This is a repository copy of *Robust Bayesian Filtering Using Bayesian Model Averaging and Restricted Variational Bayes*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/131390/

Version: Accepted Version

Proceedings Paper:
Khalid, S., Rehman, N.U., Abrar, S. et al. (1 more author) (2018) Robust Bayesian Filtering Using Bayesian Model Averaging and Restricted Variational Bayes. In: Proceedings of the International Conference on Information Fusion. International Conference on Information Fusion, 10-13 Jul 2018, Cambridge, UK. IEEE . ISBN 978-0-9964527-6-2

10.23919/ICIF.2018.8455608

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

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/

Robust Bayesian Filtering Using Bayesian Model Averaging and Restricted Variational Bayes

S. S. Khalid, N. U. Rehman
EE Department, COMSATS Institute
of Information Technology
Islamabad, Pakistan
{safwan_khalid,naveed.rehman}
@comsats.edu.pk

Shafayat Abrar
School of Science and Engineering
Habib University, Karachi, Pakistan
shafayat.abrar@sse.habib.edu.pk

Lyudmila Mihaylova
Department of Automatic
Control and Systems Engineering
The University of Sheffield, UK
L.S.Mihaylova@sheffield.ac.uk

***Abstract*—Bayesian filters can be made robust to outliers if the solutions are developed under the assumption of heavy-tailed distributed noise. However, in the absence of outliers, these robust solutions perform worse than the standard Gaussian assumption based filters. In this work, we develop a novel robust filter that adopts both Gaussian and multivariate t-distributions to model the outliers contaminated measurement noise. The effects of these distributions are combined within a Bayesian Model Averaging (BMA) framework. Moreover, to reduce the computational complexity of the proposed algorithm, a restricted variational Bayes (RVB) approach handles the multivariate t-distribution instead of its standard iterative VB (IVB) counterpart. The performance of the proposed filter is compared against a standard cubature Kalman filter (CKF) and a robust CKF (employing IVB method) in a representative simulation example concerning target tracking using range and bearing measurements. In the presence of outliers, the proposed algorithm shows a 38% improvement over CKF in terms of root-mean-square-error (RMSE) and is computationally 2.5 times more efficient than the robust CKF.**

I. Introduction

Target tracking deals with the estimation of unknown states, such as position, velocity and acceleration of a moving target, using noisy measurements in a given coordinate space. Algorithms that can accurately track the mobility of a target offer numerous advantages in a wide range of applications. The most obvious example is tracking an aircraft using radar measurements. It is of immense importance in many military applications and is also essential for air traffic control required by civilian airlines. Some other examples include tracking of a mobile node in a cellular network which is required for efficient radio resource management and tracking in autonomous cars and robots. Target tracking algorithms, despite having a diverse range of applications, employ a common structure based on the Bayesian filtering framework for extracting useful information from the available data. The standard Bayesian filtering solutions such as traditional Kalman filters (in case of linear systems) and sigma-point filters (e.g., Cubature Kalman Filter (CKF), Unscented Kalman Filter (UKF), etc., for nonlinear systems), assume that the noises have Gaussian distribution [1]. In practice, however, large deviations (outliers) occur in real data frequently and these cannot be modeled accurately by a Gaussian distribution only [2]. As a result, filters relying on the Gaussian assumption do not perform well when outliers are present[3].

A Bayesian filter can be made robust to outliers if the Gaussian assumption is dropped in favor of a heavy-tailed distribution. A suitable choice is the use of multivariate generalization of Student t-distribution [2–8]; hereafter, referred to as t-distribution. However, the incorporation of t-distributed uncertainties in a Bayesian framework is not trivial as the required posterior probability becomes intractable. Recently, a number of works [2, 5, 6, 9] have advocated the use of variational Bayes (VB) framework to handle t-distributed measurement noise in Bayesian filters. In the VB method, a solution is obtained by approximating the intractable posterior probability density function into a tractable factored form. These state-of-the-art robust solutions, however, suffer from two drawbacks: 1) The standard application of the VB method results in an iterative procedure (IVB) that requires a number of fixed-point iterations to converge to an admissible inference. These iterations, though few in number (usually four or five), may become prohibitive in real-time applications due to the involvement of matrix inversion operations; 2) The resulting solutions though indeed robust to outliers, do not perform well in the absence of outliers, as compared to the conventional filters based on Gaussian assumption.

In this work, we propose a filter able to deal with both these challenges by,

1) Adopting a restricted VB (RVB) approach to get rid of the iterative procedure, and develop an approximate computationally-efficient recursive solution for Bayesian filtering under t-distributed measurement noise.
2) Instead of modeling the observation noise using a single t-distributed process, we advocate the use of two separate models, one Gaussian distributed and one t-distributed. The proposed filter then combines these two models using a Bayesian Model Averaging (BMA) [7] approach.

Note that BMA based particle filters have recently been discussed in [7] and [10]. However, particle filters are known to exhibit heavy computational expense and this leads to challenges in many real-time applications. As we shall show in

this work, the proposed BMA-RVB method can easily be combined with sigma-point methods to develop computationally efficient robust solutions for nonlinear systems. The rest of this paper is organized as follows: in Section II, we describe the system models and develop the proposed filtering algorithm. In section III, we present simulation results and in Section IV, we draw conclusions.

