Nonlinear Kalman Filtering With Divergence Minimization

San Gultekin and John Paisley

***Abstract*—We consider the nonlinear Kalman filtering problem using Kullback–Leibler (KL) and α-divergence measures as optimization criteria. Unlike linear Kalman filters, nonlinear Kalman filters do not have closed form Gaussian posteriors because of a lack of conjugacy due to the nonlinearity in the likelihood. In this paper, we propose novel algorithms to approximate this posterior by optimizing the forward and reverse forms of the KL divergence, as well as the α-divergence that contains these two as limiting cases. Unlike previous approaches, our algorithms do not make approximations to the divergences being optimized, but use Monte Carlo techniques to derive unbiased algorithms for direct optimization. We assess performance on radar and sensor tracking, and options pricing, showing general improvement over the extended, unscented, and ensemble Kalman filters, as well as competitive performance with particle filtering.**

***Index Terms*—Nonlinear Kalman filtering, Kullback-Leibler divergence, α-divergence, variational inference, Monte Carlo.**

I. Introduction

MODELING and analysis of time-varying signals is one of the most important subfields of signal processing. The problem arises in many different forms, such as communications data sent over a channel, video and audio data, and real-time tracking. A wide variety of algorithms have been developed in the statistics and engineering communities to deal with such dynamic systems. One classic algorithm is the Kalman filter [1], which performs minimum mean square error estimation of the hidden state of a time-varying linear system. The Kalman filter is recursive and online, making it suitable for real-time signal processing applications. Another advantage is its optimality for a large class of state-space models.

Kalman filtering has been applied extensively in control, communication, and signal processing settings, such as robot motion control and radar target tracking. With the recent explosions in sequential and streaming data, Kalman filters have also become a promising means for approaching machine learning problems, such as natural language processing [2], collaborative filtering [3] and topic modeling [4].

Manuscript received May 1, 2017; revised July 30, 2017; accepted August 22, 2017. Date of publication September 14, 2017; date of current version September 30, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sotirios Chatzis. *(Corresponding author: San Gultekin.)*

The authors are with the Department of Electrical Engineering and the Data Science Institute, Columbia University, New York, NY 10027 USA (e-mail: sg3108@columbia.edu; jpaisley@columbia.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2017.2752729

An important issue that often arises requiring modification to the basic Kalman filtering framework is nonlinearity. For example, in radar tracking distance and bearing measurements require a Cartesian-to-polar transformation [5], whereas the dynamic collaborative filtering model contains a bilinear form in two unknown vectors [6]. The nonlinear problem has been studied extensively in the literature, resulting in well-known filtering algorithms such as the extended Kalman filter (EKF) [7] and unscented Kalman filter (UKF) [8]. On the other hand, Monte Carlo (MC) methods have been developed [9], which are nonparametric and can represent any probability distribution using a discrete set of points, also referred to as particles.

While particle filters (PF) can approximate arbitrary densities, it may still be important to find the best *parametric* distribution according to a particular objective function. This has been a major goal in Bayesian inference, where the exact posterior distribution is usually intractable and approximated by a known, simpler distribution. Two established ways to handle this problem are variational inference [10] and expectation-propagation [11], in which the Kullback-Leibler (KL) divergence between the true posterior and the approximating distribution are minimized. Ideas from approximate inference have also been used in the Kalman filtering framework [12]–[14]. However, a thorough analysis of posterior optimization for nonlinear Kalman filters has not yet been made.

In this paper we fill this gap by presenting three algorithms for nonlinear Kalman filtering based on three respective divergence measures for posterior approximation, each based on a parametric form (in our case, a multivariate Gaussian). These approximations are obtained by algorithms for *approximation-free* divergence minimization. The divergence measures we consider are: 1) the forward KL divergence as used in variational inference; 2) the reverse KL divergence as used in expectation-propagation; and 3) the α-divergence, which is a generalized family that contains the former two as special cases. We also show that well-known algorithms such as the EKF and UKF are actually solving *approximations* to KL divergence minimization problems. This further motivates our study to address these shortcomings.[1]

The main machinery we use for obtaining these minimization algorithms is importance sampling. However, the resulting algorithms are all computationally cheaper than particle filtering since 1) no resampling is necessary, and 2) the number of

[1]We emphasize that our methods are still approximate in that the true non-Gaussian posterior will be approximated by a Gaussian. It is approximation-free in that the three algorithms directly optimize the three divergences.

unnecessary samples can be reduced by our proposed adaptive sampling procedure. We show advantages of our algorithms for target tracking and options pricing problems compared with the EKF, UKF and particle filter.

We organize the paper as follows: In Section II we define our filtering framework by reviewing the Kalman filter and discussing its non-linear variants. In particular, we discuss parametric approaches, also called assumed density filters, and nonparametric approaches, also called particle filters. In Section II-B we present three divergence minimization problems based on the forward and reverse KL divergences and the α-divergence. For each case we propose an algorithm which minimizes the corresponding objective function. Our algorithms are based on Monte Carlo integration techniques. Section IV contains a number of experiments to show how these divergence measures compare with each other and with standard approaches. Finally we conclude in Section V.

II. Kalman Filtering

A. Basic Linear Framework

The Kalman filter [1] has been developed and motivated as an optimal filter for linear systems. A key property is that this optimality is assured for general state-space models. This has made Kalman filtering widely applicable to a range of applications that make linearity assumptions. The Kalman filter can be written compactly at time step t as

$$x_t = F_t x_{t-1} + w_t, \qquad y_t = H_t x_t + v_t, \tag{1}$$

where w_t and v_t are independent zero-mean Gaussian random vectors with covariances Q_t and R_t respectively.[2] The latent variable $x_t \in \mathbb{R}^d$ is the unobserved state of the system. The vector $y_t \in \mathbb{R}^p$ constitutes the measurements made by the system.

The two main tasks of Kalman filtering are prediction and posterior calculation [7],

$$\begin{aligned} p(x_t|y_{1:t-1}) &= \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}, \\ p(x_t|y_{1:t}) &\propto p(y_t|x_t)p(x_t|y_{1:t-1}). \end{aligned} \tag{2}$$

When the initial distribution on $p(x)$ is Gaussian these calculations are all in closed form and are Gaussian, which is an attractive feature of the linear Kalman filter.

B. Nonlinear Framework

For many problems the measurements y_t involve nonlinear functions of x_t. In this case the Kalman filter becomes nonlinear and the closed-form posterior calculation discussed above no longer applies. The nonlinear process is

$$x_t = F_t x_{t-1} + w_t, \qquad y_t = h(x_t) + v_t, \tag{3}$$

where the noise process is the same as in (1), but $h(\cdot)$ is a nonlinear function of x_t.[3] While formally Bayes' rule lets us write

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int p(y_t|x_t)p(x_t|y_{1:t-1})dx_t}, \tag{4}$$

the normalizing constant is no longer tractable and the distribution $p(x_t|y_{1:t})$ is not known. Although the nonlinearity in h may be required by the problem, a drawback is the loss of fast and exact analytical calculations. In this paper we discuss three related techniques to approximating $p(x_t|y_{1:t})$, but first we review two standard approaches to the problem.

C. Parametric Approach: Assumed Density Filtering

To address the computational problem posed by (4), assumed density filters (ADF) project the nonlinear update equation to a tractable distribution. Building on the linear Gaussian state-space model, Gaussian assumed density filtering has found wide applicability [7], [8], [15], [16], [17]. The main ingredient here is an assumption of *joint Gaussianity* of the latent and observed variables. This takes the form,

$$p(x_t, y_t) \sim N\left(\begin{bmatrix}\mu_x \\ \mu_y\end{bmatrix}, \begin{bmatrix}\Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy}\end{bmatrix}\right). \tag{5}$$

(We've suppressed some time indexes and conditioning terms.) Under this joint Gaussian assumption, by standard computations the conditional distribution $p(x_t|y_t)$ is

$$\begin{aligned} p(x_t|y_t) &= N\left(\mu_{x|y}, \Sigma_{x|y}\right), \\ \mu_{x|y} &= \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y_t - \mu_y), \\ \Sigma_{x|y} &= \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}. \end{aligned} \tag{6}$$

In this case, the conditional distribution is also the posterior distribution of interest. Using this approximation, Kalman filtering can be carried out. For reference we provide the prediction and update equations in Appendix A.

There are several methods for making this approximation. We briefly review the two most common here: the extended Kalman filter and the unscented Kalman filter. The EKF approximates h using the linearization

$$h(x_t) \approx h(x_0) + H(x_0)(x_t - x_0),$$

where $H(x_0)$ is the Jacobian matrix evaluated at the point x_0. For example, x_0 could be the mean of the prior $p(x_t|y_{1:t-1})$. By plugging this approximation directly into the likelihood of y_t, the form of a linear Kalman filter is recovered and a closed form Gaussian posterior can be calculated.

As discussed in [8], the first-order approximation made by the EKF is often poor and performance can suffer as a result. Instead, they propose to estimate the quantities in (5) with an unscented transform—a numerical quadrature method. This result in UKF, which has similar computational cost as the EKF and higher accuracy. Based on the calculated Gaussian prior

[2]Kalman's formulation in [1] is optimal for more general noise models, but Gaussian noise is the most common choice, which we also use in this paper.

[3]We focus on measurement nonlinearity in this paper, assuming the same state space model. The techniques described in this paper can be extended to nonlinearity in the state space as well.

$p(x_t|y_{1:t-1}) = N(x_t|\mu_x, \Sigma_{xx})$, the UKF selects a discrete set of sigma points at which to approximate μ_y, Σ_{yy} and Σ_{xy}. Let $d_x = \dim(x_t)$ and $N_s = 2 \times d_x + 1$. These sigma points $x^1, \ldots, x^{N_s}$ are

$$x^s = \begin{cases} \mu_x & \text{for } s = 0 \\ \mu_x + [\sqrt{(d_x + \lambda)\Sigma_{xx}}]_s & \text{for } s = 1, \ldots, d_x \\ \mu_x - [\sqrt{(d_x + \lambda)\Sigma_{xx}}]_{s-d_x} & \text{for } s = d_x + 1, \ldots, 2d_x \end{cases} \tag{7}$$

The vector $[\sqrt{(d_x + \lambda)\Sigma}]_s$ corresponds to the sth column of the Cholesky decomposition of the matrix $\sqrt{(d_x + \lambda)\Sigma}$. Positive weights w^s are also defined for each x^s. The constant λ controls these sigma point locations, as well as the weights (along with additional fixed parameters). These N_s locations are used to empirically approximate all means and covariances in (5). Once y_t is measured, the approximation of $p(x_t|y_t)$ can then be calculated using (6).

