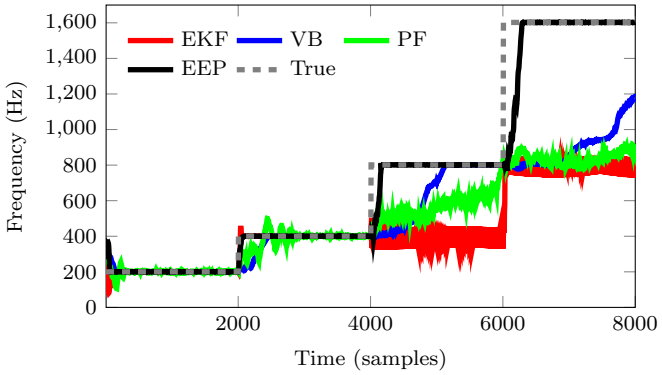


Fig. 11. Instantaneous frequency estimation of a noisy sinusoid's frequency as it doubles every 2000 samples. The proposed EEP filtering algorithm quickly adapts to the abruptly altered frequency while the other filters do not.

C. Model learning experiments

To evaluate the inference and learning EM algorithm for the Laplace state space model (LSSM), we first applied it to the task of learning an oscillatory sequence with a period of increased noise levels. For comparison, we also evaluated the Gaussian state space model (GSSM) EM algorithm [34]. The true latent state was two-dimensional and oscillated according to a system rotation matrix $\mathbf{A}$ as in Equation (86). The true latent noise covariance matrix $\mathbf{Q}$ was diagonal with the diagonal elements equal to $1e{-}3$. Rows of the output matrix were independently sampled from a zero-mean unit-variance Gaussian distribution and normalized: $\mathbf{C}_{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $|\sum_j C_{(i,j)}| = 1$, for $i \in [1..M]$. Output $\mathbf{C}\mathbf{x}_n$ was corrupted by Gaussian noise with variance of 1 for $30 \leq n \leq 50$ and $1e{-}2$ otherwise. Both LSSM and GSSM models to be learned were initialized with $D = 10$ latent dimensions and an identity dynamics matrix $\mathbf{A} = \mathbf{I}$.

Figure 12 shows that LSSM was robust to the period of drastically increased noise and successfully learned the latent space dynamics. Moreover, it correctly pruned the unnecessary dimensions of the output matrix $\mathbf{C}$, leaving only one significant value as in the true generative model. GSSM over-fit the data through the increased noise and provided a less accurate estimation of the latent dynamics. This quality can be seen from the samples drawn from each of the learned models, as displayed in Figure 13. The proposed LSSM EM algorithm learned the latent dynamics matrix $\mathbf{A}$ and provided an accurate estimate of the small values in covariance matrix $\mathbf{Q}$. GSSM estimated large values for the latent noise covariance matrix, which resulted in very noisy latent samples and indiscernible system dynamics.

For a second experiment, a sequence was generated from the same latent oscillatory dynamics model (with random oscillation frequency) except with data randomly set to zero at a rate of 1 in every 10 samples. Such data is not representative of any particular distribution, but rather a challenging case of repeated sensor failures. The data sequence and the learned filtered sequences are shown in Figure 14. As before, LSSM learned parameters that match the oscillatory behavior and avoided over-fiting the data. GSSM over-fit the sequence as it tried to capture the zero-valued data.

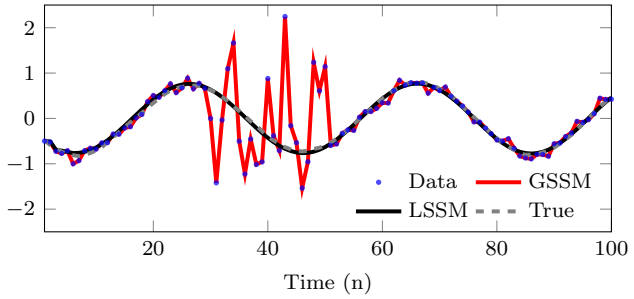
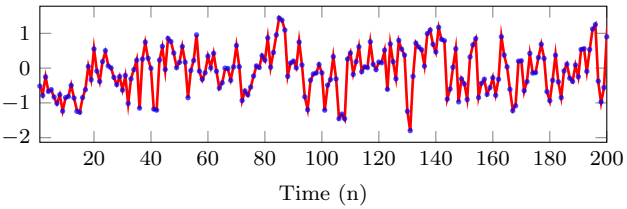
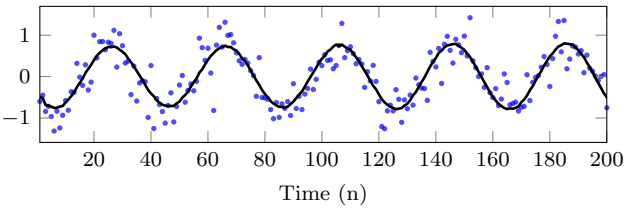


Fig. 12. Learning a state space model from data. The data had moderate Gaussian noise except for a drastic increase for $30 \leq n \leq 50$. The LSSM learned a better latent representation and did not over-fit the noise like the GSSM model.