A. Notations

We represent scalers using small letters. Column vectors denoting states and measurements are represented by bold-faced small letters. Matrices are represented using bold-faced capital letters. A set of column vectors is also represented using bold-faced capital letters. We use $\boldsymbol{I}_n$ to denote an $n \times n$ identity matrix. We use T in superscript to represent the transpose operation of a matrix. A variable $\boldsymbol{x}$ that is distributed according to t-distribution is denoted as $\boldsymbol{x} \sim \text{St}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \eta)$, i.e.,

$$p(\boldsymbol{x}) = \frac{\Gamma((\eta+d)/2)}{\Gamma(\eta/2)} \frac{1}{(\eta\pi)^{d/2}\sqrt{\boldsymbol{\Sigma}}} \left(1 + \frac{\delta^2(\boldsymbol{x})}{\eta}\right)^{-(\eta+d)/2},$$

where $\delta^2(\boldsymbol{x}) = (\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$, $d = \dim(\boldsymbol{x})$, $\boldsymbol{\mu}$ is the mean, η is the degree-of-freedom parameter, and $\boldsymbol{\Sigma}$ is the scale matrix of the $p(\boldsymbol{x})$.

Note that η is a shape parameter that determines tail-behavior [11]. Heavier tails are obtained when η is close to one. Conversely, for larger values of η, $p(\boldsymbol{x})$ approaches the standard normal distribution. Also note that t-distribution has infinite variance for $\eta < 2$; therefore, throughout this work, we shall assume that $\eta > 2$. Finally, the covariance matrix of $\boldsymbol{x} \sim \text{St}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \eta)$ is given by $\frac{\eta}{\eta-2}\boldsymbol{\Sigma}$ for $\eta > 2$.

II. System Model and Proposed Algorithm

Let us consider the following dynamic system:

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_k, \tag{1a}$$

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k, \tag{1b}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ is the dynamic state vector, $\boldsymbol{y}_k \in \mathbb{R}^m$ is the observation vector, $f(\cdot)$ and $h(\cdot)$ are arbitrary nonlinear functions, $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}; \boldsymbol{Q}_k)$ models the uncertainties in the system model and $\boldsymbol{v}_k$ is the outliers contaminated observation noise. To account for the effects of the outliers, we model $\boldsymbol{v}_k$ as a combination of a Gaussian and a t-distribution. The transition between these two distributions is governed by a first-order jump Markovian process s_k that can take two possible values s_1 and s_2, i.e., $\boldsymbol{v}_k^{(s_k=s_1)} \sim \mathcal{N}(\boldsymbol{0}; \boldsymbol{R}_k)$ and $\boldsymbol{v}_k^{(s_k=s_2)} \sim \text{St}(\boldsymbol{0}; \boldsymbol{\Sigma}_k; \eta)$, where $\boldsymbol{\Sigma}_k = \frac{\eta-2}{\eta}\boldsymbol{R}_k$. Hereafter, we use the notation $s_k^{(i)}$ to denote $s_k = s_i$ for $i = 1, 2$. We assume that the transition probabilities $p(s_k^{(i)}|s_{k-1}^{(j)}) = \pi_{ji}$ are known a priori. Note that the noise sequences, $\{\boldsymbol{w}_k\}$ and $\{\boldsymbol{v}_k\}$, are assumed to be independent for each k.

Let $\boldsymbol{Y}_k := \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_k\}$ be the set of all available observations at instant k; the task of a Bayesian filtering algorithm is to recursively evaluate an estimate of the state vector $\hat{\boldsymbol{x}}_{k|k} = \text{E}[\boldsymbol{x}_k|\boldsymbol{Y}_k] = \int \boldsymbol{x}_k p(\boldsymbol{x}_k|\boldsymbol{Y}_k) d\boldsymbol{x}_k$[1]. Noting that at any instant, the observation noise $\boldsymbol{v}_k$ may belong to one of the two possible models, we expand $p(\boldsymbol{x}_k|\boldsymbol{Y}_k)$ as follows:

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_k) = \sum_{i=1}^{2} \underbrace{p(\boldsymbol{x}_k|\boldsymbol{Y}_k, s_k^{(i)})}_{\text{Posterior}} \underbrace{p(s_k^{(i)}|\boldsymbol{Y}_k)}_{\text{Weighting Factor}}. \tag{2}$$

Using Bayes theorem, $p(\boldsymbol{x}_k|\boldsymbol{Y}_k, s_k^{(i)})$ can be written as a product of a *likelihood* and a *prediction* density, as follows:

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_k, s_k^{(1i)}) \propto \underbrace{p(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{Y}_{k-1}, s_k^{(i)})}_{\text{likelihood}} \underbrace{p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1}, s_k^{(i)})}_{\text{Prediction}}. \tag{3}$$

In the following, we derive expressions for the evaluation of *prediction*, *likelihood*, *posterior*, and the *weighting factor*. We discuss how the required probability $p(\boldsymbol{x}_k|\boldsymbol{Y}_k)$ is approximated at each instant k and also discuss the computational cost of the resulting algorithm.