There are many extensions to the UKF framework such as cubature Kalman filtering [18] and quadrature Kalman filtering [17], which use different numerical quadratures to carry out the approximation, but still correspond to the joint Gaussian assumption of (5). With that said, however, not all Gaussian ADFs make a joint Gaussianity assumption. For example, methods based on expectation-propagation [11] use moment matching (e.g., [19]) to obtain a Gaussian posterior approximation without modifying the joint likelihood distribution. We focus on an EP-like method in Section III-B.

D. Nonparametric Approach: Particle Filtering

We have seen that the main theme of ADF is approximating the posterior with a pre-specified joint probability density; when this joint density is Gaussian then $p(x_t|y_t) \approx N(\mu_t, \Sigma_t)$. On the other hand, nonparametric versions use sampling for posterior approximation without making any density assumptions on the form of this posterior,

$$p(x_t|y_{1:t}) \approx \sum_{s=1}^{N_s} w_t^s \delta_{x_t^s}. \tag{8}$$

The positive weights w_t^s sum to one, and $\delta_{x_t^s}$ is a point mass at the location x_t^s. The main approach is to use particle filters, a method base on importance sampling. In the case of particle filtering with sequential importance resampling (SIR) [9], updating an empirical approximation of $p(x_t|y_{1:t})$ uses a uniform-weighted prior approximation, $p(x_t) \approx \sum_{s=1}^{N_s} \frac{1}{N_s} \delta_{x_t^s}$ to calculate the posterior importance weights

$$w_t^s \propto \frac{1}{N_s} p(y_t|x_t^s). \tag{9}$$

It then constructs the uniform-weighted prior approximation by sampling N_s times

$$x_{t+1}^s \overset{iid}{\sim} \sum_{s=1}^{N_s} w_t^s N(F_t x_t^s, Q_t), \quad p(x_{t+1}) \approx \sum_{s=1}^{N_s} \frac{1}{N_s} \delta_{x_{t+1}^s}$$

While SIR particle filters can adaptively approximate any posterior density, the double sampling has computational cost, making these filters considerably slower compared to the above ADF approaches. Another potential issue is the need to propagate particles between time frames, which can be prohibitively expensive in communication-sensitive distributed applications, such as sensor networks [14].

III. Three Divergence Minimization Approaches

In this section we discuss the three proposed divergence minimization approaches to the nonlinear Kalman filtering problem. These include the two directions of the Kullback-Leibler (KL) divergence as well as the related α-divergence that contains both KL divergence measures as limiting cases. In all cases, our goal is to approximate the intractable posterior distribution $p(x_t|y_t)$ with a multivariate Gaussian distribution $q(x_t) = N(\mu_t, \Sigma_t)$, using these three divergences as potential quality measures. Since we will directly optimize these objectives, we anticipate an improvement over the standard EKF and UKF approximations. In the following three subsections, we first present one divergence objective and review its tractability issues, followed by our approach to resolving this issue.

A. Approach 1: Forward KL Divergence Minimization

Given two distributions $p(x|y)$ and $q(x)$, the forward KL divergence is defined as

$$\mathrm{KL}[q\|p] = \int q(x) \ln \frac{q(x)}{p(x|y)} dx. \tag{10}$$

The KL divergence is always nonnegative, becomes smaller the more q and p overlap, and equals zero if and only if $q = p$. These properties of the KL divergence make it a useful tool for measuring how "close" two distributions are. It is not a distance measure however, as $\mathrm{KL}[q\|p] \neq \mathrm{KL}[p\|q]$; we discuss the latter in Section III-B. In Bayesian machine learning, minimizing an objective of this form over q is known as variational inference (VI) [20]. In this case, $p(x|y)$ corresponds to an unknown posterior distribution of the model parameters, and q is its simpler approximation.

For the nonlinear Kalman filtering problem, the posterior is on the latent state vector x_t and so is intractable. Therefore, as is often the case, $\mathrm{KL}[q\|p]$ is not calculable. Variational inference [20], [21] instead uses the identity

$$\ln p(y) = \mathcal{L}(q, p(y, x)) + \mathrm{KL}[q(x)\|p(x|y)], \tag{11}$$

where

$$\mathcal{L}(q, p(y, x)) = \int q(x) \ln \frac{p(y, x)}{q(x)} dx. \tag{12}$$

This often is tractable since the joint distribution $p(y, x)$ is defined by the model. Since the marginal $\ln p(y)$ is constant and $\mathrm{KL} \geq 0$, variational inference instead maximizes $\mathcal{L}$ with respect to parameters of $q(x)$ to equivalently minimize KL.

While nonlinear Kalman filters have a simply-defined joint likelihood $p(y_t, x_t|y_{1:t-1})$ at time t, a significant problem still arises in calculating $\mathcal{L}$ due to the nonlinear function h. That is, if we define $q(x_t) = N(\mu_t, \Sigma_t)$, then for the Gaussian generative

process of (3) we optimize μ_t and Σ_t over the function

$$\begin{aligned}\mathcal{L} = & -\frac{1}{2}\mathbb{E}_q[(y_t - h(x_t))^\top R^{-1}(y_t - h(x_t))] \\ & + \mathbb{E}_q[\ln p(x_t|y_{1:t-1})] - \mathbb{E}_q[\ln q(x_t)] + \text{const.}\end{aligned} \tag{13}$$

The terms in the second line are tractable, but in the first line the nonlinearity of $h(x_t)$ will often result in an integral not having a closed form solution.

In the variational inference literature, common approaches to fixing this issue typically involve making tractable approximations to $h(x_t)$. For example, one such approximation would be to pick a point x_0 and make the first-order Taylor approximation $h(x_t) \approx h(x_0) + H(x_0)(x_t - x_0)$. One then replaces $h(x_t)$ in (13) with this approximation and optimizes $q(x_t)$. In fact, in this case the resulting update of $q(x_t)$ is identical to the EKF. This observation implies a correspondence between variational inference and commonly used approximations to the non-linear Kalman filters such as the EKF. We make this formal in the following theorem.

Theorem 1: Let the joint Gaussian ADF correspond to the class of filters which make the joint distribution assumption in (5). Then, all filters in this class optimize an approximate form of the variational lower bound in (13).

We present a complete proof in Appendix B. Theorem 1 is general in that it contains the most successfully-applied ADFs such as the EKF and UKF, among others. For the special case of EKF, the nature of this approximation is more specific.

Corollary 2: The EKF corresponds to optimizing the objective (13) using a first order Taylor approximation of h.

Please see Appendix C for a derivation. Consequently, the existing algorithms *modify* $\mathcal{L}$ and so the resulting optimization of this approximation is no longer guaranteed to minimize KL$[q\|p]$. Instead, in this paper we are motivated to fill in this gap and find ways to *directly* optimize objectives such as (13), and thus minimize divergence measures between q and the intractable posterior $p(x_t|y_t)$. We next devise a method for KL$[q\|p]$.

Recently Paisley, *et al*. [22] proposed a stochastic method for sampling *unbiased* gradients of $\mathcal{L}$, allowing for approximation-free minimization of the forward KL divergence using stochastic gradient descent. We derive this technique for the nonlinear Kalman filter, which will allow for approximate posterior inference having smaller KL divergence than the EKF and UKF. Using simpler notation, we seek to maximize an objective of the form,

$$\mathcal{L} = \mathbb{E}_q[f(x_t)] + \mathbb{E}_q[\ln p(x_t)] - \mathbb{E}_q[\ln q(x_t)] \tag{14}$$

$$f(x_t) = -\frac{1}{2}(y_t - h(x_t))^\top R^{-1}(y_t - h(x_t)) \tag{15}$$

over the parameters of $q(x_t) = N(x_t|\mu_t, \Sigma_t)$, and thereby minimize KL$[q\|p]$. This can be done by gradient ascent. However, since $\mathbb{E}_q[f(x_t)]$ does not have a closed form solution, $\nabla\mathcal{L}$ cannot be evaluated analytically. The proposed solution in [22] is to instead step in the direction of an unbiased stochastic gradient. To this end, the observation is made that

$$\nabla\mathcal{L} = \mathbb{E}_q[f(x_t)\nabla \ln q(x_t)] + \nabla\mathbb{E}_q\Big[\ln \frac{p(x_t)}{q(x_t)}\Big], \tag{16}$$

where the identity $\nabla q(x_t) = q(x_t)\nabla \ln q(x_t)$ is used. While the second gradient can be calculated analytically with respect to either μ_t or Σ_t, the first gradient can be sampled using Monte Carlo integration,

$$\mathbb{E}_q[f(x_t)\nabla \ln q(x_t)] \approx \frac{1}{S}\sum_{s=1}^{S} f(x_t^s)\nabla \ln q(x_t^s), \quad x_t^s \overset{iid}{\sim} q(x_t). \tag{17}$$

A second observation is made by [22] that the variance of these samples may be so large that S needs to be too large a number to make this approximation computationally feasible. For this reason employing variance reduction methods is crucial. Paisley, *et al*. [22] propose introducing a *control variate* $g(x_t)$ that is highly correlated with $f(x_t)$, but has an analytic expectation $\mathbb{E}_q[g(x_t)]$. The gradient of $\mathcal{L}$ with a control variate is equal to

$$\begin{aligned}\nabla\mathcal{L} = & \ \mathbb{E}_q[(f(x_t) - \lambda g(x_t))\nabla \ln q(x_t)] + \lambda\nabla\mathbb{E}_q[g(x_t)] \\ & + \nabla\mathbb{E}_q[\ln p(x_t)] - \nabla\mathbb{E}_q[\ln q(x_t)].\end{aligned} \tag{18}$$

Though this leaves the gradient unchanged, MC sampling of the first term has much smaller variance when $|corr(f, g)|$ is large (calculated using $q(x_t)$). The parameter $\lambda \in \mathbb{R}$ is set to minimize the variance.[4] Intuitively, this can be seen by noting that if $f(x_t^s) \approx \lambda g(x_t^s)$ at the sampled values x_t^s, then $|f(x_t^s) - \lambda g(x_t^s)| \ll |f(x_t^s)|$. In this case, the analytic gradient $\lambda\nabla\mathbb{E}_q[g(x_t)]$ gives an initial approximation of $\mathbb{E}_q[f(x_t)\nabla \ln q(x_t)]$, which is then corrected to be made unbiased by the MC-sampled $\mathbb{E}_q[(f(x_t) - \lambda g(x_t))\nabla \ln q(x_t)]$. Since $g(x_t)$ is a good approximation of $f(x_t)$ in the region of high probability defined by $q(x_t)$, the analytic approximation captures most information, but is refined by the MC-sampled gradient to make the method approximation-free.

