This is a repository copy of *On the variational Gaussian filtering with natural gradient descent*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/227277/

Version: Accepted Version

Proceedings Paper:
Li, X., Liu, Y., Yang, L. et al. (2 more authors) (2025) On the variational Gaussian filtering with natural gradient descent. In: Proceedings of 2025 28th International Conference on Information Fusion (FUSION). 2025 28th International Conference on Information Fusion (FUSION), 07-11 Jul 2025, Rio de Janeiro, Brazil. Institute of Electrical and Electronics Engineers (IEEE). ISBN: 9798331503505.

https://doi.org/10.23919/FUSION65864.2025.11124137

© 2025 The Author(s). Except as otherwise noted, this author-accepted version of a paper published in Proceedings of 2025 28th International Conference on Information Fusion (FUSION) is made available via the University of Sheffield Research Publications and Copyright Policy under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/

Reuse
This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

On the Variational Gaussian Filtering with Natural Gradient Descent

Xi Li[1◇], Yi Liu[2◇], Le Yang[2*], Lyudmila Mihaylova[3], Ji Li[4]
1. State Key Lab of Complex Electromagnetic Environment Effects on Electronics and Information Systems, National University of Defense Technology, Changsha China
2. Department of Electrical and Computer Engineering, University of Canterbury, Christchurch NZ
3. Department of Automatic Control and System Engineering, University of Sheffield, Sheffield UK
4. Locaris Electronic Technology Co. Ltd, Zhengzhou China
◇: Equal contributors, *: corresponding author, le.yang@canterbury.ac.nz

***Abstract*—Variational Gaussian filter (VGF) approximates the intractable posterior of the state of a non-linear non-Gaussian system using a single Gaussian density normally found through Kullback-Leibler divergence minimization. This paper focuses on the VGFs whose measurement update is realized by employing the natural gradient descent (NGD). Under the assumption that the state predictive distribution is also Gaussian, we re-examine the iterative NGD-based measurement update under two different parameterizations of the Gaussian posterior. The first one consists of the mean and covariance, while the other comprises the mean and precision matrix (i.e., the inverse of the covariance). Their NGD-based update rules are derived in an alternative but unified way using matrix calculus. They are compared against each other and with the one developed using the natural parameterization of the Gaussian density. Important new insights are obtained. Modifications to the established update rules, which guarantee the positive definiteness of the covariance/precision matrix of the Gaussian posterior, are re-visited as well. Simulations are used to corroborate the theoretical results and evaluate the performance of the developed algorithms in range-bearing tracking.**

I. Introduction

State estimation for a dynamic system based on the noisy measurements collected up to the current time, commonly known as filtering, has a variety of applications in navigation, tracking, signal processing and finance. Mathematically, the state filtering requires finding the posterior $p(\mathbf{x}_t|\mathbf{z}_{1:t})$, where $\mathbf{x}_t$ represents the system state at time t, and $\mathbf{z}_{1:t}$ denotes the measurements received so far. For a first-order Markov system, $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ can be evaluated recursively using [1]

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})\mathrm{d}\mathbf{x}_{t-1}, \quad (1a)$$

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})}. \quad (1b)$$

(1a) computes the predictive distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ using the state transition density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and state posterior from the previous time step. (1b) refines $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ to generate the current state posterior $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ via incoporating the information from the measurement $\mathbf{z}_t$ given in the likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$, a process known as the measurement update.

The recursion (1) generally does not admit a closed-form solution. One exception is when linear Gaussian systems are considered. In this case, evaluating (1) results in the celebrated Kalman filter (KF) [2]. In practice, however, system non-linearity and/or non-Gaussianity may arise, which makes solving (1) analytically intractable. The non-linearity could come from that the measurements, such as the range and bearing in tracking applications, depend in a non-linear manner on the state to be estimated [3]. The non-Gaussianity may be due to the presence of measurement outliers, which renders it necessary to adopt heavy-tailed distributions such as the Student's t-distribution in the likelihood (see e.g., [4]).

For non-linear non-Gaussian state estimation, particle filters [5], which represent the posterior using a large number of weighted particles, may be used. An alternative approach is the assumed density filtering with Gaussian assumption, or simply referred to as the Gaussian filter (GF). It approximates the state predictive distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ in (1a) and state posterior $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ in (1b) using two Gaussian densities $\mathcal{N}(\mathbf{x}_t;\mathbf{m}_t,\mathbf{P}_t)$ and $\mathcal{N}(\mathbf{x}_t;\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)$. Here, $\mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$ denotes the Gaussian distribution in $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. The parameters of the two approximate Gaussian densities can be found through the use of linearization, leading to the extended KF (EKF) [6], [7] and posterior linearization filters [8]–[10]. Numerical integration-based methods can also be applied, resulting in the unscented KF (UKF) [11], Gaussian-Hermite KF (GHKF) [12]–[14], cubature KF (CKF) [15], [16] and Gaussian-Hermite quadrature filter (GHQF) [17]. Unlike the above GFs, the variational GF (VGF) achieves Gaussian filtering with improved performance from an optimization perspective. It employs the variational inference [18]–[20], and finds the approximate state posterior normally through minimizing the forward Kullback-Leibler divergence (KLD) between it and the true posterior (1b). The use of the backward KLD minimization has also been considered in [21], [22].

Existing VGFs adopt various iterative algorithms to realize KLD minimization. Gradient descent was used in [21], [23]–[28] to compute the posterior mean $\boldsymbol{\mu}_t$ and covariance $\boldsymbol{\Sigma}_t$. These work assumed Gaussian measurement noise and required extra pre-conditioning [21], [23], [24], [28], dimensionality expansion [25] or approximate Hessian [27] for enhancing numerical stability. [29] presented a linearized alternating direction method of multipliers (LADMM) algorithm for estimating $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$. It still needs careful selection of

penalty parameters though.

Methods that utilizes natural gradient descent (NGD) [30]–[33] are popular as well, probably because the NGD introduces pre-conditioning in a principled way. Specifically, [34] derived an approximate NGD rule that iteratively refines the estimates of $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$. [35] gave the NGD-based rule for identifying the posterior mean $\boldsymbol{\mu}_t$ and precision matrix $\mathbf{S}_t$, which is the inverse of the posterior covariance $\boldsymbol{\Sigma}_t$ (i.e., $\mathbf{S}_t = \boldsymbol{\Sigma}_t^{-1}$). The development of these methods again assumed a measurement model with Gaussian noise. Noting that the Gaussian distribution belongs to the exponential family [36], we established the NGD update rule in the natural parameter space [37]. The obtained algorithm is equivalent to the conjugate-computation variational inference (CVI) proposed in [38]. Besides, it is a generalized version of the Bayesian online natural gradient (BONG) method [39] that executes one NGD iteration only.

The purpose of this paper is twofold. We first re-examine deriving using matrix calculus the exact NGD update rules under two parameterizations of the Gaussian posterior required for realizing VGFs. One parameterization contains the posterior mean $\boldsymbol{\mu}_t$ and covariance $\boldsymbol{\Sigma}_t$, and the other consists of $\boldsymbol{\mu}_t$ and the posterior precision matrix $\mathbf{S}_t$. Different from [34], [35], the derivation here is general enough so that the results are applicable to non-Gaussian measurement models. We compare the developed algorithms with those scattered in literature such as [35], [37], [39]–[42]. Important new insights into their subtle but inadequately explored difference are gained. Besides, this paper presents an alternative way for finding the modifications needed by the obtained NGD rules to guarantee that the posterior covariance $\boldsymbol{\Sigma}_t$ and precision matrix $\mathbf{S}_t$ remain positive definite during the iterative update [40]. Finally, we integrate the established NGD rules into the VGF proposed in [37] and apply them to a range-bearing tracking task for simulation-based performance evaluation.

The rest of this paper is organized as follows. Section II formulates the KLD minimization problem under the considered parameterizations of the Gaussian posterior, and presents useful matrix calculus results. Section III derives the desired NGD update rules and compares them with existing methods. Section IV gives the modified rules with positive definiteness guarantee for the covariance/precision matrix. Simulation results are given in Section V. Section VI concludes the paper.

II. Problem Formulation and Preliminaries

A. Problem Formulation

The theoretical development starts with assuming that the state predictive distribution at time t, which is $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ given in (1a), has been approximated as a Gaussian density $\mathcal{N}(\mathbf{x}_t; \mathbf{m}_t, \mathbf{P}_t)$. The parameters $\mathbf{m}_t$ and $\mathbf{P}_t$ can be obtained via e.g., moment matching [35], [43] (see Appendix I). Under this assumption, VGFs aim at finding another Gaussian $q(\mathbf{x}_t)$ with the smallest forward KLD between itself and the 'true' state posterior $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ to achieve variational Gaussian filtering.