A. Prediction

Let us first consider the evaluation of prediction density. By introducing marginalization over $\boldsymbol{x}_{k-1}$, we can write

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1}, s_k^{(i)}) &= \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{Y}_{k-1}, s_k^{(i)}) \times \\ &\qquad p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1}, s_k^{(i)}) d\boldsymbol{x}_{k-1} \\ &= \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1}) d\boldsymbol{x}_{k-1}, \end{aligned} \tag{4}$$

where we have used the fact that the probability of $\boldsymbol{x}_k$ is completely specified given $\boldsymbol{x}_{k-1}$ and the value of s_k at kth instant does not affect the probability of $\boldsymbol{x}_{k-1}$. Accordingly, the prediction density is independent of the value of $s_k^{(i)}$, i.e., $p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1}, s_k^{(i)}) = p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1})$. In BMA framework, we assume that $p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1})$ is approximated using a single Gaussian distribution, i.e., $p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1}) \approx \mathcal{N}_{\boldsymbol{x}_{k-1}}(\hat{\boldsymbol{x}}_{k-1|k-1}; \boldsymbol{P}_{k-1|k-1})$, where $\hat{\boldsymbol{x}}_{k-1|k-1}$ and $\boldsymbol{P}_{k-1|k-1}$ are known from previous recursion. Also, from (1a), we note that $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = \mathcal{N}_{\boldsymbol{x}_k}(f(\boldsymbol{x}_{k-1}); \boldsymbol{Q}_k)$. To derive a closed-form expression for (4), we still require to *linearize the nonlinear function* $f(\cdot)$. To achieve this result, we apply statistical linear regression (SLR) [12, 13] on $f(\boldsymbol{x}_{k-1})$ as follows:

$$f(\boldsymbol{x}_{k-1}) \approx \boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{b}_{k-1} + \boldsymbol{e}_{k-1}^f, \tag{5}$$

where $\boldsymbol{F}_{k-1} \in \mathbb{R}^{n\times n}$, $\boldsymbol{b}_{k-1} \in \mathbb{R}^n$ are to be determined and $\boldsymbol{e}_{k-1}^f$ is the linearization error that is assumed to be a zero-mean Gaussian distributed process with covariance equal to $\boldsymbol{\Omega}_{k-1}^f$. We also assume that $\boldsymbol{e}_{k-1}^f$ is independent from $\boldsymbol{x}_{k-1}$ and $\boldsymbol{w}_k$. Note that $\boldsymbol{b}_{k-1}$ is introduced to make the approximation in (5) unbiased, we evaluate $\boldsymbol{b}_{k-1}$ as

$$\begin{aligned} \boldsymbol{b}_{k-1} &= \text{E}[f(\boldsymbol{x}_{k-1}) - \boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1}] \\ &= \bar{\boldsymbol{x}}_{k|k-1} - \boldsymbol{F}_{k-1}\hat{\boldsymbol{x}}_{k-1|k-1}, \end{aligned} \tag{6}$$

[1]For nonlinear systems, the required expectation operator for the optimal estimate does not admit a closed-form solution in general, and we work with approximations only.

where

$$\bar{\boldsymbol{x}}_{k|k-1} = \int f(\boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1}) d\boldsymbol{x}_{k-1}. \tag{7}$$

Now, from (5), the linearization error can be written as $\boldsymbol{e}^f_{k-1} \approx f(\boldsymbol{x}_{k-1}) - \boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1} - \boldsymbol{b}_{k-1}$. The value of $\boldsymbol{F}_{k-1}$ is evaluated by minimizing the mean square of this linearization error, i.e.,

$$\begin{aligned}\boldsymbol{F}^{\dagger}_{k-1} = \underset{F}{\operatorname{argmin}}\, \mathrm{E}[(f(\boldsymbol{x}_{k-1}) - \boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1} - \boldsymbol{b}_{k-1})^T \times \\ (f(\boldsymbol{x}_{k-1}) - \boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1} - \boldsymbol{b}_{k-1}) | \boldsymbol{Y}_{k-1}] \\ = \mathrm{E}\Big[\big\{(f(\boldsymbol{x}_{k-1}) - \bar{\boldsymbol{x}}_{k-1}) - \boldsymbol{F}_{k-1}(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1|k-1})\big\} \times \\ \big\{(f(\boldsymbol{x}_{k-1}) - \bar{\boldsymbol{x}}_{k-1}) - \boldsymbol{F}_{k-1}(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1|k-1})\big\}^T | \boldsymbol{Y}_{k-1}\Big].\end{aligned} \tag{8}$$

Let us define $\boldsymbol{P}^{xf}_{k-1} := \mathrm{E}[(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1|k-1})(f(\boldsymbol{x}_{k-1}) - \bar{\boldsymbol{x}}_{k|k-1})^T | \boldsymbol{Y}_{k-1}]$, then taking the derivative of (8) with respect to $\boldsymbol{F}_{k-1}$ and setting it to zero, we get

$$\boldsymbol{F}^{\dagger}_{k-1} = (\boldsymbol{P}^{xf}_{k-1})^T \boldsymbol{P}^{-1}_{k-1|k-1}. \tag{9}$$

In the following, we simply use $\boldsymbol{F}_{k-1}$ instead of $\boldsymbol{F}^{\dagger}_{k-1}$, to keep the notation simple. Using the expression for $\boldsymbol{F}_{k-1}$, the covariance matrix of $\boldsymbol{e}^f_{k-1}$ is evaluated as

$$\begin{aligned}\boldsymbol{\Omega}^f_{k-1} &:= \mathrm{E}[\boldsymbol{e}^f_{k-1}(\boldsymbol{e}^f_{k-1})^T | \boldsymbol{Y}_{k-1}] \\ &= \boldsymbol{P}^{ff}_{k-1} - \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1|k-1}\boldsymbol{F}^T_{k-1},\end{aligned} \tag{10}$$

where

$$\begin{aligned}\boldsymbol{P}^{ff}_{k-1} := \int (f(\boldsymbol{x}_{k-1}) - \bar{\boldsymbol{x}}_{k|k-1}) \times \\ (f(\boldsymbol{x}_{k-1}) - \bar{\boldsymbol{x}}_{k|k-1})^T p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1})\, d\boldsymbol{x}_{k-1}.\end{aligned} \tag{11}$$