(a) GSSM.



(b) LSSM.

Fig. 13. Sequences generated from the learned models before adding noise (solid lines) and after adding noise (blue dots). The LSSM learned a better representation of the latent space dynamics because it was robust to outlier data and avoided model over-fitting.

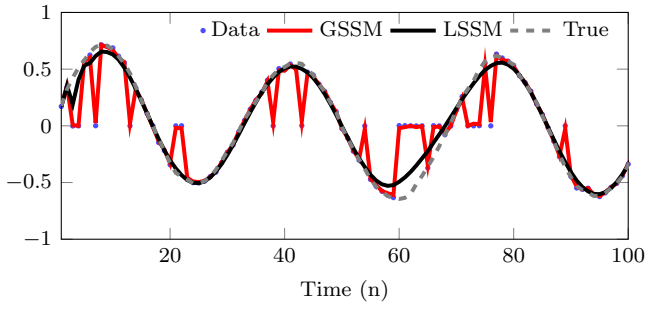


Fig. 14. Learning a state space model from data. The data had low noise except for zero values occurring randomly at a rate of 1 in every 10 samples. The LSSM successfully avoided the zero values and learned a better latent representation than the GSSM.

IX. Conclusion

In this paper, we introduced an approximate inference and learning algorithm for state space models with Laplace-distributed noise that is robust to heavy-tailed and outlier-ridden time series data. Locally exact inference of the Laplace state space model was developed and embedded in an expectation propagation algorithm for multivariate data. After developing the filtering and smoothing algorithms and demonstrating their advantages over existing approximate inference methods like variational Bayes and particle filtering, we proposed an expectation maximization (EM) algorithm for the automatic learning of model parameters. Through a series of experiments, the robustness of the Laplace state space model to outliers and other non-Gaussian noises was validated against existing Gaussian linear dynamical system approaches. The update equations that emerge naturally from the model offer an automatic avoidance to over-fitting data that is generally desirable for time series inference and learning applications. Given that sensor noise in a variety of real-world data is well-represented by the Laplace distribution, and that the Laplace noise model improves state space model learning, there is much potential to apply the methods presented in this paper to sophisticated time series estimation and unsupervised learning problems.

Appendix A
Variational Inference of the Gaussian Scale Mixture Laplace Distribution

This section provides the variational inference routine for the Laplace distribution as represented by a Gaussian scale mixture (GSM). A GSM distribution involves an auxiliary latent variable $z_n \in [0, \infty]$.

$$p(\mathbf{y}_n|\mathbf{x}_n, z_n) = \prod_{i=1}^{M} \mathcal{N}(y_{i,n}|\mathbf{C}_{(i)}\mathbf{x}_n, 2R_i^2 z_n) \tag{87}$$

$$p(\mathbf{x}_n) = \mathcal{N}(\mathbf{x}_n|\mathbf{m}_{n-1}, \mathbf{P}_{n-1}) \tag{88}$$

$$p(z_n) = \exp(-z_n) \tag{89}$$

In marginalizing out z_n, we get the Laplace distribution

$$p(\mathbf{y}_n|\mathbf{x}_n) = \int_0^\infty p(\mathbf{y}_n|\mathbf{x}_n, z_n)p(z_n)dz_n \tag{90}$$

$$= \prod_{i=1}^{M} \text{Lap}(y_{i,n}|\mathbf{C}_{(i)}\mathbf{x}_n, R_i). \tag{91}$$

To make approximate inference tractable, we make a mean-field approximation that induces a factorization between the two latent variables.

$$p(\mathbf{x}_n, z_n|\mathbf{y}_n) \approx q(\mathbf{x}_n, z_n) = q_x(\mathbf{x}_n)q_z(z_n) \tag{92}$$

The optimal distributions that maximize the lower bound

$$\mathcal{L}(q) = \langle \ln p(\mathbf{y}_n, \mathbf{x}_n, z_n) \rangle_q - \langle \ln q(\mathbf{x}_n, z_n) \rangle_q \tag{93}$$

are given by the calculus of variations:

$$\ln q_x(\mathbf{x}_n) = \langle \ln p(\mathbf{y}_n, \mathbf{x}_n, z_n) \rangle_{q_z} + \text{const.} \tag{94}$$

$$\ln q_z(z_n) = \langle \ln p(\mathbf{y}_n, \mathbf{x}_n, z_n) \rangle_{q_x} + \text{const.} \tag{95}$$