We are interested in establishing the iterative NGD-based update rules for computing the desired approximate Gaussian posterior under two parameterizations $q(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ and $q(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t, \mathbf{S}_t^{-1})$. The loss function is [44]

$$\begin{aligned}
&\text{KLD}(q(\mathbf{x}_t)||p(\mathbf{x}_t|\mathbf{z}_{1:t})) = \int q(\mathbf{x}_t)\text{log}\frac{q(\mathbf{x}_t)}{p(\mathbf{x}_t|\mathbf{z}_{1:t})}\text{d}\mathbf{x}_t \\
&\propto \int q(\mathbf{x}_t)\text{log}\frac{q(\mathbf{x}_t)}{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1})}\text{d}\mathbf{x}_t \\
&\approx -E_{q(\mathbf{x}_t)}[\text{log}p(\mathbf{z}_t|\mathbf{x}_t)] + \text{KLD}(q(\mathbf{x}_t)||\mathcal{N}(\mathbf{x}_t; \mathbf{m}_t, \mathbf{P}_t)),
\end{aligned} \tag{2}$$

where we have substituted (1b), neglected the normalization factor $p(\mathbf{z}_t|\mathbf{z}_{1:t-1})$ and replaced the state predictive distribution with its Gaussian approximation. Note that minimizing (2) to find the approximate posterior $q(\mathbf{x}_t)$ can be considered an instance of the generalized variational inference [45].

B. Mathematical Preliminaries

We present some useful matrix calculus results to facilitate the algorithm development in the following sections. Let $\mathbf{A}$ be a $N \times N$ matrix with the partitioned form $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N]$, where $\mathbf{a}_j$, $j = 1, 2, ..., N$, is the j-th column of $\mathbf{A}$. $\mathbf{A}_{ij}$ denotes the element in the i-th row and j-th column of $\mathbf{A}$. $\text{vec}(\mathbf{A})$ is the column-vectorised version of $\mathbf{A}$, which is

$$\text{vec}(\mathbf{A}) = \left[\mathbf{a}_1^T, \mathbf{a}_2^T, ..., \mathbf{a}_N^T\right]^T. \tag{3}$$

It can be seen that $\text{vec}(\mathbf{A})$ is a $N^2 \times 1$ column vector with its $(i + (j-1)N)$-th element being equal to $\mathbf{A}_{ij}$. Suppose $\mathbf{B}$ is a $N \times N$ matrix as well. We can show that

$$\text{vec}(\mathbf{A})^T\text{vec}(\mathbf{B}) = \begin{cases} \text{tr}(\mathbf{AB}), & \text{if } \mathbf{A}^T = \mathbf{A} \\ \text{tr}(\mathbf{A}^T\mathbf{B}), & \text{otherwise} \end{cases}, \tag{4}$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix.

We next evaluate the partial derivative $\frac{\partial \text{vec}(\mathbf{A}^{-1})}{\partial \text{vec}(\mathbf{A})^T}$ under the assumption that $\mathbf{A}$ is symmetric and invertible. We note from (3) that it can be expressed in the partitioned form $\frac{\partial \text{vec}(\mathbf{A}^{-1})}{\partial \text{vec}(\mathbf{A})^T} = \left[\frac{\partial \text{vec}(\mathbf{A}^{-1})}{\partial \mathbf{a}_1^T}, \frac{\partial \text{vec}(\mathbf{A}^{-1})}{\partial \mathbf{a}_2^T}, ..., \frac{\partial \text{vec}(\mathbf{A}^{-1})}{\partial \mathbf{a}_N^T}\right]$, where the j-th block is

$$\frac{\partial \text{vec}(\mathbf{A}^{-1})}{\partial \mathbf{a}_j^T} = \left[\text{vec}\left(\frac{\partial \mathbf{A}^{-1}}{\partial \mathbf{A}_{1j}}\right), ..., \text{vec}\left(\frac{\partial \mathbf{A}^{-1}}{\partial \mathbf{A}_{Nj}}\right)\right]. \tag{5}$$

Applying (59) in Chapter 2 of [46], we have that for $i = 1, 2, ..., N$,

$$\text{vec}\left(\frac{\partial \mathbf{A}^{-1}}{\partial \mathbf{A}_{ij}}\right) = -\text{vec}\left(\mathbf{A}^{-1}\mathbf{e}_i\mathbf{e}_j^T\mathbf{A}^{-1}\right) = -\mathbf{A}^{-1}\mathbf{e}_j \otimes \mathbf{A}^{-1}\mathbf{e}_i, \tag{6}$$

where $\mathbf{e}_i$ is an one-shot vector with the i-th element being 1 and other being 0s, $\otimes$ denotes the Kronecker product, and $\mathbf{A}$ being symmetric has been used to obtain the second equality. Putting (6) into (5) yields

$$\begin{aligned}
\frac{\partial \text{vec}(\mathbf{A}^{-1})}{\partial \mathbf{a}_j^T} &= \left[-\mathbf{A}^{-1}\mathbf{e}_j \otimes \mathbf{A}^{-1}\mathbf{e}_1, ..., -\mathbf{A}^{-1}\mathbf{e}_j \otimes \mathbf{A}^{-1}\mathbf{e}_N\right] \\
&= -\mathbf{A}^{-1}\mathbf{e}_j \otimes \mathbf{A}^{-1}.
\end{aligned} \tag{7}$$

Applying (7) and following a similar argument, we obtain that

$$\frac{\partial \text{vec}(\mathbf{A}^{-1})}{\partial \text{vec}(\mathbf{A})^T} = -\mathbf{A}^{-1} \otimes \mathbf{A}^{-1}. \tag{8}$$

We next show that if $\mathbf{A}$ is symmetric and invertible,

$$\frac{\partial \text{vec}(\mathbf{A}^{-1})}{\partial \text{vec}(\mathbf{A})^T} \text{vec}(\mathbf{B}) = -\text{vec}(\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}). \tag{9}$$

The proof begins with noting that the $(i+(j-1)N)$-th element of the column vector on the left hand side is equal to

$$\begin{aligned}\left(\frac{\partial \mathbf{A}^{-1}_{ij}}{\partial \text{vec}(\mathbf{A})}\right)^T \text{vec}(\mathbf{B}) &= \text{vec}\left(\frac{\partial \mathbf{e}_i^T\mathbf{A}^{-1}\mathbf{e}_j}{\partial \mathbf{A}}\right)^T \text{vec}(\mathbf{B}) \\ &= -\text{vec}(\mathbf{A}^{-1}\mathbf{e}_i\mathbf{e}_j^T\mathbf{A}^{-1})^T\text{vec}(\mathbf{B}) \\ &= -\text{tr}(\mathbf{A}^{-1}\mathbf{e}_j\mathbf{e}_i^T\mathbf{A}^{-1}\mathbf{B}) \\ &= -\mathbf{e}_i^T\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\mathbf{e}_j,\end{aligned} \tag{10}$$

where (61) in Chapter 2 of [46], (4) and $\mathbf{A}$ being symmetric have been used to arrive at the second and third equalities. (10) reveals that the $(i+(j-1)N)$-th element is the same as the element in the i-th row and j-th column of $\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}$. This proves (9).

Given that $\mathbf{A}$ is symmetric, we are going to show that

$$\frac{\partial \text{vec}(\mathbf{A})^T(\mathbf{B}\otimes\mathbf{B})\text{vec}(\mathbf{A})}{\partial \text{vec}(\mathbf{B})} = 2\text{vec}(\mathbf{A}\mathbf{B}\mathbf{A}). \tag{11}$$

In particular, the $(i+(j-1)N)$-th element of the column vector on the left hand side of (11) is

$$\begin{aligned}&\frac{\partial \text{vec}(\mathbf{A})^T(\mathbf{B}\otimes\mathbf{B})\text{vec}(\mathbf{A})}{\partial \mathbf{B}_{ij}} \\ &= \text{vec}(\mathbf{A})^T\left(\mathbf{e}_i\mathbf{e}_j^T\otimes\mathbf{B} + \mathbf{B}\otimes\mathbf{e}_i\mathbf{e}_j^T\right)\text{vec}(\mathbf{A}).\end{aligned} \tag{12}$$

Utilizing (3), we have that

$$\text{vec}(\mathbf{A})^T(\mathbf{e}_i\mathbf{e}_j^T\otimes\mathbf{B})\text{vec}(\mathbf{A}) = \mathbf{e}_i^T\mathbf{A}\mathbf{B}\mathbf{A}\mathbf{e}_j, \tag{13}$$

where the symmetry of $\mathbf{A}$ has been exploited to arrive at the second equality. Moreover, the remaining quadratic term on the right hand side of (12) can be written as

$$\begin{aligned}\text{vec}(\mathbf{A})^T\left(\mathbf{B}\otimes\mathbf{e}_i\mathbf{e}_j^T\right)\text{vec}(\mathbf{A}) &= \sum_{n=1}^N\sum_{m=1}^N \mathbf{B}_{nm}\mathbf{A}_{in}\mathbf{A}_{jm} \\ &= \mathbf{e}_i^T\mathbf{A}\mathbf{B}\mathbf{A}\mathbf{e}_j,\end{aligned} \tag{14}$$

where $\mathbf{A}$ being symmetric such that $\mathbf{A}_{jm} = \mathbf{A}_{mj}$ has been applied when deriving the second equality. Combining (12)-(14) gives that the $(i+(j-1)N)$-th element of $\frac{\partial \text{vec}(\mathbf{A})^T(\mathbf{B}\otimes\mathbf{B})\text{vec}(\mathbf{A})}{\partial \text{vec}(\mathbf{B})}$ is equal to the element in the i-th row and j-th column of the matrix product $2\mathbf{A}\mathbf{B}\mathbf{A}$. This completes the proof of (11).