Inserting (5) in (1a), we can write

$$\boldsymbol{x}_k \approx \boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{b}_{k-1} + \boldsymbol{e}^f_{k-1} + \boldsymbol{w}_k. \tag{12}$$

Consequently, we can approximate $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \approx \mathcal{N}_{\boldsymbol{x}_k}(\boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{b}_{k-1}; \boldsymbol{Q}_k + \boldsymbol{\Omega}^f_{k-1})$. Accordingly, the expression in (4) becomes

$$\begin{aligned}p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1}, s^{(1)}_k) \approx \int \mathcal{N}_{\boldsymbol{x}_k}(\boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{b}_{k-1}; \boldsymbol{Q}_k + \boldsymbol{\Omega}^f_{k-1}) \\ \mathcal{N}_{\boldsymbol{x}_{k-1}}(\hat{\boldsymbol{x}}_{k-1|k-1}; \boldsymbol{P}_{k-1|k-1})\, d\boldsymbol{x}_{k-1}.\end{aligned} \tag{13}$$

To develop a closed-form expression of (13) we require the following theorem:

Theorem 1 (Gaussian Product Theorem [14]): Let $\boldsymbol{x}_1, \boldsymbol{\mu}_1 \in \mathbb{R}^n$, $\boldsymbol{H} \in \mathbb{R}^{m \times n}$, $\boldsymbol{x}_2 \in \mathbb{R}^m$ and $\boldsymbol{P}_1, \boldsymbol{P}_2$ be positive definite matrices, then

$$\mathcal{N}_{\boldsymbol{x}_2}(\boldsymbol{H}\boldsymbol{x}_1; \boldsymbol{P}_2)\mathcal{N}_{\boldsymbol{x}_1}(\boldsymbol{\mu}_1; \boldsymbol{P}_1) = \mathcal{N}_{\boldsymbol{x}_2}(\boldsymbol{H}\boldsymbol{\mu}_1; \boldsymbol{P}_3)\mathcal{N}_{\boldsymbol{x}_1}(\boldsymbol{\mu}; \boldsymbol{P}),$$

where $\boldsymbol{P}_3 = \boldsymbol{H}\boldsymbol{P}_1\boldsymbol{H}^T + \boldsymbol{P}_2$, $\boldsymbol{\mu} = \boldsymbol{\mu}_1 + \boldsymbol{K}(\boldsymbol{x}_2 - \boldsymbol{H}\boldsymbol{\mu}_1)$, $\boldsymbol{P} = \boldsymbol{P}_1 - \boldsymbol{K}\boldsymbol{H}\boldsymbol{P}_1$ and $\boldsymbol{K} = \boldsymbol{P}_1\boldsymbol{H}^T\boldsymbol{P}_3^{-1}$.

Applying Gaussian Product Theorem (GPT) on (13), we get

$$\begin{aligned}p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1}) \approx \int \mathcal{N}_{\boldsymbol{x}_k}(\boldsymbol{F}_{k-1}\hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{b}_{k-1}; \boldsymbol{P}_{k|k-1}) \times \\ \mathcal{N}_{\boldsymbol{x}_{k-1}}(\boldsymbol{\mu}_k; \boldsymbol{P}_k) d\boldsymbol{x}_{k-1} \\ = \mathcal{N}_{\boldsymbol{x}_k}(\hat{\boldsymbol{x}}_{k|k-1}; \boldsymbol{P}_{k|k-1}),\end{aligned} \tag{14}$$

where $\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{F}_{k-1}\hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{b}_{k-1} = \bar{\boldsymbol{x}}_{k|k-1}$ and $\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1|k-1}\boldsymbol{F}^T_{k-1} + \boldsymbol{Q}_k + \boldsymbol{\Omega}^f_{k-1} = \boldsymbol{P}^{ff}_{k-1} + \boldsymbol{Q}_k$.

B. Likelihood

The expressions for likelihood can be found from (1b). Firstly, we note that $p(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{Y}_{k-1}, s^{(i)}_k) = p(\boldsymbol{y}_k|\boldsymbol{x}_k, s^{(i)}_k)$. Now, when $s_k = s_1$ (i.e., $\boldsymbol{v}_k$ is Gaussian), we have $p(\boldsymbol{y}_k|\boldsymbol{x}_k, s^{(1)}_k) = \mathcal{N}_{\boldsymbol{y}_k}(h(\boldsymbol{x}_k); \boldsymbol{R}_k)$. Similarly, for $s_k = s_2$ (i.e., $\boldsymbol{v}_k$ is t-distributed), we have $p(\boldsymbol{y}_k|\boldsymbol{x}_k, s^{(2)}_k) = \mathrm{St}(h(\boldsymbol{x}_k); \boldsymbol{\Sigma}_k; \eta)$

C. Posterior

From (3), we note that the posterior probability density function is proportional to the product of likelihood and prediction densities. Since the expression for likelihood is dependent on $s^{(i)}_k$; therefore, we evaluate $p(\boldsymbol{x}_k|\boldsymbol{Y}_k, s^{(i)}_k)$ separately for $i = 1$ and $i = 2$, in the following:

For $s_k = s_1$:

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_k, s^{(1)}_k) \propto \mathcal{N}_{\boldsymbol{y}_k}(h(\boldsymbol{x}_k); \boldsymbol{R}_k)\mathcal{N}_{\boldsymbol{x}_k}(\hat{\boldsymbol{x}}_{k|k-1}; \boldsymbol{P}_{k|k-1}). \tag{15}$$

To apply GPT on (15), we first linearize $h(\boldsymbol{x}_k)$. We apply SLR on $h(\boldsymbol{x}_k)$, as follows:

$$h(\boldsymbol{x}_k) \approx \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{c}_k + \boldsymbol{e}^h_k. \tag{16}$$

Using a procedure, similar to that outlined above, for $f(\boldsymbol{x}_{k-1})$, we get $\boldsymbol{c}_k = \bar{\boldsymbol{y}}_{k|k-1} - \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1}$, where

$$\bar{\boldsymbol{y}}_{k|k-1} = \int h(\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1})\, d\boldsymbol{x}_k. \tag{17}$$

Also, $\boldsymbol{H}_k = (\boldsymbol{P}^{xh}_k)^T \boldsymbol{P}^{-1}_{k|k-1}$, where

$$\boldsymbol{P}^{xh}_k = \int (\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1})(h(\boldsymbol{x}_k) - \bar{\boldsymbol{y}}_{k|k-1})^T p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1})\, d\boldsymbol{x}_k, \tag{18}$$

$$\boldsymbol{P}^{hh}_k = \int (h(\boldsymbol{x}_k) - \bar{\boldsymbol{y}}_{k|k-1})(h(\boldsymbol{x}_k) - \bar{\boldsymbol{y}}_{k|k-1})^T p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1}) d\boldsymbol{x}_k. \tag{19}$$

The linearization error $\boldsymbol{e}^h_k$ is assumed to be Gaussian distributed with zero mean and covariance equal to $\boldsymbol{\Omega}^h_k$, where $\boldsymbol{\Omega}^h_k = \boldsymbol{P}^{hh}_k - \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}^T_k$. Inserting (16) in (1b), we get a linearized observation model as follows:

$$\boldsymbol{y}_k \approx \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{c}_k + \boldsymbol{e}^h_k + \boldsymbol{v}_k. \tag{20}$$

Consequently, $p(\boldsymbol{y}_k|\boldsymbol{x}_k, s_k^{(1)}) \approx \mathcal{N}_{\boldsymbol{y}_k}(\boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{c}_k; \boldsymbol{\Omega}_k^h + \boldsymbol{R}_k)$. Inserting (20) in (15) and applying GPT, we get

$$\begin{aligned} &p(\boldsymbol{x}_k|\boldsymbol{Y}_k, s_k^{(1)}) \\ &\propto \mathcal{N}_{\boldsymbol{y}_k}(\boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{c}_k; \boldsymbol{\Omega}_k^h + \boldsymbol{R}_k)\mathcal{N}_{\boldsymbol{x}_k}(\hat{\boldsymbol{x}}_{k|k-1}; \boldsymbol{P}_{k|k-1}) \\ &\propto \mathcal{N}_{\boldsymbol{y}_k}(\boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{c}_k; \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k + \boldsymbol{\Omega}_k^h)\times \\ &\quad \mathcal{N}_{\boldsymbol{x}_k}(\hat{\boldsymbol{x}}_{k|k}^{(1)}; \boldsymbol{P}_{k|k}^{(1)}). \end{aligned} \tag{21}$$

Note that $\mathcal{N}_{\boldsymbol{y}_k}(\boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1}+\boldsymbol{c}_k; \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T+\boldsymbol{R}_k+\boldsymbol{\Omega}_k^h)$ is a term independent of $\boldsymbol{x}_k$; hence, dropping $\mathcal{N}_{\boldsymbol{y}_k}(\cdot,\cdot)$ in (21), we obtain $p(\boldsymbol{x}_k|\boldsymbol{Y}_k, s_k^{(1)}) = \mathcal{N}_{\boldsymbol{x}_k}(\hat{\boldsymbol{x}}_{k|k}^{(1)}; \boldsymbol{P}_{k|k}^{(1)})$, where

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k}^{(1)} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k^{(1)}(\boldsymbol{y}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1} - \boldsymbol{c}_k) \\ &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k^{(1)}(\boldsymbol{y}_k - \bar{\boldsymbol{y}}_{k|k-1}), \end{aligned} \tag{22}$$

the expression for $\boldsymbol{K}_k^{(1)}$ is given as

$$\begin{aligned} \boldsymbol{K}_k^{(1)} &= \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T(\boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{\Omega}_k^h + \boldsymbol{R}_k)^{-1} \\ &= \boldsymbol{P}_k^{xh}(\boldsymbol{P}_k^{hh} + \boldsymbol{R}_k)^{-1}, \end{aligned} \tag{23}$$

and the term $\boldsymbol{P}_{k|k}^{(1)}$ can be evaluated as

$$\begin{aligned} \boldsymbol{P}_{k|k}^{(1)} &= \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k^{(1)}\boldsymbol{H}_k\boldsymbol{P}_{k|k-1} \\ &= \boldsymbol{P}_{k|k-1} - \boldsymbol{P}_k^{xh}(\boldsymbol{P}_k^{hh} + \boldsymbol{R}_k)^{-1}(\boldsymbol{P}_k^{xh})^T. \end{aligned} \tag{24}$$