The requirements on $g(x_t)$ to be a good control variate for $f(x_t)$ are that: 1) it is an approximation of $f(x_t)$, and 2) the expectation $\mathbb{E}_q[g(x_t)]$ is solvable. There are many possible control variates for the function $(y - h(x))^\top R^{-1}(y - h(x))$. However, building on the EKF framework we propose setting

$$\begin{aligned}g(x_t) = & \ -\frac{1}{2}(y_t - \widetilde{h}(\mu_t, x_t))^\top R^{-1}(y_t - \widetilde{h}(\mu_t, x_t)) \\ \widetilde{h}(\mu_t, x_t) = & \ h(\mu_t) + H(\mu_t)(x_t - \mu_t)\end{aligned} \tag{19}$$

We let μ_t be the current value of the mean of $q(x_t)$ at a given iteration of time t. If we define $\widetilde{y}_t = y_t - h(\mu_t) + H(\mu_t)\mu_t$, then equivalently we can write

$$g(x_t) = -\frac{1}{2}(\widetilde{y}_t - H(\mu_t)x_t)^\top R^{-1}(\widetilde{y}_t - H(\mu_t)x_t). \tag{20}$$

The expectation is now in closed form. While a better approximation may have greater variance reduction for a fixed number

[4] As shown in [22], when $\lambda \equiv cov(f, g)/var(g)$ (approximated by sampling) the variance reduction ratio is $var(f - \lambda g)/var(f) = 1 - corr(f, g)^2$.

of MC-samples, we emphasize this would not make the algorithm "more correct." Where the EKF simply *replaces* $f(x)$ with $g(x)$, our stochastic gradient approach then *corrects* the error of this approximation.

Next we derive the unbiased gradients. In this case, these gradients are $\nabla_{\widehat{\mu}_t}\mathcal{L}$ and $\nabla_{\widehat{\Sigma}_t}\mathcal{L}$. We note that, if we were using the EKF framework, by replacing $f(x_t)$ with $g(x_t)$ in (15), the roots of these gradients could be solved and the EKF solutions for $\widehat{\mu}_t$ and $\widehat{\Sigma}_t$ would result. However, since we have the additional stochastic gradient term, we must perform gradient ascent. The final expressions for the unbiased gradients using samples $x_t^s \sim_{iid} q(x_t)$ are:

$$\nabla_{\widehat{\mu}_t}\mathcal{L} = \frac{1}{S}\sum_{s=1}^{S}\left[f(x^s) - g(x^s)\right]\left[\widehat{\Sigma}_t^{-1}x^s - \widehat{\Sigma}_t^{-1}\widehat{\mu}_t\right] \tag{21}$$
$$+\ \Sigma_t^{-1}(\mu_t - \widehat{\mu}_t) + H(x_0)^\top R^{-1}(\widetilde{y}_t - H(x_0)\widehat{\mu}_t),$$
$$\nabla_{\widehat{\Sigma}_t}\mathcal{L} = \frac{1}{S}\sum_{s=1}^{S}\left[f(x^s) - g(x^s)\right]$$
$$\times\ \frac{1}{2}\left[\widehat{\Sigma}_t^{-1}(x^s - \widehat{\mu}_t)(x^s - \widehat{\mu}_t)^\top\widehat{\Sigma}_t^{-1} - \widehat{\Sigma}_t^{-1}\right]$$
$$+\ \frac{1}{2}(\widehat{\Sigma}_t^{-1} - \Sigma_t^{-1}) - \frac{1}{2}H(x_0)^\top R^{-1}H(x_0). \tag{22}$$

It is important to note that, without proper scaling of the gradients we can easily have a numerically unstable algorithm as the covariance matrix can lose its positive definiteness. To fix this we pre-condition the gradients with a symmetric positive definite matrix C and perform the following updates

$$\widehat{\mu}_t^{(i+1)} = \widehat{\mu}_t^{(i)} + \rho^{(i)}\,[C^{(i)}\,\nabla_{\widehat{\mu}_t}\mathcal{L}], \tag{23}$$
$$\widehat{\Sigma}_t^{(i+1)} = \widehat{\Sigma}^{(i)} + \rho^{(i)}\,[C^{(i)}\,\nabla_{\widehat{\Sigma}_t}\mathcal{L}\,C^{(i)}]. \tag{24}$$

We highlight the difference between index t and i. The first is the time frame we are currently processing, while the second is the iteration number at time t since we are using a gradient optimization method. For the conditioning matrix we choose $C^{(i)} = [\Sigma_t^{(i)}]^{-1}$, which approximates the natural gradient [23] for $\widehat{\mu}$ and $\widehat{\Sigma}$. When the step size satisfies the Robbins-Monro conditions, $\sum_{i=1}^{\infty}\rho^{(i)} = \infty$ and $\sum_{i=1}^{\infty}[\rho^{(i)}]^2 < \infty$, the gradients in (23)–(24) converge to a fixed point of the exact variational lower bound. In practice we can, for example, choose $\rho^{(i)} = (w+i)^{-\eta}$ with $\eta \in (0.5, 1]$ and $w \geq 0$. In simulations we observed that when natural gradients are employed a generic schedule for step sizes can be used and no further hand-tuning is necessary. We refer to this algorithm as *stochastic search Kalman filtering* (SKF) and summarize it in Algorithm 1 for a single time step.

Algorithm 1 SKF: stochastic search Kalman filter.

1: **Input:** Posterior $q(x_{t-1})$, sample size S, and iterations I.
2: Calculate prior $p(x_t) = N(\mu_t, \Sigma_t)$.
3: **for** $i = 1, \ldots, I$ **do**
4: Sample $x_t^s \sim_{iid} q(x_t)$ for $s = 1, \ldots, S$.
5: Compute $\nabla_{\widehat{\mu}_t}\mathcal{L}$ and $\nabla_{\widehat{\Sigma}_t}\mathcal{L}$ as in (21) and (22).
6: Update

$$\widehat{\mu}_t^{(i+1)} = \widehat{\mu}_t^{(i)} + \rho_i\,[C^{(i)}\,\nabla_{\widehat{\mu}_t}\mathcal{L}]$$
$$\widehat{\Sigma}_t^{(i+1)} = \widehat{\Sigma}_t^{(i)} + \rho_i\,[C^{(i)}\,\nabla_{\widehat{\Sigma}_t}\mathcal{L}\,C^{(i)}]$$

7: **Return** $q(x_t) = \mathcal{N}(\widehat{\mu}_t, \widehat{\Sigma}_t)$

B. Approach 2: Reverse KL Divergence Minimization

As mentioned in Section III-A, the KL divergence is not a distance measure because it is not symmetric. The complement of the forward KL divergence defined in (10) is the reverse KL divergence:

$$\mathrm{KL}[p\|q] = \int_{\mathcal{X}} p(x)\log\frac{p(x)}{q(x)}dx. \tag{25}$$

We can see that (25) offers an alternative measure of how similar two probability distributions are; therefore we can use it to approximate an intractable posterior distribution.

Note that for either objective function, (10) or (25), the optimal solution will be $q(x) = p(x|y)$. However, since the approximating distribution family is typically different from the exact posterior distribution family, the two optimization problems will give different solutions in practice. In particular, the reverse KL divergence has shown to be a better fit for unimodal approximations, while forward KL works better in the multimodal case [21]. Consequently, we can expect that optimizing the reverse KL will be a better choice for the nonlinear Kalman filtering problem (this is supported by our experiments). In Section III-A, finding a fixed point of the forward KL problem required an iterative scheme for maximizing the variational objective function. The fixed point of the reverse KL has a more interpretable form, as we will show.

To this end, we first note that an exponential family distribution has the form

$$q(x) = h(x)\exp\{\eta^\top s(x) - \log A(\eta)\},$$

where η is the natural parameter and $s(x)$ is the sufficient statistic. Therefore inference in exponential families correspond to determining η. Substituting this parametrized form in (25) and setting the derivative with respect to the natural parameter equal to zero, one can show that

$$0 = \nabla_\eta \mathrm{KL}[p\|q] = \mathbb{E}_q[s(x)] - \mathbb{E}_p[s(x)],$$

which follows from the exponential family identity $\nabla_\eta \log A(\eta) = \mathbb{E}_q[s(x)]$. Therefore the fixed points of the objective are given by

$$\mathbb{E}_q[s(x)] = \mathbb{E}_p[s(x)]. \tag{26}$$

This moment matching is well-known in statistics, machine learning, and elsewhere [21]. In machine learning it appears prominently in expectation-propagation (EP) [11], [24].

A common choice for the approximating exponential family distribution is again Gaussian because it is the maximum entropy

distribution for the given first and second order moments [20]. Since a Gaussian is completely specified by its mean and covariance, when the approximating distribution $q(x)$ is selected to be Gaussian, the optimal solution is simply found by matching its mean and covariance to that of $p(x|y)$.