III. NGD-based Measurement Update

This section utilizes the results in Section II.B and establishes the measurement update rule relying on NGD to find the Gaussian posterior $q(\mathbf{x}_t)$ that minimizes the forward KLD given in (2). Two parameterizations of $q(\mathbf{x}_t)$, $q(\mathbf{x}_t) = N(\mathbf{x}_t;\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)$ and $q(\mathbf{x}_t) = N(\mathbf{x}_t;\boldsymbol{\mu}_t,\mathbf{S}_t^{-1})$, are considered. Their difference is that with the second parameterization, the posterior precision matrix $\mathbf{S}_t$, rather than the posterior covariance $\boldsymbol{\Sigma}_t$, is to be estimated. The obtained update rules are compared with each other and with the one developed using the natural parameterization of $q(\mathbf{x}_t)$ [37].

We introduce two unknown vectors $\boldsymbol{\theta}_1 = [\boldsymbol{\mu}_t^T, \text{vec}(\boldsymbol{\Sigma}_t)^T]^T$ and $\boldsymbol{\theta}_2 = [\boldsymbol{\mu}_t^T, \text{vec}(\mathbf{S}_t)^T]^T$ to represent the two parameterizations. Putting the analytical expression for the last term in (2), which is the KLD between two Gaussian densities, yields the loss functions with respect to $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. They are given by

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}_1) \propto &- E_{q(\mathbf{x}_t)}[\log p(\mathbf{z}_t|\mathbf{x}_t)] - \frac{1}{2}\log|\boldsymbol{\Sigma}_t| \\ &+ \frac{1}{2}\text{tr}\left(\mathbf{P}_t^{-1}(\boldsymbol{\Sigma}_t + (\boldsymbol{\mu}_t - \mathbf{m}_t)(\boldsymbol{\mu}_t - \mathbf{m}_t)^T)\right),\end{aligned} \tag{15a}$$

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}_2) \propto &- E_{q(\mathbf{x}_t)}[\log p(\mathbf{z}_t|\mathbf{x}_t)] + \frac{1}{2}\log|\mathbf{S}_t| \\ &+ \frac{1}{2}\text{tr}\left(\mathbf{P}_t^{-1}(\mathbf{S}_t^{-1} + (\boldsymbol{\mu}_t - \mathbf{m}_t)(\boldsymbol{\mu}_t - \mathbf{m}_t)^T)\right).\end{aligned} \tag{15b}$$

With NGD, the Fisher information matrix (FIM) of the unknowns are incorporated when we update their estimates to produce principled pre-conditioning and improve convergence speed [30]–[33]. Mathematically, the desired iterative NGD-based update for minimizing (15), which is an instance of the Bayesian learning rule (BLR) [47], can be expressed as

$$\boldsymbol{\theta}_{i,k+1} = \boldsymbol{\theta}_{i,k} - \alpha_k \cdot \text{FIM}(\boldsymbol{\theta}_{i,k})^{-1} \cdot \frac{\partial \mathcal{L}(\boldsymbol{\theta}_{i,k})}{\partial \boldsymbol{\theta}_{i,k}}, \tag{16}$$

where $\boldsymbol{\theta}_{i,k}$ is the estimate of $\boldsymbol{\theta}_i$, $i = 1, 2$, in the k-th iteration, and α_k is the step size. The desired update rule in (16) requires evaluating the FIM and the gradient $\partial\mathcal{L}(\boldsymbol{\theta}_{i,k})/\partial\boldsymbol{\theta}_{i,k}$.

A. NGD-based Update Rule for $q(\mathbf{x}_t) = N(\mathbf{x}_t;\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)$

In this case, the unknown vector is $\boldsymbol{\theta}_1 = [\boldsymbol{\mu}_t^T, \text{vec}(\boldsymbol{\Sigma}_t)^T]^T$. Its FIM can be written in the following generic form [48]

$$\text{FIM}(\boldsymbol{\theta}_1) = -E_{q(\mathbf{x}_t)}\begin{bmatrix}\frac{\partial^2 \log q(\mathbf{x}_t)}{\partial\boldsymbol{\mu}_t\partial\boldsymbol{\mu}_t^T} & \frac{\partial^2 \log q(\mathbf{x}_t)}{\partial\boldsymbol{\mu}_t\partial\text{vec}(\boldsymbol{\Sigma}_t)^T} \\ \frac{\partial^2 \log q(\mathbf{x}_t)}{\partial\text{vec}(\boldsymbol{\Sigma}_t)\partial\boldsymbol{\mu}_t^T} & \frac{\partial^2 \log q(\mathbf{x}_t)}{\partial\text{vec}(\boldsymbol{\Sigma}_t)\partial\text{vec}(\boldsymbol{\Sigma}_t)^T}\end{bmatrix}. \tag{17}$$

Here, $\log q(\mathbf{x}_t) \propto -\frac{1}{2}\log|\boldsymbol{\Sigma}_t| - \frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_t)^T\boldsymbol{\Sigma}_t^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_t)$. It is straightforward to show that

$$\frac{\partial^2 \log q(\mathbf{x}_t)}{\partial\boldsymbol{\mu}_t\partial\boldsymbol{\mu}_t^T} = -\boldsymbol{\Sigma}_t^{-1}. \tag{18}$$

Besides, we have

$$E_{q(\mathbf{x}_t)}\left[\frac{\partial^2 \log q(\mathbf{x}_t)}{\partial\boldsymbol{\mu}_t\partial\text{vec}(\boldsymbol{\Sigma}_t)^T}\right] = E_{q(\mathbf{x}_t)}\left[\frac{\partial\boldsymbol{\Sigma}_t^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_t)}{\partial\text{vec}(\boldsymbol{\Sigma}_t)^T}\right] = \mathbf{O}. \tag{19}$$

The second equality comes from that the term $\boldsymbol{\Sigma}_t^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_t)$ is affine in $(\mathbf{x}_t - \boldsymbol{\mu}_t)$ and as such, the expectation with respect to $q(\mathbf{x}_t) = N(\mathbf{x}_t;\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)$ is zero. Lastly, the lower-right block of $\text{FIM}(\boldsymbol{\theta}_1)$ in (17) is, after being multiplied by 2,

$$\begin{aligned}E_{q(\mathbf{x}_t)}\left[\frac{\partial\text{vec}\left(\frac{\partial\log|\boldsymbol{\Sigma}_t| + (\mathbf{x}_t-\boldsymbol{\mu}_t)^T\boldsymbol{\Sigma}_t^{-1}(\mathbf{x}_t-\boldsymbol{\mu}_t)}{\partial\boldsymbol{\Sigma}_t}\right)}{\partial\text{vec}(\boldsymbol{\Sigma}_t)^T}\right] &= \frac{\partial\text{vec}(\boldsymbol{\Sigma}_t^{-1})}{\partial\text{vec}(\boldsymbol{\Sigma}_t)^T} \\ - E_{q(\mathbf{x}_t)}\left[\frac{\partial\text{vec}(\boldsymbol{\Sigma}_t^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_t)(\mathbf{x}_t - \boldsymbol{\mu}_t)^T\boldsymbol{\Sigma}_t^{-1})}{\partial\text{vec}(\boldsymbol{\Sigma}_t)^T}\right],\end{aligned} \tag{20}$$

where (57) and (61) in Chapter 2 of [46] have been utilized.

To evaluate the last term in (20), we assume without the loss of generality that the posterior covariance $\boldsymbol{\Sigma}_t$ is a $N \times N$ matrix. The $(i + (j-1)N)$-th column of the last term in (20) can then be shown to be $-\text{vec}(\boldsymbol{\Sigma}_t^{-1}\mathbf{e}_i\mathbf{e}_j^T\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\Sigma}_t\boldsymbol{\Sigma}_t^{-1} + \boldsymbol{\Sigma}_t^{-1}\boldsymbol{\Sigma}_t\boldsymbol{\Sigma}_t^{-1}\mathbf{e}_i\mathbf{e}_j^T\boldsymbol{\Sigma}_t^{-1}) = -2\text{vec}(\boldsymbol{\Sigma}_t^{-1}\mathbf{e}_i\mathbf{e}_j^T\boldsymbol{\Sigma}_t^{-1})$, where (6) and $E_{q(\mathbf{x}_t)}[(\mathbf{x}_t - \boldsymbol{\mu}_t)(\mathbf{x}_t - \boldsymbol{\mu}_t)^T] = \boldsymbol{\Sigma}_t$ have been used. Applying (6) again reveals that this is exactly the $(i+(j-1)N)$-th column of $-2\frac{\partial \text{vec}(\boldsymbol{\Sigma}_t^{-1})}{\partial \text{vec}(\boldsymbol{\Sigma}_t)^T}$. Putting the above result into (20) and dividing both sides by 2 yield

$$-E_{q(\mathbf{x}_t)}\left[\frac{\partial^2 \text{log}q(\mathbf{x}_t)}{\partial \text{vec}(\boldsymbol{\Sigma}_t)\partial \text{vec}(\boldsymbol{\Sigma}_t)^T}\right] = -\frac{1}{2}\frac{\partial \text{vec}(\boldsymbol{\Sigma}_t^{-1})}{\partial \text{vec}(\boldsymbol{\Sigma}_t)^T}. \quad (21)$$