For s_k = s_2:
Similar to the previous case, we write

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{Y}_k, s_k^{(2)}) &\propto p(\boldsymbol{y}_k|\boldsymbol{x}_k, s_k^{(2)})p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1}) \\ &\propto \mathrm{St}(h(\boldsymbol{x}_k); \boldsymbol{\Sigma}_k; \eta)p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1}). \end{aligned} \tag{25}$$

By introducing a Gamma distributed auxiliary variable $\lambda_k \sim \mathrm{G}_{\lambda_k}(\frac{\eta}{2}, \frac{\eta}{2})$, the density $p(\boldsymbol{y}_k|\boldsymbol{x}_k, s_k^{(2)})$ may be expressed as [2]:

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k, s_k^{(2)}) = \int p(\boldsymbol{y}_k|\boldsymbol{x}_k, s_k^{(2)}, \lambda_k)p(\lambda_k)\, d\lambda_k, \tag{26}$$

where $p(\boldsymbol{y}_k|\boldsymbol{x}_k, s_k^{(2)}, \lambda_k) = \mathcal{N}_{\boldsymbol{y}_k}(h(\boldsymbol{x}_k); \frac{1}{\lambda_k}\boldsymbol{\Sigma}_k)$. Furthermore, the joint density $p(\boldsymbol{x}_k, \lambda_k|\boldsymbol{Y}_k, s_k^{(2)})$ may be expressed as:

$$\begin{aligned} p(\boldsymbol{x}_k, \lambda_k|\boldsymbol{Y}_k, s_k^{(2)}) &\propto p(\boldsymbol{y}_k|\boldsymbol{x}_k, s_k^{(2)}, \lambda_k)p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1})p(\lambda_k) \\ &= \mathcal{N}_{\boldsymbol{y}_k}\big(h(\boldsymbol{x}_k), \lambda_k^{-1}\boldsymbol{\Sigma}_k\big)\,\mathcal{N}_{\boldsymbol{x}_k}\big(\hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{P}_{k|k-1}\big)\mathrm{G}_{\lambda_k}\big(\frac{\eta}{2}, \frac{\eta}{2}\big), \end{aligned} \tag{27}$$

where we exploit the facts that $p(\boldsymbol{x}_k|\lambda_k, \boldsymbol{Y}_{k-1}) = p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1})$ and $p(\lambda_k|\boldsymbol{Y}_{k-1}) = p(\lambda_k)$. Note that the required posterior density $p(\boldsymbol{x}_k|\boldsymbol{Y}_k, s_k^{(2)})$ can be evaluated by marginalizing (27) over λ_k. To make this tractable, we approximate $p(\boldsymbol{x}_k, \lambda_k|\boldsymbol{Y}_k, s_k^{(2)})$ as a product of two independent factors, i.e., $p(\boldsymbol{x}_k, \lambda_k|\boldsymbol{Y}_k, s_k^{(2)}) \approx f_1(\boldsymbol{x_k}|\boldsymbol{Y}_k, s_k^2)f_2(\lambda_k|\boldsymbol{Y}_k)$. In the VB framework, $f_1(\cdot)$ and $f_2(\cdot)$ are determined by minimizing Kullback-Leibler (KL) divergence between the true and the approximate posteriors. If no fixed functional form is assumed for $f_1(\cdot)$ and $f_2(\cdot)$ then minimizing KL divergence results in coupling of moments of $f_1(\cdot)$ and $f_2(\cdot)$. Consequently, a number of fixed-point iterations are required to arrive at a solution (refer to iterative solutions in [2, 15]). These iterations, however, may be avoided if a fixed functional form is imposed on one of the distributions (say, as in our case, $f_2(\lambda_k|\boldsymbol{Y}_k)$). The other factor (i.e., $f_1(\boldsymbol{x_k}|\boldsymbol{Y}_k)$) can then be evaluated using the following proposition:

Proposition 1 ([15]): Let $f(\boldsymbol{\theta}|\boldsymbol{Y})$ be the posterior distribution of multivariate parameter $\boldsymbol{\theta}$, where the latter is partitioned into two sub-vectors of parameters, $\boldsymbol{\theta} = [{\boldsymbol{\theta_1}}^t, {\boldsymbol{\theta_2}}^t]^t$. Let $\widehat{f}(\boldsymbol{\theta}|\boldsymbol{Y})$ be an approximation of $f(\boldsymbol{\theta}|\boldsymbol{Y})$ of the kind $\widehat{f}(\boldsymbol{\theta}|\boldsymbol{Y}) = \widehat{f}(\boldsymbol{\theta_1}|\boldsymbol{Y})\widehat{f}(\boldsymbol{\theta_2}|\boldsymbol{Y})$, where $\widehat{f}(\boldsymbol{\theta_2}|\boldsymbol{Y})$ be a posterior distribution of $\boldsymbol{\theta_2}$ of fixed functional form. Then, the minimum KL divergence, i.e., $\mathrm{KL}(\widehat{f}(\boldsymbol{\theta}|\boldsymbol{Y})||f(\boldsymbol{\theta}|\boldsymbol{Y}))$, is reached for $\widehat{f}(\boldsymbol{\theta_1}|\boldsymbol{Y}) \propto \exp\big(\mathrm{E}_{\widehat{f}(\boldsymbol{\theta_2}|\boldsymbol{Y})}[\ln(f(\boldsymbol{\theta}|\boldsymbol{Y}))]\big)$.