Therefore, in the context of exponential families the task of finding the optimal distribution for the reverse KL reduces to the task of matching moments. However, there is still a difficulty in the need to compute the moments of an unknown posterior distribution. Fortunately, Monte Carlo methods prove useful here as well. Let $\mathbb{E}_q[f(x)]$ be the expectation we wish to compute. For example, choosing $f(x) = x$ and $f(x) = xx^\top - \mathbb{E}[x]\mathbb{E}[x]^\top$ gives the mean and covariance respectively. This expectation can be approximated as

$$\begin{aligned}\mathbb{E}_q[f(x)] &= \int f(x)\frac{p(x|y)}{\pi(x)}\pi(x)dx,\\ &= \int f(x)\frac{[p(y|x)p(x)]/\pi(x)}{\int p(y|x')p(x')dx'}\pi(x)dx,\\ &\approx \sum_{s=1}^{S} f(x^s)\frac{[p(y|x^s)p(x^s)]/\pi(x^s)}{\sum_j [p(y|x^j)p(x^j)]/\pi(x^j)}, \end{aligned} \tag{27}$$

where $x^s \sim_{iid} \pi(x)$. We will define

$$w^s = \frac{p(y|x^s)p(x^s)}{\pi(x^s)}, \quad W = \sum_s \frac{p(y|x^s)p(x^s)}{\pi(x^s)},$$

and so $\mathbb{E}_q[f(x)] \approx \frac{1}{W}\sum_{s=1}^{S} f(x^s)w^s$. This is related to importance sampling, with the added normalizer W. As we can see from (27) this procedure is biased as it is a ratio of two approximations, yet it converges to the true expectation $\mathbb{E}_q[f(x)]$ almost surely. Therefore, we have an asymptotically unbiased divergence minimization procedure. We call this the *moment matching Kalman filter* (MKF) and summarize it in Algorithm 2 using the Gaussian distribution. We observe that a major difference between the MKF and SKF of the previous section is that the MKF only needs to sample once to obtain the moment estimates for a time step. Therefore, the MKF is not an iterative algorithm and is much faster. Also, the MKF requires slightly less computation compared to particle filtering because it eliminates the need for resampling.

Algorithm 2 MKF: moment matching Kalman filter.

1: **Input:** Posterior $q(x_{t-1})$, sample size S, proposal dist. π_t.
2: Calculate prior $p(x_t) = N(\mu_t, \Sigma_t)$.
3: Sample $x^s \sim_{iid} \pi_t(x_t)$ for $s = 1, \ldots, S$.
4: Calculate $w^s = p(y_t|x^s)p(x^s)/\pi_t(x^s)$, $W = \sum_{s=1}^{S} w^s$.
5: Approximate the moments of $p(x_t|y_t)$ as

$$\widehat{\mu}_t = \frac{1}{W}\sum_{s=1}^{S} w^s x^s$$

$$\widehat{\Sigma}_t = \frac{1}{W}\sum_{s=1}^{S} w^s (x^s - \widehat{\mu}_t)(x^s - \widehat{\mu}_t)^\top$$

6: **Return** $q(x_t) = \mathcal{N}(\widehat{\mu}_t, \widehat{\Sigma}_t)$

C. Approach 3: α-Divergence Minimization

In Sections III-A and III-B we showed how nonlinear Kalman filtering can be performed by minimizing the forward and reverse KL divergences. A further generalization is possible by considering the α-divergence, which contains both KL divergences as a special case. Following [24] we define the α-divergence to be

$$D_\alpha[p\|q] = \frac{1}{\alpha(1-\alpha)}\left(1 - \int p(x)^\alpha q(x)^{1-\alpha}dx\right), \tag{28}$$

where the parameter α can take any value in $(-\infty, \infty)$. Some special cases are

$$\lim_{\alpha\to 0} D_\alpha[p\|q] = \mathrm{KL}[q\|p], \lim_{\alpha\to 1} D_\alpha[p\|q] = \mathrm{KL}[p\|q],$$
$$D_{\frac{1}{2}}[p\|q] = 2\int (\sqrt{p(x)} - \sqrt{q(x)})^2 dx = 4\mathrm{Hel}^2[p\|q], \tag{29}$$

where $\mathrm{Hel}[p\|q]$ is the Hellinger distance. We see that when $\alpha = 1/2$ we get a valid distance metric. Similar to before, we seek a q-distribution which approximates $p(x|y)$, where approximation quality is now measured by the α-divergence.

We again begin assuming that the approximating distribution is in the exponential family, $q(x) = h(x)\exp\{\eta^\top s(x) - \log A(\eta)\}$. The gradient of the α-divergence shows that

$$\begin{aligned} 0 = \nabla_\eta D_\alpha[p\|q] &= \frac{1-\alpha}{Z_{\widetilde{p}}}\int \widetilde{p}(x)\big[s(x) - \mathbb{E}_q[s(x)]\big]\\ &= \mathbb{E}_{\widetilde{p}}[s(x)] - \mathbb{E}_q[s(x)]. \end{aligned} \tag{30}$$

Note that we defined a new probability distribution $\widetilde{p}(x) = p(x)^\alpha q(x)^{1-\alpha}/Z_{\widetilde{p}}$ where the denominator term is the cumulant function. This leads to a *generalized* moment matching condition,

$$\mathbb{E}_q[s(x)] = \mathbb{E}_{\widetilde{p}}[s(x)]. \tag{31}$$

This problem is more complicated than the reverse KL version because the left hand side also depends on the q-distribution. The α-divergence generalizes a number of known divergence metrics. In context of EP, it is possible to obtain a generalization which is called Power EP [25]. More recently, [24] used a similar black-box optimization, where they showed that by varying the value of α the algorithm varies between variational inference and expectation propagation. It turns out that, for many practical problems, using a fractional value of α can give better performance than the limiting cases $\alpha = 0$ or $\alpha = 1$. This motivates our following α-divergence minimization scheme.

Algorithm 3 αKF : α-divergence Kalman filter.

1: **Input:** Posterior $q(x_{t-1})$, sample size S, and proposal π_t.
2: Calculate prior $p(x_t) = N(\mu_t, \Sigma_t)$.
3: Sample $x^s \sim_{iid} \pi_t(x_t)$ for $s = 1, \ldots, S$.
4: Calculate $w^s = \frac{[p(y_t|x^s)p(x^s)]^\alpha q(x^s)^{1-\alpha}}{\pi_t(x^s)}$, $W = \sum_{s=1}^S w^s$
5: Approximate the moments of $\widetilde{p}(x_t)$

$$\widehat{\mu}_t = \frac{1}{W}\sum_{s=1}^{S} w^s x^s$$

$$\widehat{\Sigma}_t = \frac{1}{W}\sum_{s=1}^{S} w^s (x^s - \widehat{\mu}_t)(x^s - \widehat{\mu}_t)^\top$$

6: **Return** $q(x_t) = \mathcal{N}(\widehat{\mu}_t, \widehat{\Sigma}_t)$

A similar importance sampling approach can be used as for the reverse KL divergence. Using similar notation, we can write

$$\mathbb{E}_{\widetilde{p}}[f(x)] = \int f(x)\frac{p(x|y)^\alpha q(x)^{1-\alpha}}{\pi(x)}\pi(x)dx, \tag{32}$$

$$\approx \sum_{s=1}^{S} f(x^s)\frac{[p(y|x^s)p(x^s)]^\alpha q(x^s)^{1-\alpha}/\pi(x^s)}{\sum_j [p(y|x^j)^\alpha p(x^j)^\alpha q(x^j)^{1-\alpha}]/\pi(x^j)},$$

where $x^s \sim_{iid} \pi(x)$. Again we define

$$w^s = [p(y|x^s)p(x^s)]^\alpha q(x^s)^{1-\alpha}/\pi(x^s),\ W = \sum_s w^s.$$

We see that the procedure in (27) is a special case of this when we set $\alpha = 1$. However, there is a significant difference in that the moment matching of (26) can be done in one iteration since it only depends on p. In (32) the q distribution appears on both sides of the equality. This is similar to EP and Power-EP algorithms, where multiple iterations can be run to update q. Upon convergence we know that the solution is a fixed point of (28), but convergence of the procedure is not guaranteed and multiple iterations might degrade the performance. In our experiments we will only iterate once to avoid possibly diverging and also to keep the cost of the algorithm the same as that of MKF in the previous section. We call this algorithm α-divergence Kalman filter (αKF) and summarize it in Algorithm 3. We note that the only difference between αKF and MKF is in step 4.

We can get a better understanding of α-divergence by analyzing the weight coefficients. In particular, assume that we choose the proposal distribution to be the prior, $\pi(x) = p(x)$. Then the MKF weights are $w^s \propto p(y|x^s)$ and the αKF weights are $w^s \propto p(y|x^s)^\alpha$. Therefore, the likelihood term is scaled by alpha, and as $\alpha \to 0$ all particles will have equal contribution. For very low values of α this will discard information, but for intermediate values this can alleviate the impact of sharply changing likelihood values. As we will show in our experiments, when the measurement noise is strong, choosing an intermediary α value provides robustness.

D. Adaptive Sampling

The main parameter in the implementation of sampled filters such as particle filters and the three filters proposed here is the number of particles S to use. Hence it is desirable to have a method for estimating the number of samples necessary for a given degree of accuracy. This computed sample size can be used to adaptively reduce the computation as much as possible, while still maintaining a desired accuracy. In Fig. 1 we illustrate this problem for tracking a moving target. At time t the target makes an abrupt maneuver that requires more particles for accurate tracking, but this size can be reduced afterwards.

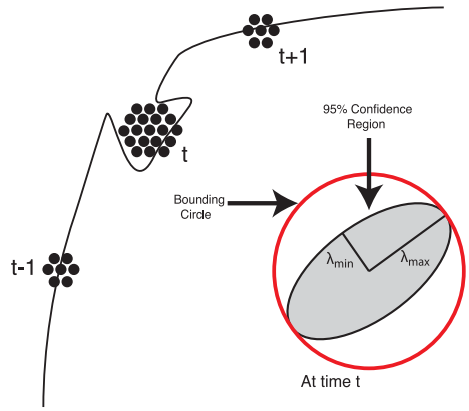


Fig. 1. Illustration of adaptive sampling. Due to unexpected changes in a target trajectory, more samples may be needed at a given time point. Also shown is the bounding circle for a confidence ellipsoid in two dimensions.