Substituting (18)-(21) into (17) and using (8), we obtain

$$\begin{aligned}\text{FIM}(\boldsymbol{\theta}_{1,k}) &= \begin{bmatrix}\boldsymbol{\Sigma}_{t,k}^{-1} & \mathbf{O} \\ \mathbf{O} & -\frac{1}{2}\frac{\partial \text{vec}(\boldsymbol{\Sigma}_{t,k}^{-1})}{\partial \text{vec}(\boldsymbol{\Sigma}_{t,k})^T}\end{bmatrix} \\ &= \begin{bmatrix}\boldsymbol{\Sigma}_{t,k}^{-1} & \mathbf{O} \\ \mathbf{O} & \frac{1}{2}\boldsymbol{\Sigma}_{t,k}^{-1} \otimes \boldsymbol{\Sigma}_{t,k}^{-1}\end{bmatrix}.\end{aligned} \quad (22)$$

Under $\boldsymbol{\theta}_1$, the gradient in (16) becomes

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}_{1,k})}{\partial \boldsymbol{\theta}_{1,k}} = \left[\left(\frac{\partial \mathcal{L}(\boldsymbol{\theta}_{1,k})}{\partial \boldsymbol{\mu}_{t,k}}\right)^T, \left(\frac{\partial \mathcal{L}(\boldsymbol{\theta}_{1,k})}{\partial \text{vec}(\boldsymbol{\Sigma}_{t,k})}\right)^T\right]^T. \quad (23)$$

By utilizing (15a) as well as the Bonnet's theorem and Price's theorem [49], [50], we have

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}_{1,k})}{\partial \boldsymbol{\mu}_{t,k}} = \mathbf{P}_t^{-1}(\boldsymbol{\mu}_{t,k} - \mathbf{m}_t) - E_{q(\mathbf{x}_t)}\left[\frac{\partial \text{log}p(\mathbf{z}_t|\mathbf{x}_t)}{\partial \mathbf{x}_t}\right]\Bigg|_{\boldsymbol{\theta}_{1,k}}, \quad (24a)$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}_{1,k})}{\partial \boldsymbol{\Sigma}_{t,k}} = \frac{1}{2}\Big(\mathbf{P}_t^{-1} - \boldsymbol{\Sigma}_{t,k}^{-1} - E_{q(\mathbf{x}_t)}\left[\frac{\partial^2 \text{log}p(\mathbf{z}_t|\mathbf{x}_t)}{\partial \mathbf{x}_t \partial \mathbf{x}_t^T}\right]\Bigg|_{\boldsymbol{\theta}_{1,k}}\Big). \quad (24b)$$

Here, $\boldsymbol{\mu}_{t,k}$ and $\boldsymbol{\Sigma}_{t,k}$ are the estimates of the posterior mean $\boldsymbol{\mu}_t$ and covariance $\boldsymbol{\Sigma}_t$ in the k-th iteration. Note that the second block on the right hand side of (23) can in fact be found using (24b) and $\partial \mathcal{L}(\boldsymbol{\theta}_{1,k})/\partial \text{vec}(\boldsymbol{\Sigma}_{t,k}) = \text{vec}\left(\partial \mathcal{L}(\boldsymbol{\theta}_{1,k})/\partial \boldsymbol{\Sigma}_{t,k}\right)$.

Substituting (24) into (23), and putting the result together with (22) back into (16) yield the NGD-based update rule for the approximate Gaussian posterior $N(\mathbf{x}_t; \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, which is

$$\begin{aligned}&\boldsymbol{\mu}_{t,k+1} = \boldsymbol{\mu}_{t,k} \\ &- \alpha_k \boldsymbol{\Sigma}_{t,k}\Big(\mathbf{P}_t^{-1}(\boldsymbol{\mu}_{t,k} - \mathbf{m}_t) - E_{q(\mathbf{x}_t)}\left[\frac{\partial \text{log}p(\mathbf{z}_t|\mathbf{x}_t)}{\partial \mathbf{x}_t}\right]\Bigg|_{\boldsymbol{\theta}_{1,k}}\Big),\end{aligned} \quad (25)$$

$$\begin{aligned}\text{vec}(\boldsymbol{\Sigma}_{t,k+1}) &= \text{vec}(\boldsymbol{\Sigma}_{t,k}) + 2\alpha_k \frac{\partial \text{vec}(\boldsymbol{\Sigma}_{t,k})}{\partial \text{vec}(\boldsymbol{\Sigma}_{t,k}^{-1})^T}\frac{\partial \mathcal{L}(\boldsymbol{\theta}_{1,k})}{\partial \text{vec}(\boldsymbol{\Sigma}_{t,k})} \\ &= \text{vec}(\boldsymbol{\Sigma}_{t,k}) - 2\alpha_k \text{vec}\left(\boldsymbol{\Sigma}_{t,k}\frac{\partial \mathcal{L}(\boldsymbol{\theta}_{1,k})}{\partial \boldsymbol{\Sigma}_{t,k}}\boldsymbol{\Sigma}_{t,k}\right),\end{aligned} \quad (26)$$

where (9) has been used to arrive at the second equality of (26). Noting that the column-vectorisation operator on both sides of (26) can be safely removed and using (24b) leads to the simplifed rule for updating the posterior covariance:

$$\begin{aligned}&\boldsymbol{\Sigma}_{t,k+1} = \boldsymbol{\Sigma}_{t,k} \\ &- \alpha_k \boldsymbol{\Sigma}_{t,k}\Big(\mathbf{P}_t^{-1} - \boldsymbol{\Sigma}_{t,k}^{-1} - E_{q(\mathbf{x}_t)}\left[\frac{\partial^2 \text{log}p(\mathbf{z}_t|\mathbf{x}_t)}{\partial \mathbf{x}_t \partial \mathbf{x}_t^T}\right]\Bigg|_{\boldsymbol{\theta}_{1,k}}\Big)\boldsymbol{\Sigma}_{t,k}.\end{aligned} \quad (27)$$

B. NGD-based Update Rule for $q(\mathbf{x}_t) = N(\mathbf{x}_t; \boldsymbol{\mu}_t, \mathbf{S}_t^{-1})$

Under this parameterization, the unknowns are collected in $\boldsymbol{\theta}_2 = [\boldsymbol{\mu}_t^T, \text{vec}(\mathbf{S}_t)^T]^T$. The FIM of $\boldsymbol{\theta}_2$ has the same functional form as FIM($\boldsymbol{\theta}_1$) given in (17), except that vec($\boldsymbol{\Sigma}_t$) needs to be replaced with vec($\mathbf{S}_t$) and now, $\text{log}q(\mathbf{x}_t) \propto \frac{1}{2}\text{log}|\mathbf{S}_t| - \frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_t)^T\mathbf{S}_t(\mathbf{x}_t - \boldsymbol{\mu}_t)$. We can show that (18) still holds but the right hand side should be $-\mathbf{S}_t$. Meanwhile, we have

$$E_{q(\mathbf{x}_t)}\left[\frac{\partial^2 \text{log}q(\mathbf{x}_t)}{\partial \boldsymbol{\mu}_t \partial \text{vec}(\mathbf{S}_t)^T}\right] = E_{q(\mathbf{x}_t)}\left[\frac{\partial \mathbf{S}_t(\mathbf{x}_t - \boldsymbol{\mu}_t)}{\partial \text{vec}(\mathbf{S}_t)^T}\right] = \mathbf{O}, \quad (28)$$

again since the expectation of $(\mathbf{x}_t - \boldsymbol{\mu}_t)$ is zero, similar to the derivation of (19). The lower-right block of FIM($\boldsymbol{\theta}_2$) is

$$-E_{q(\mathbf{x}_t)}\left[\frac{\partial^2 \text{log}q(\mathbf{x}_t)}{\partial \text{vec}(\mathbf{S}_t)\partial \text{vec}(\mathbf{S}_t)^T}\right] = -\frac{1}{2}\frac{\partial \text{vec}(\mathbf{S}_t^{-1})}{\partial \text{vec}(\mathbf{S}_t)^T}. \quad (29)$$

Combining these results gives

$$\begin{aligned}\text{FIM}(\boldsymbol{\theta}_{2,k}) &= \begin{bmatrix}\mathbf{S}_{t,k} & \mathbf{O} \\ \mathbf{O} & -\frac{1}{2}\frac{\partial \text{vec}(\mathbf{S}_{t,k}^{-1})}{\partial \text{vec}(\mathbf{S}_{t,k})^T}\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{S}_{t,k} & \mathbf{O} \\ \mathbf{O} & \frac{1}{2}\mathbf{S}_{t,k}^{-1} \otimes \mathbf{S}_{t,k}^{-1}\end{bmatrix},\end{aligned} \quad (30)$$

where (8) has been applied to obtain the second equality. $\mathbf{S}_{t,k}$ is the estimated posterior precision matrix in the k-th iteration.