While the proposition is valid for any $\widehat{f}(\boldsymbol{\theta_2}|\boldsymbol{Y})$, the choice of the functional form, however, greatly affects the accuracy of the resulting algorithm. Owing to [15], a reasonable choice is to select the exact marginal distribution of the joint posterior, i.e., using (27)

$$\begin{aligned} p(\lambda_k|\boldsymbol{Y}_k) \propto \int &\mathcal{N}_{\boldsymbol{y}_k}\big(h(\boldsymbol{x}_k), \lambda_k^{-1}\boldsymbol{\Sigma}_k\big)\,\mathcal{N}_{\boldsymbol{x}_k}\big(\hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{P}_{k|k-1}\big)\times \\ &\mathrm{G}_{\lambda_k}\big(\frac{\eta}{2}, \frac{\eta}{2}\big)\, d\boldsymbol{x}_k. \end{aligned} \tag{28}$$

However, the exact marginal in (28) does not yield a tractable form. On the other hand, if we replace $\mathcal{N}_{\boldsymbol{x}_k}(\widehat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{P}_{k|k-1})$ in the marginalization integral by its certainty equivalence approximation [16], i.e., $\delta(\boldsymbol{x}_k - \widehat{\boldsymbol{x}}_{k|k-1})$, where $\delta(\cdot)$ denotes the Dirac delta function, then the resulting posterior becomes Gamma distributed; this is shown below:

$$f_2(\lambda_k|\boldsymbol{Y}_k) \propto \int \mathcal{N}_{\boldsymbol{y}_k}\big(h(\boldsymbol{x}_k), \lambda_k^{-1}\boldsymbol{\Sigma}_k\big)\delta\big(\boldsymbol{x}_k - \widehat{\boldsymbol{x}}_{k|k-1}\big)\times \tag{29a}$$

$$\mathrm{G}_{\lambda_k}\big(\tfrac{\eta}{2}, \tfrac{\eta}{2}\big)d\boldsymbol{x}_k \tag{29b}$$

$$= \mathcal{N}_{\boldsymbol{y}_k}\big(h(\widehat{\boldsymbol{x}}_{k|k-1}), \lambda_k^{-1}\boldsymbol{\Sigma}_k\big)\,\mathrm{G}_{\lambda_k}\big(\tfrac{\eta}{2}, \tfrac{\eta}{2}\big) \tag{29c}$$

$$\propto \lambda_k^{\left(\frac{\eta+m}{2}-1\right)}\exp\big(-\tfrac{\lambda_k}{2}\big(\boldsymbol{\epsilon}_k^t\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\epsilon}_k + \eta\big)\big), \tag{29d}$$

$$\propto G_{\lambda_k}\big(\tfrac{1}{2}\big(\eta + m\big), \tfrac{1}{2}\big(\boldsymbol{\epsilon}_k^t\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\epsilon}_k + \eta\big)\big) \tag{29e}$$

where $\boldsymbol{\epsilon}_k = \boldsymbol{y}_k - h(\widehat{\boldsymbol{x}}_{k|k-1})$. The primary motivation behind using the certainty equivalence approximation is that the resulting posterior $f_2(\lambda_k|\boldsymbol{Y}_k)$ has the same functional form (i.e., Gamma distribution) as that of the optimal VB-posterior (refer to [2, eq (2)]). Next we apply the aforementioned proposition to determine $f_1(\boldsymbol{x_k}|\boldsymbol{Y}_k)$, i.e., $f_1(\boldsymbol{x_k}|\boldsymbol{Y}_k) \propto \exp\big(\mathrm{E}_{f_2(\boldsymbol{\lambda_k}|\boldsymbol{Y})}[\ln(p(\boldsymbol{x}_k, \lambda_k|\boldsymbol{Y}))]\big)$; we evaluate

$$\begin{aligned} &\mathrm{E}_{f_2(\boldsymbol{\lambda_k}|\boldsymbol{Y})}[\ln(p(\boldsymbol{x}_k, \lambda_k|\boldsymbol{Y}))] = \\ &-\tfrac{1}{2}(\boldsymbol{x}_k - \widehat{\boldsymbol{x}}_{k|k-1})^T\boldsymbol{P}_{k|k-1}^{-1}(\boldsymbol{x}_k - \widehat{\boldsymbol{x}}_{k|k-1}) \\ &-\tfrac{1}{2}\bar{\lambda}_k(\boldsymbol{y}_k - h(\boldsymbol{x}_k))^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{y}_k - h(\boldsymbol{x}_k)) + C, \end{aligned} \tag{30}$$

where C represents those terms which are independent of $\boldsymbol{x}_k$, and $\bar{\lambda}_k := \big(\eta + m\big)/\big(\boldsymbol{\epsilon}_k^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\epsilon}_k + \eta\big)$ denotes the mean of λ_k.

The argument (30) is quadratic in $\boldsymbol{x}_k$, and this is desirable for obtaining a closed-form solution. Further, we evaluate

$$f_1(\boldsymbol{x}_k|\boldsymbol{Y}_k, s_k^{(2)}) \propto \exp\left(\mathrm{E}_{f_2(\boldsymbol{\lambda_k}|\boldsymbol{Y})}[\ln(p(\boldsymbol{x}_k, \lambda_k|\boldsymbol{Y}))]\right) \propto \mathcal{N}_{\boldsymbol{x}_k}\left(\widehat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{P}_{k|k-1}\right)\mathcal{N}_{\boldsymbol{y}_k}\left(h(\boldsymbol{x}_k), \bar{\lambda}_k^{-1}\boldsymbol{\Sigma}_k\right). \tag{31}$$