Solving for these optimal distributions gives the following iterative algorithm:

$$\widetilde{\mathbf{y}}_n = \mathbf{y}_n - \mathbf{C}\boldsymbol{\mu}_n \tag{96}$$

$$\xi_n = \widetilde{\mathbf{y}}_n^{\mathsf{T}} \mathbf{R}^{-1} \widetilde{\mathbf{y}}_n + \text{Tr}(\mathbf{R}^{-1}(\mathbf{C}\mathbf{V}_n\mathbf{C}^{\mathsf{T}})) \tag{97}$$

$$\langle z_n^{-1} \rangle = \sqrt{\frac{2}{\xi_n}} \frac{\mathcal{K}_{M/2}\left(\sqrt{2\xi_n}\right)}{\mathcal{K}_{M/2-1}\left(\sqrt{2\xi_n}\right)} \tag{98}$$

$$\mathbf{S}_n = \mathbf{C}\mathbf{P}_{n-1}\mathbf{C}^{\mathsf{T}} + \mathbf{R}\langle z_n^{-1} \rangle^{-1} \tag{99}$$

$$\mathbf{K}_n = \mathbf{P}_{n-1}\mathbf{C}^{\mathsf{T}}\mathbf{S}_n^{-1} \tag{100}$$

$$\boldsymbol{\mu}_n = \mathbf{m}_{n-1} + \mathbf{K}_n(\mathbf{y}_n - \mathbf{C}\mathbf{m}_{n-1}) \tag{101}$$

$$\mathbf{V}_n = (\mathbf{I} - \mathbf{K}_n\mathbf{C})\mathbf{P}_{n-1} \tag{102}$$

where $\mathcal{K}_m(x)$ is the modified Bessel function of the second kind [38]. Note that $\langle z_n^{-1} \rangle \neq \langle z_n \rangle^{-1}$. A logical way to initialize the algorithm is with $\boldsymbol{\mu}_n = \mathbf{m}_{n-1}$ and $\mathbf{V}_n = \mathbf{P}_{n-1}$.

The variational lower bound is useful for monitoring the algorithm's convergence and is given by

$$\begin{aligned}\mathcal{L}(q) = &\frac{M}{4} \ln \frac{1}{2\pi^2 \xi_n} + \frac{1}{2} \ln 2\xi_n + \ln \mathcal{K}_{\frac{M}{2}-1}(\sqrt{2\xi_n}) \\ &- \frac{1}{2}(\boldsymbol{\mu}_n - \mathbf{m}_{n-1})^{\mathsf{T}} \mathbf{P}_{n-1}^{-1} (\boldsymbol{\mu}_n - \mathbf{m}_{n-1}) - \frac{1}{2} \mathrm{Tr}(\mathbf{P}_{n-1}^{-1} \mathbf{V}_n) \\ &- \frac{1}{2} \left(D + \ln \det(\mathbf{R}) + \ln \det(\mathbf{P}_{n-1}) - \ln \det(\mathbf{V}_n) \right) . \quad (103)\end{aligned}$$

Due to numerical round-off errors, estimated covariance matrices $\mathbf{P}_{n-1}$ and $\mathbf{V}_n$ may not be positive semi-definite and, therefore, may not have positive determinants. In practice, a simple approach to retain the necessary positive semi-definite condition is to enforce symmetry, *e.g.* $\mathbf{V}_n \leftarrow \frac{1}{2}(\mathbf{V}_n + \mathbf{V}_n^{\mathsf{T}})$.

REFERENCES

[1] R. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the American Society for Mechanical Engineering, Series D, Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[2] G. Hewer, R. Martin, and J. Zeh, "Robust preprocessing for Kalman filtering of glint noise," *IEEE transactions on Aerospace and Electronic Systems*, vol. 23, no. 1, pp. 120–128, 1987.

[3] R. Pearson, "Outliers in process modeling and identification," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 1, pp. 55–63, 2002.

[4] I. Schick and S. Mitter, "Robust recursive estimation in the presence of heavy-tailed observation noise," *The Annals of Statistics*, vol. 22, no. 2, pp. 1045–1080, 1994.

[5] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[6] B. Chen, X. Liu, H. Zhao, and J. Principe, "Maximum correntropy Kalman filter," *Automatica*, vol. 76, pp. 70–77, 2016.