The gradient $\frac{\partial \mathcal{L}(\boldsymbol{\theta}_{2,k})}{\partial \boldsymbol{\theta}_{2,k}}$ can be expressed as

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}_{2,k})}{\partial \boldsymbol{\theta}_{2,k}} = \left[\left(\frac{\partial \mathcal{L}(\boldsymbol{\theta}_{2,k})}{\partial \boldsymbol{\mu}_{t,k}}\right)^T, \left(\frac{\partial \mathcal{L}(\boldsymbol{\theta}_{2,k})}{\partial \text{vec}(\mathbf{S}_{t,k})}\right)^T\right]^T. \quad (31)$$

By substituting (15b), the partial derivative $\frac{\partial \mathcal{L}(\boldsymbol{\theta}_{2,k})}{\partial \boldsymbol{\mu}_{t,k}}$ can be shown to be equal to $\frac{\partial \mathcal{L}(\boldsymbol{\theta}_{1,k})}{\partial \boldsymbol{\mu}_{t,k}}$ given in (24a) with $\boldsymbol{\theta}_{1,k}$ being replaced by $\boldsymbol{\theta}_{2,k}$. Putting the above result together with (30) into (16) produces the NGD-based update rule for the parameterization $q(\mathbf{x}_t) = N(\mathbf{x}_t; \boldsymbol{\mu}_t, \mathbf{S}_t^{-1})$, which is

$$\begin{aligned}&\boldsymbol{\mu}_{t,k+1} = \boldsymbol{\mu}_{t,k} \\ &- \alpha_k \mathbf{S}_{t,k}^{-1}\Big(\mathbf{P}_t^{-1}(\boldsymbol{\mu}_{t,k} - \mathbf{m}_t) - E_{q(\mathbf{x}_t)}\left[\frac{\partial \text{log}p(\mathbf{z}_t|\mathbf{x}_t)}{\partial \mathbf{x}_t}\right]\Bigg|_{\boldsymbol{\theta}_{2,k}}\Big),\end{aligned} \quad (32)$$

$$\begin{aligned}\text{vec}(\mathbf{S}_{t,k+1}) &= \text{vec}(\mathbf{S}_{t,k}) + 2\alpha_k \frac{\partial \text{vec}(\mathbf{S}_{t,k})}{\partial \text{vec}(\mathbf{S}_{t,k}^{-1})^T}\frac{\partial \mathcal{L}(\boldsymbol{\theta}_{2,k})}{\partial \text{vec}(\mathbf{S}_{t,k})} \\ &= \text{vec}(\mathbf{S}_{t,k}) + 2\alpha_k \text{vec}\left(\frac{\partial \mathcal{L}(\boldsymbol{\theta}_{2,k})}{\partial \mathbf{S}_{t,k}^{-1}}\right).\end{aligned} \quad (33)$$

Applying again (15b) and the Price's theorem [49], [50], and removing the column-vectorisation operator from both sides

of (33) yield the update rule for estimating iteratively the posterior precision matrix $\mathbf{S}_t$:

$$\mathbf{S}_{t,k+1} = \mathbf{S}_{t,k} - \alpha_k\big(\mathbf{S}_{t,k} - \mathbf{P}_t^{-1} + E_{q(\mathbf{x}_t)}\left[\frac{\partial^2 \log p(\mathbf{x}_t|\mathbf{x}_t)}{\partial \mathbf{x}_t \partial \mathbf{x}_t^T}\right]\Big|_{\boldsymbol{\theta}_{2,k}}\big). \quad (34)$$

C. Discussions

Several important observations can be obtained. First, under the two parameterizations of the approximate Gaussian posterior $q(\mathbf{x}_t)$, the FIMs of their unknown parameters are both block diagonal (see (22) and (30)). The two parameterizatons are thus block coordinate (BC) parameterizations [40].

Second, from (25) and (32), we can see that the update rules for the posterior mean $\boldsymbol{\mu}_t$ under the two parameterizations in consideration are indeed the same. This is somewhat expected, as the parameterizations differ only in how they represent the posterior covariance, and they are also BC parameterizations.

Third, the stationary points of the update rules under the two parameterizations are identical. From (25), (27) and (34), we have that they must satisfy $\boldsymbol{\mu}_t = \mathbf{m}_t + \mathbf{P}_t E_{q(\mathbf{x}_t)}\left[\frac{\partial \log p(\mathbf{z}_t|\mathbf{x}_t)}{\partial \mathbf{x}_t}\right]$ and $\mathbf{S}_t = \boldsymbol{\Sigma}_t^{-1} = \mathbf{P}_t^{-1} - E_{q(\mathbf{x}_t)}\left[\frac{\partial^2 \log p(\mathbf{z}_t|\mathbf{x}_t)}{\partial \mathbf{x}_t \partial \mathbf{x}_t^T}\right]$. For linear Gaussian measurements (i.e., $p(\mathbf{z}_t|\mathbf{x}_t) = N(\mathbf{z}_t; \mathbf{H}\mathbf{x}_t, \mathbf{R})$ with $\mathbf{H}$ and $\mathbf{R}$ being the measurement matrix and noise covariance), these conditions recover the KF. The developed update rules are therefore optimal for linear Gaussian systems.

Fourth, the update rules for the posterior covariance $\boldsymbol{\Sigma}_t$ in (27) and posterior precision matrix $\mathbf{S}_t$ in (34) are approximately equivalent if their correction terms are sufficiently small. For example, in (34), the correction term is

$$\mathbf{M}_{2,k} = -\alpha_k\big(\mathbf{S}_{t,k} - \mathbf{P}_t^{-1} + E_{q(\mathbf{x}_t)}\left[\frac{\partial^2 \log p(\mathbf{x}_t|\mathbf{x}_t)}{\partial \mathbf{x}_t \partial \mathbf{x}_t^T}\right]\Big|_{\boldsymbol{\theta}_{2,k}}\big). \quad (35)$$

Having a small step size a_k and/or the matrix in the parentheses being close to a zero matrix, which occurs when the iteration converges, would greatly decrease $\mathbf{M}_{2,k}$. In this case, inverting both sides of (34) and applying the approximation for small σ^2, $(\mathbf{A}+\sigma^2\mathbf{B})^{-1} \approx \mathbf{A}^{-1} - \sigma^2\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}$ (see (191) in Chapter of [46]), yields (27). Similarly, we can show that taking the matrix inverse on both sides of (27) under a small correction term leads to (34) as well.

Fifth, [37] developed the update rule based on the natural parameterization of $q(\mathbf{x}_t)$, which is, with $\boldsymbol{\lambda}_{t,k}$ representing the estimate of the natural parameter of $q(\mathbf{x}_t)$ in the k-th iteration,

$$\mathbf{S}_{t,k+1} = (1-\alpha_k)\mathbf{S}_{t,k} + \alpha_k\mathbf{P}_t^{-1} - \alpha_k E_{q_{\boldsymbol{\lambda}_{t,k}}(\mathbf{x}_t)}\left[\frac{\partial^2 \log(p(\mathbf{z}_t|\mathbf{x}_t))}{\partial \mathbf{x}_t \partial \mathbf{x}_t^T}\right], \quad (36)$$

$$\boldsymbol{\mu}_{t,k+1} = \boldsymbol{\mu}_{t,k} - \alpha_k\mathbf{S}_{t,k+1}^{-1}\cdot\mathbf{P}_t^{-1}(\boldsymbol{\mu}_{t,k} - \mathbf{m}_t) + \alpha_k\mathbf{S}_{t,k+1}^{-1}\cdot E_{q_{\boldsymbol{\lambda}_{t,k}}(\mathbf{x}_t)}\left[\frac{\partial \log(p(\mathbf{z}_t|\mathbf{x}_t))}{\partial \mathbf{x}_t}\right]. \quad (37)$$

Comparing (36) with (34) indicates that using $q(\mathbf{x}_t) = N(\mathbf{x}_t; \boldsymbol{\mu}_t, \mathbf{S}_t^{-1})$ and the natural parameterization would result in the *same* update rule for the posterior precision matrix $\mathbf{S}_t$. However, contrasting (37) with (32) reveals that with natural parameterization, the pre-conditioning matrix for refining the posterior mean $\boldsymbol{\mu}_t$ is the *updated* posterior covariance, rather than the one from the previous iteration[1]. The impact of this subtle difference on the estimation accuracy and convergence speed will be investigated in Section V.

Lastly, the FIMs and natural gradients for the two parameterizations are functionally consistent with these from [40], [41]. But the derivations in this work do not utilize differentials as in [41]. Moreover, they are established explicitly for state filtering applications in a more detailed and unified manner than [40] using the Bonnet's theorem and Price's theorem. The main purpose is to facilitate comparison between themselves and against the results under natural parameterization, which was less explored in literature such as [35], [39]–[41].