Since the approximate Gaussian posterior distribution has a parametric form, we are able to use an adaptive sampling method for the MKF and αKF [5]. To determine the appropriate number of samples, we measure the uncertainty of our mean approximation for $q(x_t) = \mathcal{N}(\mu_t, \Sigma_t)$, where $\mu_t = \sum_{s=1}^S w^s x^s / W$. For importance sampling, the variance of this estimator is approximately

$$\mathbb{V}(\mu_t) \approx \sum_{s=1}^{S}\left[\frac{w^s}{W}\right]^2 (x^s - \mu_t)(x^s - \mu_t)^\top. \tag{33}$$

Here, if S is large enough, the estimator can be approximated as normal by the central limit theorem [26]. We use this fact to compute the radius of a 95% confidence region. Assuming that the estimator is zero mean, which is justified by the asymptotic unbiasedness of the unnormalized importance sampling procedure, we denote the estimator by $\widehat{X} \sim N(0, \mathbb{V}(\mu_t))$. It follows that

$$P(\widehat{X}^\top \mathbb{V}(\mu_t)^{-1}\widehat{X} \leq \chi_d^2(p)) = p, \tag{34}$$

where $\chi_d^2(p)$ is the quantile function of a chi-squared distribution with d degrees of freedom (set to the state-space dimension), and p is the probability value (for 95% confidence intervals this is set to $p = 0.95$). The region described by (34) is a hyper-ellipsoid, so the maximum possible radius will correspond to the major axis, which is given by

$$r_{\max} = \sqrt{\lambda_{\max}(\mathbb{V}(\mu_t)) \times \chi_d^2(0.95)}. \tag{35}$$

[5]For the SKF the per-iteration sample size is much smaller, so there is less benefit in using this technique.

Note that this is a conservative estimate, as the hypersphere with radius $r_{\max}$ will typically be much larger than the hyperellipsoid. An illustration of the *bounding circle* for a 2D multivariate normal distribution is given in Fig. 1.

Now assume that, based on a small sample set S_{base}, we wish to estimate the minimum number of samples $S_{\min}$ required to achieve a certain $r_{\max}$. We have the result that

$$\frac{r_1}{r_2} \propto \sqrt{\frac{S_2}{S_1}}, \qquad S_{\min} = S_{\text{base}} \times \left[\frac{r_{\text{base}}}{r_{\max}}\right]^2. \tag{36}$$

As expected, the smaller radius we desire, the larger sample size we need.

IV. Experimental Results

We experiment with all three proposed nonlinear Kalman filtering algorithms, as well as the EKF, ENKF, UKF and particle filter, on radar and sensor tracking problems, as well as an options pricing problem.

A. Target Tracking

The first problem we consider is target tracking. This problem arises in various settings, but here we consider two established cases: radar and sensor networks. The radar tracking problem has been a primary application area for nonlinear Kalman filtering. Wireless sensor networks are another emerging area where nonlinear filtering is useful. Driven by the advances in wireless networking, computation and micro-electro-mechanical systems (MEMS), small inexpensive sensors can be deployed in a variety of environments for many applications [27], [28].

For both problems the state-space has the form

$$\begin{aligned} x_t &=, F_t x_{t-1} + w_t \quad w_t \sim N(0, Q_t), \\ y_t &=, h(x_t) + v_t, \quad v_t \sim N(0, R_t). \end{aligned} \tag{37}$$

Here, F_t and Q_t model the dynamics of target motion and are usually time-varying. On the other hand, $h(\cdot)$ specifies the equipment that performs the measurements, and the environment and equipment based inaccuracies are represented by R_t. In the radar setting, when the target is far away and the angle measurement noise is strong enough, the problem can become highly nonlinear. For sensor networks, the nonlinearity is caused by the small number of active sensors (due to energy constraints) with large measurement noise (due to the attenuation in received signal) [29]. While the value of R_t can be determined to some extent through device calibration, it is more challenging to do this for Q_t [30].

Our experiments are based on synthetic data using a constant velocity model in $\mathbb{R}^2$ which corresponds to the state vector vector $x_t = [x_1,\ \dot{x_1},\ x_2,\ \dot{x_2}]^\top$; the second and fourth entries correspond to the velocity of the target in each dimension. Following [31], we set the parameters for the state variable equation to

$$F_t = \begin{bmatrix} F_2 & 0 \\ 0 & F_2 \end{bmatrix}, \quad F_2 = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \tag{38}$$

$$Q_t = \begin{bmatrix} Q_2 & 0 \\ 0 & Q_2 \end{bmatrix}, \quad Q_2 = \sigma_{CV} \begin{bmatrix} \Delta t^4/4 & \Delta t^3/2 \\ \Delta t^3/2 & \Delta t^2 \end{bmatrix}. \tag{39}$$

The radar measures the distance and bearing of the target via the nonlinear function $h(\cdot)$ of the target location,

$$h(x_t) = \left[\sqrt{x_t(1)^2 + x_t(3)^2},\ \tan^{-1}[x_t(3)/x_t(1)]\right]^\top,$$

i.e., the Cartesian-to-polar transformation [5]. For the sensor networks problem, we will consider a scenario which uses range-only measurements from multiple sensors. This yields the model in (37) where $h(\cdot)$ is the measurement function such that the i-th dimension (i.e., measurement of sensor s_i) is given by

$$[h(x_t)]_i = \sqrt{[x_t(1) - s_i(1)]^2 + [x_t(2) - s_i(2)]^2},$$

and the length of $h(x_t)$ will be the number of activated sensors at time t.

We consider two types of problems: tracking with uncertain parameters and tracking with known parameters. For the case of uncertain parameters, we set the radar and sensor simulation settings as follows. First, we note that for both simulations we assume a constant measurement rate, and so $\Delta t = 1$. For radar we sweep the process noise values in (38) as $\sigma_{CV} \in \{10^{-3},\ 2 \times 10^{-3},\ \ldots, 10^{-2}\}$. We generate 20 data sets for each value of σ_{CV}, yielding a total of 200 experiments. For the measurement noise we use a diagonal R with entries $\sigma_r^2 = 10^{-1}$ and $\sigma_\theta^2 = 10^{-2}$ which controls the noise of distance and bearing measurements, respectively. The initial state is $x_0 = [1000,\ 10,\ 1000,\ 10]^\top$; this distance from origin and angle noise variance results in a severely nonlinear model, making filtering quite challenging. For sensor network simulations, we use the same constant-velocity model of (38) with $\sigma_{CV} = 10^{-2}$. We deploy 200 sensors and at each time point there are exactly 3 activated for range measurements. The sensors are scattered over a square field of 100×100 units sampled from a uniform distribution. For reference all sensors are shown as background in Fig. 2. The measurement covariance matrix is $R = \sigma_R^2 I$ where we set $\sigma_R = 20$. We set the initial state $x_0 = [1000,\ 1,\ 1000,\ 1]^\top$. With this, once again, we obtain a highly nonlinear system, albeit less severe than the radar case. We also consider the case where the generating parameters are known to the filter. In this case, we assess the performance of the filter as a function of process and measurement noise covariances. We sweep $\sigma_{\text{CV}} \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$ and $\sigma_r \in \{10, 15, 20, 25, 30\}$.

We implemented EKF, UKF, sequential importance resampling particle filter (PF), ensemble Kalman filter (ENKF) and our proposed SKF, MKF, and αKF for $\alpha = 0.5$. For SKF we use 500 particles per iteration, whereas we consider 10^4 particles for PF, MKF and αKF . We will later show results as a function of particles. When there is parameter uncertainty, the exact value of Q is not known to the filter, therefore we consider a scaled isotropic covariance of form $\sigma_Q^2 I$.

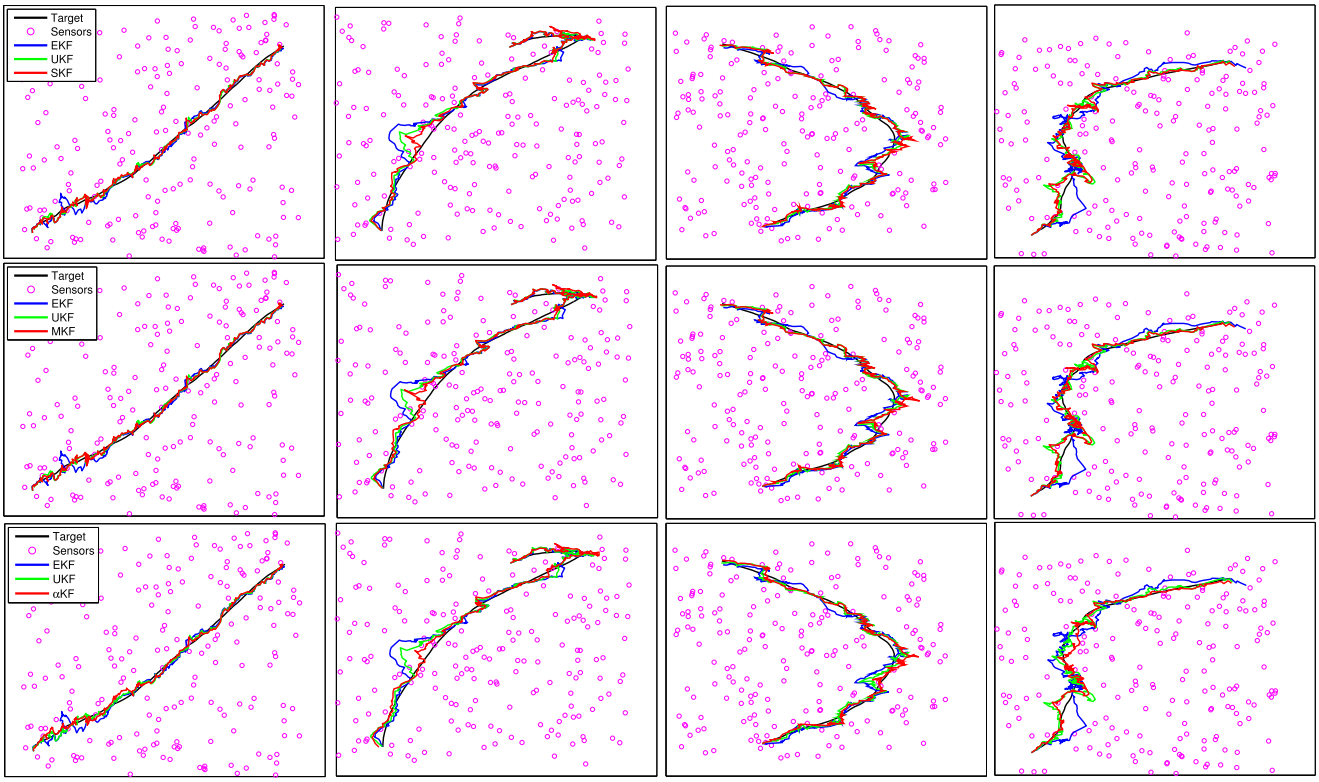


Fig. 2. Tracks estimated by various filtering schemes in sensor network setting. Top row: Comparisons of EKF, UKF, and SKF. Middle row: EKF, UKF, and MKF. Bottom row: EKF, UKF, and αKF . In the background sensor scatterplots are given. Each plot corresponds to a square field with 100 units of side length. Best viewed in color.