[7] C. Masreliez and R. Martin, "Robust Bayesian estimation for the linear model and robustifying the Kalman filter," *IEEE Transactions on Automatic Control*, vol. 22, no. 3, June 1977.

[8] Y. Huang, Y. Zhang, Y. Zhao, Z. Wu, and J. Chambers, "A novel robust Gaussian-Student's t mixture distribution based Kalman filter," *IEEE Transactions on Signal Processing*, vol. 67, no. 13, pp. 3606–3614, 2019.

[9] Y. Huang, Y. Zhang, P. Shi, Z. Wu, J. Qian, and J. Chambers, "Robust Kalman filters based on Gaussian scale mixture distributions with application to target tracking," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[10] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "Approximate inference in state-space models with heavy-tailed noise," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5024–5037, Oct. 2012.

[11] J.-A. Ting, E. Theodorou, and S. Schaal, "A Kalman filter for robust outlier detection," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, USA, Nov. 2007.

[12] T. Eltoft, T. Kim, and T. Lee, "On the multivariate Laplace distribution," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 300–303, May 2006.

[13] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*, 4th ed. John Wiley & Sons, Inc., 2011.

[14] S. Kotz, T. Kozubowski, and K. Podgorski, *The Laplace Distribution and Generalizations: A revisit with applications to communications, economics, engineering, and finance*. Springer Science+Business Media, LLC, 2001.

[15] A. Y. Aravkin, B. M. Bell, J. V. Burke, and G. Pillonetto, "An ℓ_1-Laplace robust Kalman smoother," *IEEE Transactions on Automatic Control*, vol. 56, no. 12, pp. 2898–2911, Dec. 2011.

[16] H. Wang, H. Li, W. Zhang, and H. Wang, "Laplace ℓ_1 robust Kalman filter based on majorization minimization," in *20th International Conference on Information Fusion*, Xi'an, China, July 2017.

[17] H. Wang, H. Li, W. Zhang, J. Zuo, and H. Wang, "Laplace ℓ_1 robust Kalman smoother based on majorization minimization," in *AIAA Scitech 2019 Forum*, 2019.

[18] L. Cau, D. Qiao, and X. Chen, "Laplace ℓ_1 Huber based cubature Kalman filter for attitude estimation of small satellite," *Acta Astronautica*, vol. 148, pp. 48–56, 2018.

[19] J. Neri, P. Depalle, and R. Badeau, "Laplace state space filter with exact inference and moment matching," in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Barcelona, Spain, May 2020, pp. 5880–5884.

[20] H. Rauch, C. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[21] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[22] T. Minka, "Expectation propagation for approximate Bayesian inference," in *Uncertainty in Artificial Intelligence*, 2001, pp. 362–369.

[23] H. Attias, "Inferring parameters and structure of latent variable models by variational Bayes," in *Uncertainty in Artificial Intelligence: Proceedings of the Fifth Conference*, 1999, pp. 21–30.

[24] M. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, University College London, May 2003.

[25] D. Blei, A. Kucukelbir, and J. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[26] T. Minka, "Divergence measures and message passing," Microsoft Research, Tech. Rep., 2005.

[27] L. D. Brown, *Fundamentals of Statistical Exponential Families with Applications in Statistical Decision Theory*. Lecture Notes - Monograph Series, Volume 9, Institute of Mathematical Statistics, 1986.

[28] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[29] J. Neri, P. Depalle, and R. Badeau, "Approximate inference and learning of state space models with Laplace noise - supplementary material," Tech. Rep., 2020.

[30] E. W. Ng and M. Geller, "A table of integrals of the error functions," *Journal of research of the National Bureau of Standards - B. Mathematical Sciences*, vol. 73B, no. 1, pp. 1–20, 1969.

[31] E. Wan and R. van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000.

[32] M. Chappell, A. Groves, B. Whitcher, and M. Woolrich, "Variational Bayesian inference for a nonlinear forward model," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 223–236, Jan. 2009.

[33] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[34] Z. Ghahramani and G. Hinton, "Parameter estimation for linear dynamical systems," CRG-TR-96-2, Department of Computer Science, University of Toronto,, Tech. Rep., 1996.

[35] S. Chiappa and D. Barber, "Dirichlet mixtures of Bayesian linear Gaussian state-space models: a variational approach," Max Planck Institute for Biological Cybernetics, Tech. Rep. 161, Mar. 2007.

[36] S. Boyd and L. Vandenberghe, *Convex Optimization*, 7th ed. Cambridge University Press, 2009.

[37] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings F - Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993.

[38] M. Abramowitz and I. Stegun, Eds., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 9th ed. New York: Dover, 1972.