IV. Positive Definiteness Guarantee

There is no guarantee that the update rules for the posterior covariance $\boldsymbol{\Sigma}_t$ in (27) and posterior precision matrix $\mathbf{S}_t$ in (34) can maintain the positive definiteness of their estimates. [40] showed that adding an extra term in (27) and (34) can correct this issue. With Einstein summation notation, the c-th element of the column-vectorised version of this extra term is equal to

$$-\frac{1}{2}\Gamma^c{}_{ab}g^a g^b = -\frac{1}{2}F^{cd}\Gamma_{d,ab}g^a g^b = -\frac{1}{2}\sum_d F^{cd}\sum_a\sum_b \Gamma_{d,ab}g^a g^b. \quad (38)$$

Here, g^a denotes the a-th element of the column-vectorised version of the correction term in (27) or (34). $\Gamma^c{}_{ab}$ and $\Gamma_{d,ab}$ are the Christoffel symbol of the 2nd kind and the 1st kind. In the context of NGD and BC natural (BCN) parameterizations such as the parameterization $q(\mathbf{x}_t) = N(\mathbf{x}_t; \boldsymbol{\mu}_t, \mathbf{S}_t^{-1})$, F^{cd} is the element in the c-th row and d-th column of the inverse of the FIM of $\mathbf{S}_t$, and $\Gamma_{d,ab}$ becomes (see Theorem 3 in [40])

$$\Gamma_{d,ab} = \frac{1}{2}\frac{\partial^3 A(\boldsymbol{\phi})}{\partial\phi_a\partial\phi_b\partial\phi_d}. \quad (39)$$

$A(\boldsymbol{\phi})$ is the log partition function [36], [51]. Under the BCN Gaussian posterior $q(\mathbf{x}_t) = N(\mathbf{x}_t; \boldsymbol{\mu}_t, \mathbf{S}_t^{-1})$, we have that $\boldsymbol{\phi} = \text{vec}(\mathbf{S}_t)$ and $A(\text{vec}(\mathbf{S}_t)) = -\frac{1}{2}\log|\mathbf{S}_t| + \frac{1}{2}\boldsymbol{\mu}_t^T\mathbf{S}_t\boldsymbol{\mu}_t$.

We first present an alternative way to derive the extra term that can ensure the positive definiteness of the estimated posterior precision matrix $\mathbf{S}_t$ in (34). For this purpose, $\boldsymbol{\phi}$ and $A(\boldsymbol{\phi})$ in (39) are replaced with $\text{vec}(\mathbf{S}_t)$ and $A(\text{vec}(\mathbf{S}_t))$. We substitute the transformed (39) into (38), and stack the results in the ascending order of c. The column-vectorised version of the desired extra term can be expressed in the matrix form as

$$-\frac{1}{4}\text{FIM}(\text{vec}(\mathbf{S}_{t,k}))^{-1}\frac{\partial \mathbf{g}_{2,k}^T \frac{\partial^2 A(\text{vec}(\mathbf{S}_{t,k}))}{\partial \text{vec}(\mathbf{S}_{t,k})\partial \text{vec}(\mathbf{S}_{t,k})^T}\mathbf{g}_{2,k}}{\partial \text{vec}(\mathbf{S}_{t,k})}, \quad (40)$$

where $\mathbf{g}_{2,k} = \text{vec}(\mathbf{M}_{2,k})$ and $\mathbf{M}_{2,k}$ is defined in (35).

[1]There is an error in (25) of [35], which considered the parameterization $q(\mathbf{x}_t) = N(\mathbf{x}_t; \boldsymbol{\mu}_t, \mathbf{S}_t^{-1})$ only. The updated posterior covariance was incorrectly used there as the pre-conditioner for refining the posterior mean.

Using (30) and putting the definition of $A(\text{vec}(\mathbf{S}_t))$ given under (39) convert (40) into

$$\begin{aligned}
&-\frac{1}{4}\left(\frac{\partial \text{vec}(\mathbf{S}_{t,k}^{-1})}{\partial \text{vec}(\mathbf{S}_{t,k})^T}\right)^{-1}\frac{\partial \mathbf{g}_{2,k}^T \frac{\partial^2 \log|\mathbf{S}_{t,k}|}{\partial \text{vec}(\mathbf{S}_{t,k})\partial \text{vec}(\mathbf{S}_{t,k})^T}\mathbf{g}_{2,k}}{\partial \text{vec}(\mathbf{S}_{t,k})}\\
&=-\frac{1}{4}\frac{\partial \mathbf{g}_{2,k}^T \frac{\partial \text{vec}(\mathbf{S}_{t,k}^{-1})}{\partial \text{vec}(\mathbf{S}_{t,k})^T}\mathbf{g}_{2,k}}{\partial \text{vec}(\mathbf{S}_{t,k}^{-1})}=\frac{1}{4}\frac{\partial \mathbf{g}_{2,k}^T \mathbf{S}_{t,k}^{-1}\otimes \mathbf{S}_{t,k}^{-1}\mathbf{g}_{2,k}}{\partial \text{vec}(\mathbf{S}_{t,k}^{-1})}\\
&=\frac{1}{2}\text{vec}(\mathbf{M}_{2,k}\mathbf{S}_{t,k}^{-1}\mathbf{M}_{2,k}).
\end{aligned} \tag{41}$$

Here, we have applied (11) to obtain the last equality. Dropping the column-vectorisation operator in (41) and incorporating the result into (34) yield the modified update rule for the posterior precision matrix $\mathbf{S}_t$, which is

$$\mathbf{S}_{t,k+1}=\mathbf{S}_{t,k}+\mathbf{M}_{2,k}+\frac{1}{2}\mathbf{M}_{2,k}\mathbf{S}_{t,k}^{-1}\mathbf{M}_{2,k}. \tag{42}$$

The right hand side can be re-written as $\frac{1}{2}\mathbf{S}_{t,k}(\mathbf{S}_{t,k}^{-1}+(\mathbf{S}_{t,k}^{-1}+\mathbf{S}_{t,k}^{-1}\mathbf{M}_{2,k}\mathbf{S}_{t,k}^{-1})\mathbf{S}_{t,k}(\mathbf{S}_{t,k}^{-1}+\mathbf{S}_{t,k}^{-1}\mathbf{M}_{2,k}\mathbf{S}_{t,k}^{-1}))\mathbf{S}_{t,k}$, the symmetry of which establishes the positive definiteness of $\mathbf{S}_{t,k+1}$.

The extra term for guaranteeing the estimate of the posterior covariance $\mathbf{\Sigma}_t$ in (27) being positive definite can be found by following the approach that leads to (41) and applying the chain rule $\frac{\partial f(\text{vec}(\mathbf{\Sigma}_t))}{\partial \text{vec}(\mathbf{\Sigma}_t)}=\frac{\partial \text{vec}(\mathbf{S}_t)}{\partial \text{vec}(\mathbf{\Sigma}_t)^T}\frac{\partial f(\text{vec}(\mathbf{\Sigma}_t))}{\partial \text{vec}(\mathbf{S}_t)}$. The column vectorised version of the desired extra term is given at the bottom of this page, where $\mathbf{g}_{1,k}=\text{vec}(\mathbf{M}_{1,k})$ is the column vectorised correction term in (27) and $\mathbf{M}_{1,k}$ is equal to

$$-\alpha_k\mathbf{\Sigma}_{t,k}\big(\mathbf{P}_t^{-1}-\mathbf{\Sigma}_{t,k}^{-1}-E_{q(\mathbf{x}_t)}\left[\frac{\partial^2 \log p(\mathbf{z}_t|\mathbf{x}_t)}{\partial \mathbf{x}_t\partial \mathbf{x}_t^T}\right]\Big|_{\boldsymbol{\theta}_{1,k}}\big)\mathbf{\Sigma}_{t,k}. \tag{44}$$

Applying (22), (41) and again (11) simplifies (43) into

$$\begin{aligned}
&-\frac{1}{4}\frac{\partial \mathbf{g}_{1,k}^T \frac{\partial \text{vec}(\mathbf{S}_{t,k})}{\partial \text{vec}(\mathbf{\Sigma}_{t,k})^T}\frac{\partial \text{vec}(\mathbf{\Sigma}_{t,k})}{\partial \text{vec}(\mathbf{S}_{t,k})^T}\frac{\partial \text{vec}(\mathbf{S}_{t,k})}{\partial \text{vec}(\mathbf{\Sigma}_{t,k})^T}\mathbf{g}_{1,k}}{\partial \text{vec}(\mathbf{S}_{t,k})}\\
&=-\frac{1}{4}\frac{\partial \mathbf{g}_{1,k}^T \frac{\partial \text{vec}(\mathbf{S}_{t,k})}{\partial \text{vec}(\mathbf{\Sigma}_{t,k})^T}\mathbf{g}_{1,k}}{\partial \text{vec}(\mathbf{S}_{t,k})}=\frac{1}{4}\frac{\partial \mathbf{g}_{1,k}^T \mathbf{S}_{t,k}\otimes \mathbf{S}_{t,k}\mathbf{g}_{1,k}}{\partial \text{vec}(\mathbf{S}_{t,k})}\\
&=\frac{1}{2}\text{vec}(\mathbf{M}_{1,k}\mathbf{\Sigma}_{t,k}^{-1}\mathbf{M}_{1,k}).
\end{aligned} \tag{45}$$

Including the above result in (27) after removing the column-vectorisation operator yields the modified update rule for the posterior covariance $\mathbf{\Sigma}_t$, which is

$$\mathbf{\Sigma}_{t,k+1}=\mathbf{\Sigma}_{t,k}+\mathbf{M}_{1,k}+\frac{1}{2}\mathbf{M}_{1,k}\mathbf{\Sigma}_{t,k}^{-1}\mathbf{M}_{1,k}. \tag{46}$$

The modified rules (42) and (46) are consistent with those originally developed in [40]. That $\mathbf{\Sigma}_{t,k+1}$ in (46) must be positive definite if $\mathbf{\Sigma}_{t,k}$ is positive definite can be established by following the same argument presented under (42).