TABLE I
RADAR TRACKING PROBLEM: MEAN SQUARE ERROR (MSE) OF VARIOUS FILTERING SCHEMES AS A FUNCTION OF PROCESS NOISE PARAMETER σ_Q. THE BOLDFACES SHOW THE BEST PERFORMERS FOR SMALL/LARGE PARTICLE SIZES

	σ_Q				
	10^{-2}	5×10^{-2}	10^{-1}	5×10^{-1}	1
SKF	41.4100	34.6611	29.9952	42.1360	38.0507
MKF	31.3088	27.6861	29.0376	35.2422	39.2536
αKF	30.8783	27.9475	**27.4130**	31.0271	34.9420
PF	28.5429	32.3768	35.1842	44.3704	48.9767
ENKF	33.8646	35.3385	37.1761	42.2874	45.7732
EKF	33.8611	35.8086	37.7808	42.6595	45.9788
UKF	31.7528	31.8616	33.7625	41.1282	45.4806
BASE	223.5281	223.5281	223.5281	223.5281	223.5281

1) Quantitative results: In Table I we show mean square error (MSE) for radar tracking as a function of the selected scale value (σ_Q). Here, the base error corresponds to the estimations based on measurements only, and its order-of-magnitude difference from filter MSE values show the severity of nonlinearity. Comparing MSE values we see that MKF and αKF outperforms EKF and UKF for all settings of σ_Q, which shows that the Gaussian density obtained from these filters is indeed more accurate. SKF also has better results, particularly for $\sigma_Q = 10^{-1}$ but is less robust to the changes in scale value. This is due to the iterative gradient scheme employed by SKF, which could give worse results depending on parameter changes or covariance initializations. Since MKF/αKF are based on importance sampling, they do not exhibit the same sensitivity. As for PF, this algorithm also produces competitive results when $\sigma_Q = 10^{-2}$; however its performance significantly deteriorates (even more than that of SKF) as σ_Q increases, which shows that nonparametric inference of particle filtering is more sensitive to parameter uncertainty. Lastly, ENKF has significantly worse performance than all other sampling based filters. This result is mainly because, ENKF lacks a scheme to weight the samples, ans so is more sensitive to parameter uncertainties. We observe that αKF has the highest robustness to parameter changes, making it a better choice when parameters are not known; moreover, αKF is more robust excess measurement noise, as discussed in Section III-C.

Table II presents MSE results for sensor networks. Unlike the radar problem, all particle-based filters are better than EKF and UKF for all values of σ_Q. This reduced sensitivity is due to the reduced nonlinearity in the problem. The performance of SKF, MKF, and PF are similar to each other, MKF being the favorable choice for most cases. This time ENKF does a better job, since the nonlinearity is less challenging, yet it is still inferior to UKF. Again αKF is the best performer, where the best performance is achieved at $\sigma_Q = 10^{-1}$.

2) Qualitative results: In Fig. 2 we show qualitative tracking results with sensor networks. The top, middle, and bottom rows correspond to SKF, MKF, and αKF respectively. For each row we pick four different paths (shared across different rows) and for each plot we show the true trajectory along with EKF, UKF, and one of our filters. By visual inspection we can see that our

TABLE II
SENSOR NETWORK TRACKING PROBLEM: MEAN SQUARE ERROR (MSE) OF VARIOUS FILTERING SCHEMES AS A FUNCTION OF PROCESS NOISE PARAMETER σ_Q. THE BOLDFACES SHOW THE BEST PERFORMERS FOR SMALL/LARGE PARTICLE SIZES

	σ_Q				
	10^{-2}	5×10^{-2}	10^{-1}	5×10^{-1}	1
SKF	10.4674	9.5812	9.5038	10.1664	10.5996
MKF	10.5572	9.2879	9.1684	9.8175	10.3307
αKF	9.9441	8.0913	**8.0623**	9.1002	9.7055
PF	9.5661	9.3464	9.4726	10.0422	10.3834
ENKF	10.6565	13.8709	11.5082	13.8403	15.2449
EKF	14.0034	13.9357	14.5161	15.5277	16.1438
UKF	11.5303	10.3639	10.2068	10.8830	11.5845

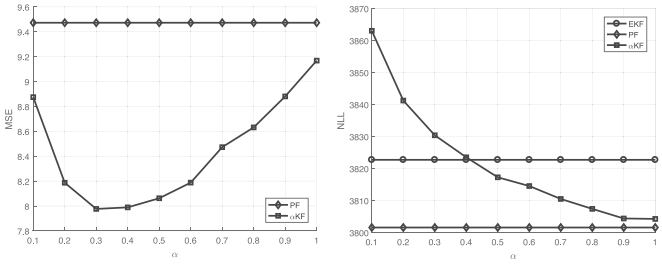


Fig. 3. Left: MSE value of αKF as a function of α for the sensor network tracking problem with $\sigma_Q = 10^{-1}$. Right: NLL values as a function of α. For both figures, when $\alpha = 1$, αKF reduces to MKF. The performance of PF and EKF are plotted as baselines. Also, for PF and EKF the markers are only given for reference, otherwise they do not depend on α.

algorithms provide more accurate tracking, which gives visual meaning to the quantitative results.

3) Discussion on α: For αKF we have only considered the case when $\alpha = 0.5$. We next focus on varying α. For this experiment we set $\sigma_Q = 10^{-1}$, which corresponds to the middle column of Table II. We plot the mean squared error as a function of α in Fig. 3. We see that low-mid ranges of α (i.e $0.3 - 0.5$) give the best MSE results. This improvement is a result of lower values of α mitigating the effects of strong measurement noise. There is a trade-off, however, since choosing α too small will discard to much of information from the measurement and give poor results. This is seen for lower values of α, where decreasing the parameter degrades performance. We note that ideal values of α may differ for different sensor characteristics. Since ground truth is necessary to know the best value, simulations using known sensor characteristics can allow for selecting this parameter using a grid search as shown in Fig. 3. Indeed, this is the main difficulty of tuning α on the fly; to illustrate the challenge, note that when the prior distribution is used as the proposal, the weights for the MC samples are obtained by scaling likelihood terms.

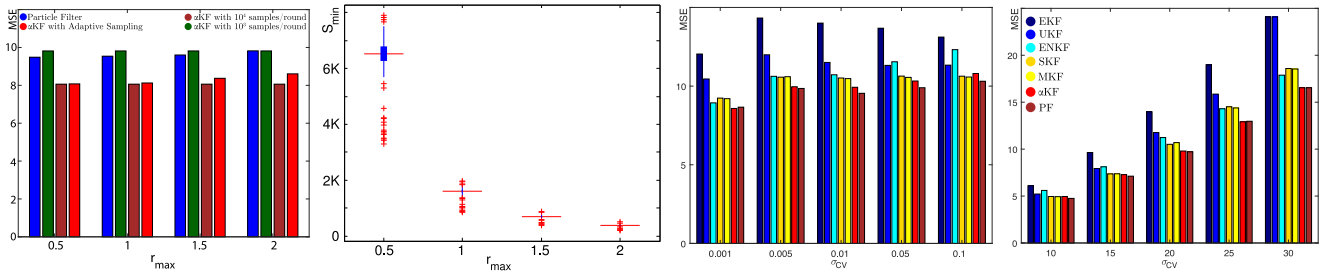
One consideration for tuning the α parameter would be to maximize the likelihood. However, this does not work in practice. In Fig. 3 we show the negative log likelihood (NLL) for the sensor network experiment using the same setup from the left pane. We see that the highest likelihood/lowest NLL is obtained by setting $\alpha = 1$. This is not surprising, as this value corresponds to zero dampening of the likelihood term itself. As we decrease α, NLL increases as well. In fact, while the best value of lowest MSE is $\alpha = 0.3$, its corresponding NLL is even worse than EKF. Indeed, the worse performance for EKF is tied to sensitivity to this likelihood term, which in the context of machine learning is similar to "overfitting." For this reason, a separate set of data may be useful to tune the α parameter.

4) Discussion on adaptive sampling: As discussed in Section III-D, one can use adaptive sampling to choose the minimum possible sample size to achieve a certain confidence region radius, $r_{\max}$. We implemented adaptive sampling for αKF using an initial batch size of $S_{\text{base}} = 500$. We picked four different values of $r_{\max}$ from $\{0.5, 1, 1.5, 2\}$. Fig. 4(a) displays the results for this experiment. Here we compare the MSE results as a function of $r_{\max}$ for αKF and PF for the sensor tracking problem with $\sigma_{\text{CV}} = 0.1$. Note that for PF adaptive sampling is not a choice since all particles should be propagated, resampled, and updated at every time step. So for PF we simply set the sample size as the average $S_{\min}$ for the αKF for each case. We can see that the MSE performances differ very little across different cases, showing that for larger target values of $r_{\max}$ both methods can still produce accurate estimates of the true state. This is also visible in our comparison to the αKF with sample size fixed to $S = 10^4$, but performance clearly degrades for $S = 10^3$, showing the advantage of not having to set this parameter. We also see that αKF outperforms PF in all cases. On the other hand, 4(a) shows the number of samples required to achieve a certain confidence radius. From this figure we can see the $\mathcal{O}(1/r^2)$ decaying rate of $S_{\min}$ as implied by (36). Given the high accuracy in the left panel, we see that on the order of hundreds of samples can be sufficient for high-quality estimates, which makes αKF competitive for real time applications.