V. Simulation Results

A. Tracking Scenario and Algorithm Implementation

We adopt the simulation experiment used in [37]. The task is to track a moving target in a 2D plane using the bearing and range measurements obtained by a stationary sensor at the origin. The sampling period is 3s. The bearing and range measurements are corrupted by additive zero-mean Gaussian noise with standard deviations $\sigma_b=0.3^o$ and $\sigma_r=50$m.

The tracking process lasts for 300s, during which the target makes two 90^o turns with a constant acceleration of 1.07g. The first turn is a right turn from 100s to 132s, while the second turn is a left turn from 132s to 200s. Other times, the target motion follows a constant velocity (CV) model with Gaussian process noise having zero mean and a standard deviation of 0.01m/s^2. At the start of the tracking process, the target is located at $[155.88, 90]^T$km. It has a speed of 200m/s and moves towards southwest with velocity $[-100, -173.2]^T$m/s.

We replace the measurement update step of the VGF from [37] with the NGD-based rules developed in Sections III and IV to obtain new VGFs for estimating the target trajectory. At time t, the established VGFs first utilize the prediction step of the GHKF [12]–[14] with 32 sigma points to find the state predictive distribution $\mathcal{N}(\mathbf{x}_t;\mathbf{m}_t,\mathbf{P}_t)$. The NGD-based rules derived under two parameterizations of the Gaussian posterior, given in (25) and (27), and (32) and (34), or their modifications (46) and (42) are used to carry out the measurement update. All of these rules are iterative, and they start with $\boldsymbol{\mu}_{t,0}=\mathbf{m}_t$, $\mathbf{\Sigma}_{t,0}=\mathbf{P}_t$, and $\mathbf{S}_{t,0}=\mathbf{P}_t^{-1}$. The NGD-based update rules are deemed to have converged if the KLD between the state posteriors from two successive iterations is smaller than 10^{-6}. The above measurement update step will then be repeated two times for achieving the desired Gaussian filtering, each time with the state prediction covariance $\mathbf{P}_t$ reset to be

$$\mathbf{\Omega}_t=\frac{\nu}{\nu+1}\mathbf{P}_t+\frac{1}{\nu+1}\left(\mathbf{\Sigma}_t+(\boldsymbol{\mu}_t-\mathbf{m}_t)(\boldsymbol{\mu}_t-\mathbf{m}_t)^T\right). \tag{47}$$

The purpose of employing (47) with $\nu=4$ is to allow the adaptive adjustment of the state prediction covariance using the newly available measurements at time t.

Implementing the NGD-based update rules requires computing expectations with respected to the estimated Gaussian posterior. We utilize the low-discrepancy generalized Fibonacci grid [52] with 32 grid points to realize deterministic Gaussian sampling-based integration for evaluating these expectations. Other details on the VGF realization can be found in [37].

B. Results and Discussions

Four VGFs established in the previous subsection are simulated. The first two VGFs ('VGF1' and 'VGF1+PD') uses the mean and covariance to parameterize the Gaussian posterior.

$$-\frac{1}{4}\text{FIM}(\text{vec}(\mathbf{\Sigma}_{t,k}))^{-1}\frac{\partial \mathbf{g}_{1,k}^T \frac{\partial \text{vec}(\mathbf{S}_{t,k})}{\partial \text{vec}(\mathbf{\Sigma}_{t,k})^T}\frac{\partial^2 A(\text{vec}(\mathbf{S}_{t,k}))}{\partial \text{vec}(\mathbf{S}_{t,k})\partial \text{vec}(\mathbf{S}_{t,k})^T}\frac{\partial \text{vec}(\mathbf{S}_{t,k})}{\partial \text{vec}(\mathbf{\Sigma}_{t,k})^T}\mathbf{g}_{1,k}}{\partial \text{vec}(\mathbf{\Sigma}_{t,k})}. \tag{43}$$

'VGF1' applies (25) and (27), and 'VGF1+PD' utilizes (25) and the modified version of (27), (46), to carry out the NGD-based measurement update. The other two VGFs ('VGF2' and 'VGF2+PD') adopts the mean and precision matrix to parameterize the Gaussian posterior. Correspondingly, (32) and (34), and (32) and (42) are respectively employed by these two VGFs in their measurement update.

Figs. 1 and 2 plot the estimation root mean square errors (RMSEs) for the target position and velocity from the four simulated VGFs. For comparison, the results of CKF [15] ('CKF'), GHKF [12]–[14] ('GHKF'), the stochastic search KF [21] ('SKF'), and VGF from [37] are included. The results are obtained by averaging over 2,000 ensemble runs.

'VGF1' and 'VGF2', which are based on the NGD-based update rules derived in Sections III.A and III.B under two parameterizations of the Gaussian posterior, yield very similar accuracy. This is expected as the update rules for the posterior mean are the same and the updates of the posterior covariance and precision matrix become equivalent when the iterations converge (see Section III.C). Applying the modified rules (46) and (42) for ensuring that the computed posterior covariance and precision matrix are positive definite does not affect the performance. This is because the extra terms introduced in (46) and (42) are quadratic in the correction terms $\mathbf{M}_{1,k}$ and $\mathbf{M}_{2,k}$. They would become negligible when the iterations converge, reducing the modified rules to (27) and (34). Finally, the estimation accuray of the four simulated VGFs is close to that of the VGF from [37] developed under the natural parameterization of the Gaussian posterior. They are superior to existing GFs, especially when the target makes turns.

'VGF1', 'VGF1+PD', 'VGF2' and 'VGF2+PD' need 56-60 iterations on average for their NGD update to converge, taking 26-28 ms CPU time. The VGF wtih natural parameterization [37] requires about 4 iterations and 2.5 ms CPU time. 'CKF' and 'GHKF' use 0.15 ms and 0.3 ms. The difference comes from that with the natural parameterization, in each iteration, the posterior covariance is updated first and then used as the pre-conditioner to refine the posterior mean (see Section III.C). With two parameterizations considered in this work, the posterior covariance from the previous iteration is employed as the pre-conditioner. This necessitates the use of a small step size of 0.05 to avoid divergence, while the VGF from [37] has a much larger step size of 1. The 'SKF' from [21] requires 100 iterations due to approximating the true gradient.

VI. Conclusions

This paper considered the VGFs whose measurement update is realized via NGD-based forward KLD minimization. We re-derived the iterative measurement update rules under two parameterizations of the Gaussian posterior, one consisting of the mean and covariance and the other comprising the mean and precision matrix. The obtained algorithms were compared with the one developed under the natural parameterization of the Gaussian posterior. It was found empirically that the three parameterizations have similar tracking performance but the natural parameterization offers the quickest convergence.

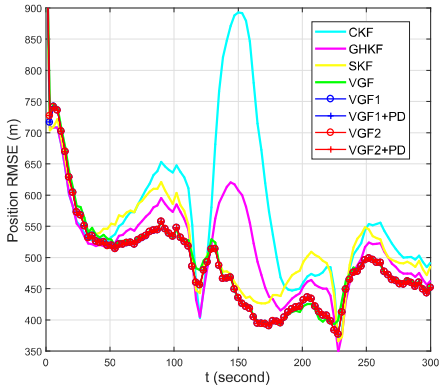

Fig. 1. Comparison of target position estimation RMSEs.

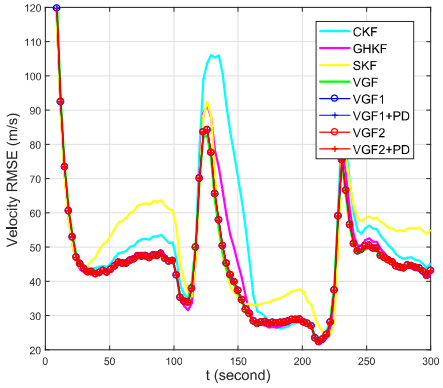

Fig. 2. Comparison of target velocity estimation RMSEs.

Appendix I

To make the derivation more general, the Gaussian approximation of the state prediction distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$, $\pi(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t; \mathbf{m}_t, \mathbf{P}_t)$, is expressed in its exponential family form as $\pi(\mathbf{x}_t) = \exp(\boldsymbol{\lambda}_t^T \boldsymbol{\phi}(\mathbf{x}_t) - A(\boldsymbol{\lambda}_t))$ [36], [51]. Therefore, finding the predictive mean $\mathbf{m}_t$ and covariance $\mathbf{P}_t$ of $\pi(\mathbf{x}_t)$ reduces to determining the natural parameter $\boldsymbol{\lambda}_t$, which is achieved by minimizing the backward KLD [44]

$$\mathrm{KLD}(p(\mathbf{x}_t|\mathbf{z}_{1:t-1})||\pi(\mathbf{x}_t)) \propto -\int p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) \log \pi(\mathbf{x}_t) \mathrm{d}\mathbf{x}_t.$$

Putting the exponential family form of $\pi(\mathbf{x}_t)$, taking the partial derivative with respect to $\boldsymbol{\lambda}_t$, setting the result to zero, and applying $\partial A(\boldsymbol{\lambda}_t)/\partial \boldsymbol{\lambda}_t = \boldsymbol{\eta}_t$, where $\boldsymbol{\eta}_t$ is the mean parameter of $\pi(\mathbf{x}_t)$ [51], we have that the minimizer must satisfy

$$\boldsymbol{\eta}_t = E_{p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})} \left[E_{p(\mathbf{x}_t|\mathbf{x}_{t-1})} [\boldsymbol{\phi}(\mathbf{x}_t)] \right]. \tag{48}$$

For Gaussian density (and other minimal exponential family distributions), it is known [37], [51] that the natural parameter $\boldsymbol{\lambda}_t$ can be uniquely found from the mean parameter $\boldsymbol{\eta}_t$. As such, the predictive mean $\mathbf{m}_t$ and covariance $\mathbf{P}_t$ of the approximate predictive distribution $\pi(\mathbf{x}_t)$ can thus be obtained via the moment matching (48).