5) Experiments with known process noise: We now turn to the case where the process noise parameter is known. In Fig. 4 we show MSE as a function of σ_{CV} and σ_R. For the measurement noise, as σ_R increases, overall MSE also increases, while for process noise this trend is not present. For both cases we see that the particle filter gives the best result overall. This is expected, since when the parameters are known perfectly particle filters can approximate the posterior with more accuracy since it is nonparametric. However, αKF is still competitive in this setting. In fact, for several cases, such as $\sigma_{\text{CV}} = 0.001$ and $\sigma_r = 25$, performance of αKF and PF are equal, while both filters perform much better than SKF and MKF in all cases.

B. Options Pricing

We also consider a problem from options pricing. In finance, an option is a derivative security which gives the holder a right to buy/sell (call/put option) the underlying asset at a certain price on or before a specific date. The underlying asset can be, for example, a stock. The price and date are called the strike price and expiry date respectively. The value of the option, called the premium, depends on a number of factors. Let C and P denote the call and put prices. We use σ and r to denote volatility and risk-free interest rate, respectively. The values of these variables are not directly observed and need to be estimated. Let S denote



(a) MSE as a function of $r_{\max}$. (b) Samples required to achieve $r_{\max}$. (c) MSE as function of process noise. (d) MSE versus measurement noise.

Fig. 4. Mean square error and minimum sample size as a function of confidence radius $r_{\max}$. Mean square error as a function of process and measurement noise parameters, where the exact parameters are known to the filter. The legend given is shared by both figures. (Best viewed in color.)

the price of the underlying asset and X the strike price. Finally, let t_m denote the time to maturity; this is the time difference between the purchase and expiry dates which is written as a fraction of a year. For example, an option which expires in two months will have $t_m = 1/6$.

Accurate pricing of options is an important problem in mathematical finance. For a European style option, the price as a function of these parameters can be modeled using the well-known Black-Scholes [32]

$$d_1 = \frac{\log(S/X) + (r + \sigma^2/2)t_m}{\sigma\sqrt{t_m}},\ d_2 = d_1 - \sigma\sqrt{t_m},$$
$$C = S\Phi(d_1) - Xe^{-rt_m}\,\Phi(d_2),$$
$$P = -S\Phi(-d_1) + Xe^{-rt_m}\,\Phi(-d_2). \tag{40}$$

Following the approach of [33], let $x = [\sigma\ r]^\top$ be the state and $y = [C\ P]^\top$ be the measurement. We therefore have the following state space representation

$$x_t = x_{t-1} + w_t,\ w_t \sim N(0, Q),$$
$$y_t = h(x_t) + v_t,\ v_t \sim N(0, R), \tag{41}$$

where the nonlinear mapping $h(\cdot)$ is given by (40). In this case we model the process and measurement noises with time-invariant covariance matrices Q and R. We consider two tasks: 1) predicting the one-step ahead prices, and 2) estimating the values of hidden state variables. This problem is also considered in [34] to assess the performance of particle filtering algorithms.

We use the Black-Scholes model as the ground truth. In order to synthesize the data, we use historical values of VIX (CBOEINDEX:VIX), which measures the volatility of S&P 500 companies. From this list we pick Microsoft (NASDAQ:MSFT), Apple (NASDAQ:AAPL), and IBM (NYSE:IBM) as underlying assets and use their historical prices. The interest rate comes from a state-space model with a process noise of zero mean and variance 10^{-4}. We set $\sigma_Q = \sigma_R = 10^{-2}$. In Table III we show the next-day prediction performance of all algorithms. We can see that the prediction performance improves as we move towards MKF. This again shows the difference between Gaussian approximations of the methods we employ. For MKF and PF we used 10^3 particles however we once again note that MKF can achieve this performance without resampling, and it can leverage adaptive sampling to reduce sample size. On the other hand, for SKF we need to use a large number of particles per iterations (around 1,000). Even though this gives better results then EKF and UKF it is much slower than MKF and PF, and its performance can vary significantly between iterations. On the other hand, since the measurement noise is small, choosing $\alpha < 1$ for αKF does not provide improvement over MKF in this case, which is consistent with our previous discussion. Therefore $\alpha = 1$ is the best choice in this case.

Fig. 5 shows the volatility estimation for three filters. EKF tends to over/under-shoot and UKF is significantly better in that respect. However, MKF improves on these two, giving the most

TABLE III
MEAN ABSOLUTE ERROR (MAE) VALUES OF VARIOUS FILTERING SCHEMES FOR THREE DIFFERENT CALL/PUT OPTION PAIRS; CALCULATED FOR $\sigma_Q = 10^{-2}$. FOR OPTION 3, EKF LOSES TRACK SO MAE IS NOT REPORTED.

		EKF	UKF	PF	SKF	MKF
Option 1 MAE	Call	0.1352	0.0788	0.0663	0.0658	0.0654
	Put	0.1528	0.0789	0.0668	0.0642	0.0654
Option 2 MAE	Call	0.0425	0.0354	0.0329	0.0312	0.0319
	Put	0.0478	0.0355	0.0340	0.0368	0.0331
Option 3 MAE	Call	-	0.2155	0.1584	0.1573	0.1586
	Put	-	0.2158	0.1579	0.1574	0.1586

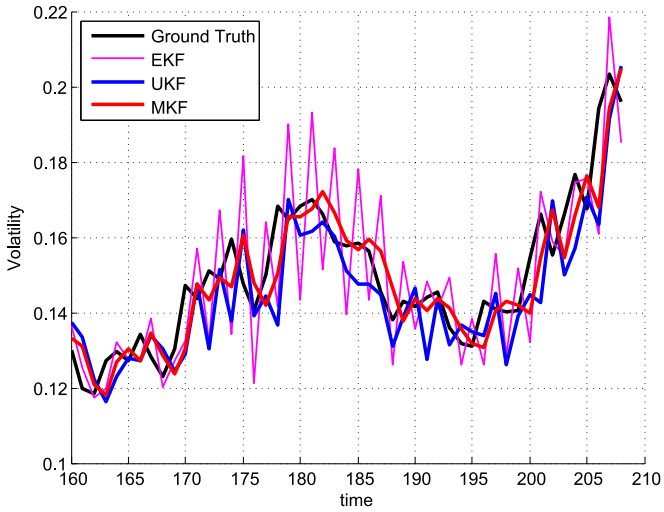


Fig. 5. Volatility estimation performance of various filtering schemes (based on Option 1). The estimates are plotted along with the ground truth. Best viewed in color.

robust estimates. We report that the plot of SKF was similar to MKF. Also, as with the target tracking experiments, MKF has better performance than SKF, which agrees with the observation that expectation-propagation typically outperforms variational inference for unimodal posteriors.

V. Conclusion

We have considered nonlinear Kalman filtering as a divergence minimization problem. In particular, we introduced three algorithms which directly minimize the forward and reverse Kullback-Leibler divergences, as well as the α-divergence, the last divergence being a generalization of the previous two. While our algorithms are based on sampling techniques, our focus has been on finding the optimal parametric distribution according to a divergence measure. We also showed how joint Gaussian assumed density filters such as the EKF and UKF optimize an approximation to the variational lower bound, meaning they only give approximately optimal solutions to this divergence.

We have conducted experiments to test the proposed methods on radar and sensor network problems, as well as options pricing. In addition to promising performance, we showed that we can obtain filters which are robust to high measurement noise. We hope that this work can serve as a building block for designing a class of filters which optimize any arbitrary divergence directly. For example, it is possible to consider heavy-tailed parametric densities or multimodal densities and build dynamic filters on top of this.

Appendix A
ADF Equations

For the proofs in the following appendices we need the predict-update equations of the joint Gaussian ADFs. Note that this corresponds to the model in (5). The equations are summarized as

$$\begin{aligned}
\text{Predict:}\quad & x_{t|t-1} = F_t x_{t-1|t-1},\\
& P_{t|t-1} = F_t P_{t-1|t-1} F_t^\top + Q_t, && (42)\\
\text{Update:}\quad & x_{t|t} = x_{t|t-1} + K_t(y_t - y_{t|t-1}),\\
& P_{t|t} = P_{t|t-1} - K_t S_t K_t^\intercal, && (43)\\
\text{Auxiliary:}\quad & y_{t|t-1} = \mu_y = h_t(x_{t|t-1}),\\
& H_t = \Sigma_{yx}\Sigma_{xx}^{-1},\\
& S_t = \Sigma_{yy} = \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy} + R_t,\\
& K_t = \Sigma_{xy}\Sigma_{yy}^{-1}. && (44)
\end{aligned}$$

We emphasize that these hold for *any* joint Gaussian ADF. When EKF is employed, H_t is the Jacobian at prior mean, and S_t and K_t are calculated accordingly.

Appendix B
Proof of Theorem 1

The joint Gaussian ADF corresponds to $f(x) \approx g(x)$; this approximation is constructed from $p(y_t|x_t)$ in (5), which is Gaussian with $\mu_{y|x} = \mu_y + \Sigma_{yx}\Sigma_{xx}^{-1}(x_t - \mu_x)$ and $\Sigma_{y|x} = \Sigma_{yy} - \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}$. This yields

$$g(x_t) = -\frac{1}{2}(\widetilde{y}_t - \Sigma_{yx}\Sigma_{xx}^{-1}x_t)^\top R_t^{-1}(\widetilde{y}_t - \Sigma_{yx}\Sigma_{xx}^{-1}x_t), \quad (45)$$

where $\widetilde{y}_t = y_t - \mu_y + \Sigma_{yx}\Sigma_{xx}^{-1}\mu_x$. Note that under (5) we have $p(x_t) \sim N(\mu_x, \Sigma_{xx})$, and let $q(x_t) \sim N(\widehat{\mu}, \widehat{\Sigma})$. Substituting $g(x_t)$ to (14) the expectations are now evaluated as

$$\begin{aligned}
-\mathbb{E}_q[\log q(x_t)] &= \frac{1}{2}\log|\widehat{\Sigma}_t|,\\
-\mathbb{E}_q[\log p(x_t)] &= -\frac{1}{2}\widehat{\mu}^\top\Sigma_{xx}^{-1}\widehat{\mu} - \frac{1}{2}tr\{\Sigma_{xx}^{-1}\widehat{\Sigma}_t\} + \widehat{\mu}_t\Sigma^{-1}\mu_x,\\
-\frac{1}{2}\mathbb{E}_q[g(x_t)] &= -\frac{1}{2}\widehat{\mu}_t^\top\Sigma_{xx}^{-1}\Sigma_{xy}R_t^{-1}\Sigma_{yx}\Sigma_{xx}^{-1}\widehat{\mu}_t\\
&\quad - \frac{1}{2}tr\{(\Sigma_{xx}^{-1}\Sigma_{xy}R_t^{-1}\Sigma_{yx}\Sigma_{xx}^{-1})\widehat{\Sigma}_t\} && (46)\\
&\quad + \widehat{\mu}_t\Sigma_{xx}^{-1}\Sigma_{xy}R_t^{-1}\widetilde{y}_t. && (47)
\end{aligned}$$

The posterior parameters are found by solving $\nabla_{\widehat{\mu}}\mathcal{L} = 0$ and $\nabla_{\widehat{\Sigma}}\mathcal{L} = 0$. Differentiating the terms in (47) we get

$$\widehat{\Sigma}_t = [\Sigma_{xx}^{-1} + \Sigma_{xx}^{-1}\Sigma_{xy}R_t^{-1}\Sigma_{yx}\Sigma_{xx}^{-1}]^{-1}, \quad (48)$$

$$\widehat{\mu}_t = \widehat{\Sigma}_t(\Sigma_{xx}^{-1}\mu_x + \Sigma_{xx}^{-1}\Sigma_{xy}R^{-1}\widetilde{y}_t). \quad (49)$$

The matrix inversion lemma asserts $(A+UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$; applying this to (48) we obtain

$$\begin{aligned}
\widehat{\Sigma}_t &= \Sigma_{xx} - \Sigma_{xy}(\Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy} + R_t^{-1})\Sigma_{yx},\\
&= \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yy}\Sigma_{yy}^{-1}\Sigma_{yx}. && (50)
\end{aligned}$$

Substituting (50) into (49) and expanding we get

$$\begin{aligned}
\widehat{\mu}_t &= \mu_x - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}\Sigma_{xx}^{-1}\mu_x\\
&\quad + \Sigma_{xy}R_t^{-1}\widetilde{y}_t - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}R_t^{-1}\widetilde{y}_t,\\
&= \mu_x - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}\Sigma_{xx}^{-1}\mu_x\\
&\quad + (I - \Sigma_{xx}\Sigma_{yy}^{-1}\Sigma_{yx}\Sigma_{xx}^{-1})\Sigma_{xy}R_t^{-1}\widetilde{y}_t,\\
&= \mu_x - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}\Sigma_{xx}^{-1}\mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}\widetilde{y}_t,\\
&= \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y_t - \mu_y). && (51)
\end{aligned}$$

Note the third line follows from the identity $\Sigma_{xy}\Sigma_{yy}^{-1} = (I - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}\Sigma_{xx}^{-1})\Sigma_{xy}R_t^{-1}$ which can be verified with straightforward manipulation. Matching the terms in (44) with (51) and (50) we obtain the updates in (43). ■

Appendix C
Proof of Corollary 2

The proof is similar to that of Theorem 1, therefore we highlight the key points. We simplify the notation to $p(x_t) \sim N(\mu_t, \Sigma_t)$ and $q(x_t) \sim N(\widehat{\mu}_t, \widehat{\Sigma}_t)$. We employ a first-order Taylor series expansion around prior mean: $h(x_t) \approx h(\mu_t) + H_t(\mu_t)(x_t - \mu_t)$ where H_t is the Jacobian. Define $\widetilde{y}_t = y_t - h(\mu_t) + H_t\mu_t$. Plugging these into the variational lower bound

(14) and differentiating we obtain

$$\widehat{\Sigma}_t = \left(\Sigma_t^{-1} + H_t^\top R_t^{-1} H_t\right)^{-1}, \tag{52}$$

$$\widehat{\mu}_t = \widehat{\Sigma}_t \left(\Sigma_t^{-1}\mu_t + H_t^\top R_t^{-1}\widetilde{y}_t\right). \tag{53}$$

Once again, using the matrix inversion lemma we get

$$\widehat{\Sigma}_t = \Sigma_t - K_t S_t K_t^\top, \tag{54}$$

where $S_t = H_t\Sigma_t H_t^\top + R_t$ and $K_t = \Sigma_t H_t^\top {S_t}^{-1}$. Plugging (54) in (53) and expanding the multiplication we get

$$\begin{aligned}\widehat{\mu}_t &= \mu_t + \Sigma_t H_t^\top R_t^{-1}\widetilde{y}_t - K_t H_t^\top \mu_t - K_t H_t \Sigma_t H_t^\top R_t^{-1}\widetilde{y}_t \\ &= \mu_t - K_t H_t^\top \mu_t + (I - K_t H_t)\Sigma_t H_t^\top R_t^{-1}\gamma \\ &= \mu_t - K_t H_t^\top \mu + K_t\widetilde{y}_t \\ &= \mu_t + K_t(y_t - h_t(\mu_t)) \end{aligned} \tag{55}$$

where the first and third lines utilize the identities $\widehat{\Sigma}_t = \Sigma_t - K_t H_t \Sigma_t$ and $K_t = (I - K_t H_t)\Sigma_t H_t^\top R_t^{-1}$ respectively. We see that (55) and (54) correspond to the EKF update equations. ■

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME–J. Basic Eng.*, vol. 82, no. Series D, pp. 35–45, 1960.

[2] D. Belanger and S. Kakade, "A linear dynamical system model for text," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 833–842.

[3] S. Gultekin and J. Paisley, "A collaborative Kalman filter for time-evolving dyadic processes," in *Proc. IEEE Int. Conf. Data Min.*, Dec. 2014, pp. 140–149.

[4] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 113–120.

[5] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking: III. Measurement models," in *Proc. Int. Symp. Opt. Sci. Technol.*, 2001, pp. 423–446.

[6] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[7] G. Welch and G. Bishop, "An introduction to the Kalman filter," Univ. North Carolina Chapel Hill, Chapel Hill, NC, USA, Tech. Rep. 95–041, 1995.

[8] S. K. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

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

[10] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, pp. 183–233, 1999.

[11] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *Proc. Uncertainty Artif. Intell.*, 2001, pp. 362–369.

[12] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, Gatsby Comput. Neurosci. Unit, Univ. London, London, U.K., 2003.

[13] J. Vermaak, N. D. Lawrence, and P. Perez, "Variational inference for visual tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2003, pp. I-773–I-780.

[14] J. Teng, H. Snoussi, C. Richard, and R. Zhou, "Distributed variational filtering for simultaneous sensor localization and target tracking in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 61, no. 5, pp. 2305–2318, Jun. 2012.

[15] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. San Diego, CA, USA: Academic, 1982.

[16] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Trans. Autom. Control*, vol. 45, no. 5, pp. 910–927, May 2000.

[17] D. Guo and X. Wang, "Quasi-monte carlo filtering in nonlinear dynamic systems," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2087–2098, Jun. 2006.

[18] B. Jia, M. Xin, and Y. Cheng, "High-degree cubature Kalman filter," *Automatica*, vol. 49, pp. 510–518, 2013.

[19] T. Heskes and O. Zoeter, "Expectation propagation for approximate inference in dynamic Bayesian networks," in *Proc. Uncertainty Artif. Intell.*, 2002, pp. 216–223.

[20] M. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, no. 1–2, pp. 1–305, 2008.

[21] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[22] J. Paisley, D. M. Blei, and M. I. Jordan, "Variational Bayesian inference with stochastic search," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1363–1370.

[23] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, Feb. 1998.

[24] J. M. Hernandez-Lobato, Y. Li, M. Rowland, D. Hernandez-Lobato, T. D. Bui, and R. E. Turner, "Black box alpha divergence minimization," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1511–1520.

[25] T. P. Minka, "Power EP," Microsoft Res., Cambridge, U.K., Tech. Rep. MSR-TR-2004–149, 2004.

[26] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Mach. Learn.*, vol. 50, no. 1–2, pp. 5–43, 2003.

[27] C. Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proc. IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.

[28] M. Tubaishat and S. Madria, "Sensor networks: An overview," *IEEE Potentials*, vol. 22, no. 2, pp. 20–23, Apr./May 2003.

[29] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Localization systems for wireless sensor networks," *IEEE Wireless Commun.*, vol. 14, no. 6, pp. 6–12, Dec. 2007.

[30] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. part V. multiple model methods," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1255–1321, Oct. 2005.

[31] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. part I. dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.

[32] J. C. Hull, *Options, Futures, and Other Derivatives*. London, U.K.: Pearson, 2006.

[33] M. Niranjan, "Sequential tracking in pricing financial options using model based and neural network approaches," in *Proc. Neural Inf. Process. Syst.*, 1997, pp. 960–966.

[34] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The unscented particle filter," in *Proc. Neural Inf. Process. Syst.*, 2000, pp. 584–590.

San Gultekin received the B.S. and M.S. degrees in the Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey. He is currently working toward the Ph.D. degree in the Department of Electrical Engineering, Columbia University, New York, NY, USA, where he is also an Armstrong Fellow. His research interests include statistical signal processing, machine learning, large scale data analysis, and time series analysis.

John Paisley received the B.S., M.S., and Ph.D. degrees in electrical engineering from Duke University, Durham, NC, USA. He was a Postdoctoral Researcher in the Computer Science Departments at UC Berkeley, Berkeley, CA, USA, and Princeton University, Princeton, NJ, USA. He is currently an Assistant Professor in the Department of Electrical Engineering, Columbia University, New York, NY, USA, where he is also a member in the Data Science Institute. His current research interests include machine learning, focusing on models and inference techniques for text, and image processing applications.