REFERENCES

[1] S. Challa, M. R. Morelande, D. Mušicki, and R. J. Evans, *Fundamentals of Object Tracking*. Cambridge University Press, 2011.

[2] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of Basic Engineering*, vol. 83, no. 1, pp. 95–108, Mar. 1961.

[3] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*. YBS Publishing, 2011.

[4] Y. Huang, Y. Zhang, N. Li, Z. Wu, and J. A. Chambers, "A novel robust Student's t-based Kalman filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, pp. 1545–1554, June 2017.

[5] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2003.

[6] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: Theory, algorithms and software*. New York: Wiley, 2001.

[7] Y. Ollivier, "The extended Kalman filter is a natural gradient descent in trajectory space," *arXiv preprint*, Jan. 2019. [Online]. Available: arxiv.org/abs/1901.00696.

[8] Ángel. F. García-Fernández, L. Svensson, M. R. Morelande, and S. Särkkä, "Posterior linearization filter: Principles and implementation using sigma points," *IEEE Trans. Signal Process.*, vol. 63, pp. 5561–5573, Oct. 2015.

[9] M. Raitoharju, L. Svensson, Ángel. F. García-Fernández, and R. Piché, "Damped posterior linearization filter," *IEEE Signal Process. Lett.*, vol. 25, pp. 536–540, April 2018.

[10] S. Särkkä and L. Svensson, *Bayesian Filtering and Smoothing*, 2nd ed. New York: Cambridge University Press, 2023.

[11] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. Autom. Control*, vol. 45, pp. 477–482, Mar. 2000.

[12] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Trans. Autom. Control*, vol. 45, pp. 910–927, May 2000.

[13] B. Jia, M. Xin, and Y. Cheng, "Sparse-grid quadrature nonlinear filtering," *Automatica*, vol. 48, pp. 327–341, Feb. 2012.

[14] H. Meng, "State estimation-beyond Gaussian filtering," Ph.D. dissertation, University of New Orleans, 2022.

[15] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Trans. Autom. Control*, vol. 54, pp. 1254–1269, Jun. 2009.

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

[17] I. Arasaratnam, S. Haykin, and R. J. Elliott, "Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature," *Proc. IEEE*, vol. 95, pp. 953–977, May. 2007.

[18] V. Šmídl and A. Quinn, "Variational Bayesian filtering," *IEEE Trans. Signal Process.*, vol. 56, pp. 5020–5030, Oct. 2008.

[19] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. of the American Statistical Association*, vol. 112, pp. 859–877, July 2017.

[20] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt, "Advances in variational inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, pp. 2008–2026, August 2019.

[21] S. Gultekin and J. Paisley, "Nonlinear Kalman fltering with divergence minimization," *IEEE Trans. Signal Process.*, vol. 65, pp. 6319–6331, Dec. 2017.

[22] J. E. Darling and K. J. DeMars, "Minimization of the Kullback-Leibler divergence for nonlinear estimation," *J. of Guidance, Control and Dynamics*, vol. 40, pp. 1739–1748, July 2017.

[23] Y. Hu, Q. Pan, Z. Guo, Z. Shi, and Z. Hu, "Iterative nonlinear Kalman filtering via variational evidence lower bound maximization," in *Proc. Intl. Conf. Information Fusion (FUSION)*, Ottawa, Canada, Jul. 2019.

[24] M. Lambert, S. Bonnabel, and F. Bach, "The recursive variational Gaussian approximation (R-VGA)," *Statistics and Computing*, vol. 32, Feb. 2022.

[25] S. Hu, L. Guo, and J. Zhou, "An iterative nonlinear filter based on posterior distribution approximation via penalized Kullback-Leibler divergence minimization," *IEEE Signal Process. Lett.*, vol. 29, pp. 1137–1141, April 2022.

[26] L. Guo, S. Hu, J. Zhou, and X. R. Li, "Gaussian approximation filter based on divergence minimization for nonlinear dynamic systems," in *Proc. Intl. Conf. Information Fusion (FUSION)*, Linköping, Sweden, Jul. 2022.

[27] E. Laz and U. Orguner, "Gaussian mixture filtering with nonlinear measurements minimizing forward Kullback-Leibler divergence," *Signal Process.*, vol. 208, July 2023.

[28] H. Lan, J. Hu, Z. Wang, and Q. Cheng, "Variational nonlinear Kalman filtering with unknown process noise covariance," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, pp. 9177–9190, Dec. 2023.

[29] L. Guo, S. Hu, J. Zhou, and X. R. Li, "Recursive nonlinear filtering via Gaussian approximation with minimized Kullback-Leibler divergence," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, pp. 965–979, Feb. 2024.

[30] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, pp. 251–276, Feb. 1998.

[31] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *Journal of Machine Learning Research*, vol. 14, pp. 1303–1347, May 2013.

[32] Y. Ollivier, "Online natural gradient as a Kalman filter," *arXiv preprint*, March 2017. [Online]. Available: arxiv.org/abs/1703.00209.

[33] L. Shoji, K. Suzuki, and L. Kozachkov, "Is all learning (natural) gradient descent?" *arXiv preprint*, Sept. 2024. [Online]. Available: arxiv.org/abs/2409.16422.

[34] Y. Hu, X. Wang, Q. Pan, Z. Hu, and B. Moran, "Variational Bayesian Kalman filter using natural gradient," *Chinese Journal of Aeronautics*, vol. 35, pp. 1–10, May 2022.

[35] W. Cao, T. Zhang, Z. Sun, C. Liu, S.-T. Yau, and S. Li, "Nonlinear Bayesian filtering with natural gradient Gaussian approximation," *arXiv preprint*, Oct. 2024. [Online]. Available: arxiv.org/abs/2410.15832.

[36] B. Efron, *Exponential Families in Theory and Practice*. Cambridge University Press, 2022.

[37] Y. Liu, X. Li, L. Yang, L. S. Mihaylova, and J. Li, "On the Gaussian filtering for nonlinear dynamic systems using variational inference," in *Proc. Intl. Conf. Information Fusion (FUSION)*, Venice, Italy, Jul. 2024.

[38] M. E. Khan and W. Lin, "Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models," in *Proc. Intl. Conf. Artificial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, FL, USA, April 2017.

[39] M. Jones, P. G. Chang, and K. Murphy, "Bayesian online natural gradient (BONG)," in *Proc. Conf. Neural Information Processing Systems (NeurIPS)*, Vancouver, BC, Canada, Dec. 2024.

[40] W. Lin, M. Schmidt, and M. E. Khan, "Handling the positive-definite constraint in the Bayesian learning rule," in *Proc. Intl. Conf. Machine Learning (ICML)*, Vienna, Austria, July 2020, pp. 6116–6126.

[41] T. D. Barfoot, "Multivariate Gaussian variational inference by natural gradient descent," *arXiv preprint*, Jan. 2020. [Online]. Available: arxiv.org/abs/2001.10025

[42] T. D. Barfoot, J. R. Forbes, and D. J. Yoon, "Exactly sparse Gaussian variational inference with application to derivative-free batch nonlinear state estimation," *The International Journal of Robotics Research*, vol. 39, pp. 1473–1502, July 2020.

[43] M. Wüthrich, S. Trimpe, D. Kappler, and S. Schaal, "A new perspective and extension of the Gaussian filter," *arXiv preprint*, June 2015. [Online]. Available: arxiv.org/abs/1504.07941.

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

[45] J. Knoblauch, J. Jewson, and T. Damoulas, "Generalized variational inference: Three arguments for deriving new posteriors," *arXiv preprint*, Dec. 2019, available at: arxiv.org/abs/1904.02063.

[46] K. B. Peterson and M. S. Peterson, *The Matrix Cookbook*, 2012. [Online]. Available: http://matrixcookbook.com

[47] M. E. Khan and H. Rue, "The Bayesian learning rule," *J. of Machine Learning Research*, vol. 24, pp. 1–46, Sept. 2023.

[48] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, 1993.

[49] M. Opper and C. Archambeau, "The variational Gaussian approximation revisited," *Neural Computation*, vol. 21, pp. 786–792, March 2009.

[50] W. Lin, M. E. Khan, and M. Schmidt, "Stein's lemma for the reparameterization trick with exponential family mixtures," *arXiv preprint*, Oct. 2019, available at: arxiv.org/abs/1910.13398.

[51] K. P. Murphy, *Probabilistic Machine Learning: Advanced Topics*. The MIT Press, 2023.

[52] D. Frisch and U. D. Hanebeck, "The generalized Fibonacci grid as low-discrepancy point set for optimal deterministic Gaussian sampling," *J. of Advances in Information Fusion*, vol. 18, pp. 16–34, June 2023.