Now using the linearization of $h(\boldsymbol{x}_k)$ as described in (16), we approximate $\mathcal{N}_{\boldsymbol{y}_k}\left(h(\boldsymbol{x}_k), \bar{\lambda}_k^{-1}\boldsymbol{\Sigma}_k\right) \approx \mathcal{N}_{\boldsymbol{y}_k}\left(\boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{c}_k, \bar{\lambda}_k^{-1}\boldsymbol{\Sigma}_k + \boldsymbol{\Omega}_k^h\right)$. Consequently,

$$f_1(\boldsymbol{x}_k|\boldsymbol{Y}_k, s_k^{(2)}) \propto \mathcal{N}_{\boldsymbol{x}_k}\left(\widehat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{P}_{k|k-1}\right)\times \mathcal{N}_{\boldsymbol{y}_k}\left(\boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{c}_k, \bar{\lambda}_k^{-1}\boldsymbol{\Sigma}_k + \boldsymbol{\Omega}_k^h\right). \tag{32}$$

Note that (32) is similar to (21) with the only difference that $\boldsymbol{R}_k$ is replaced with $\bar{\lambda}_k^{-1}\boldsymbol{\Sigma}_k$; accordingly, we get $f_1(\boldsymbol{x}_k|\boldsymbol{Y}_k, s_k^{(2)}) \approx \mathcal{N}_{\boldsymbol{x}_k}(\widehat{\boldsymbol{x}}_{k|k}^{(2)}; \boldsymbol{P}_{k|k}^{(2)})$, where

$$\widehat{\boldsymbol{x}}_{k|k}^{(2)} = \widehat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k^{(2)}(\boldsymbol{y}_k - \bar{\boldsymbol{y}}_{k|k-1}), \tag{33a}$$

$$\boldsymbol{K}_k^{(2)} = \boldsymbol{P}_k^{xh}(\boldsymbol{P}_k^{hh} + \bar{\lambda}_k^{-1}\boldsymbol{\Sigma}_k)^{-1}, \tag{33b}$$

$$\boldsymbol{P}_{k|k}^{(2)} = \boldsymbol{P}_{k|k-1} - \boldsymbol{P}_k^{xh}(\boldsymbol{P}_k^{hh} + \bar{\lambda}_k^{-1}\boldsymbol{\Sigma}_k)^{-1}(\boldsymbol{P}_k^{xh})^T. \tag{33c}$$

Remark 1: Note that if $\bar{\lambda}_k$ is known, then the variational Bayes approximation of (32) is **equivalent to approximating the likelihood as** $p(\boldsymbol{y}_k|\boldsymbol{x}_k, s_k^{(2)}) \approx \mathcal{N}_{\boldsymbol{y}_k}\left(\boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{c}_k, \bar{\lambda}_k^{-1}\boldsymbol{\Sigma}_k + \boldsymbol{\Omega}_k^h\right)$. We use this approximation in the evaluation of the weighting factor $p(s_k^{(2)}|\boldsymbol{Y}_k)$, in the next section.

Remark 2: Note that the evaluation of $\hat{\boldsymbol{x}}_{k|k-1}$, $\boldsymbol{P}_{k|k-1}$ and consequently $\hat{\boldsymbol{x}}_{k|k}^{(i)}$, $\boldsymbol{K}_k^{(i)}$ and $\boldsymbol{P}_{k|k}^{(i)}$, for $i = 1, 2$, requires the evaluation of moment integrals $\bar{\boldsymbol{x}}_{k|k-1}$, $\boldsymbol{P}_{k-1}^{ff}$, $\bar{\boldsymbol{y}}_{k|k-1}$, $\boldsymbol{P}_k^{xh}$ and $\boldsymbol{P}_k^{hh}$ using (7) (11), (17), (18) and (19), respectively. All these integrals are of the form $I(s) = \int s(\boldsymbol{x})\mathcal{N}_{\boldsymbol{x}}(\hat{\boldsymbol{x}}; \boldsymbol{P})d\boldsymbol{x}$, for some nonlinear function $s(\cdot)$. The integral $I(s)$, in general, does not admit a closed-form solution. However, a large number of numerical integration techniques have been suggested in literature to approximate such integrals [1, and references therein], [17, 18] etc. In this work, we shall employ the third degree cubature rules [17, 19], to approximate the required moment integrals. First, a change of variable is introduced to convert the non-standard Gaussian density, in the integral, into a standard one. Let $\boldsymbol{x} = \hat{\boldsymbol{x}} + \sqrt{\boldsymbol{P}}\boldsymbol{c}$, where $\boldsymbol{P} = \sqrt{\boldsymbol{P}}\sqrt{\boldsymbol{P}}^T$ [1]; then, the Gaussian weighted integral is written as $\int s(\hat{\boldsymbol{x}} + \sqrt{\boldsymbol{P}}\boldsymbol{c})\mathcal{N}_{\boldsymbol{c}}(\boldsymbol{0}, \boldsymbol{I})d\boldsymbol{c} = \int g(\boldsymbol{c})\mathcal{N}_{\boldsymbol{c}}(\boldsymbol{0}, \boldsymbol{I})d\boldsymbol{c} =: I(g)$, where $g(\boldsymbol{c}) := s(\hat{\boldsymbol{x}} + \sqrt{\boldsymbol{P}}\boldsymbol{c})$. Note that $\sqrt{\boldsymbol{P}}$ is a lower triangular matrix obtained from the Cholesky decomposition of $\boldsymbol{P}$. Now we approximate $I(g)$ using the third-degree cubature method as follows [19, eq (45)]:

$$I(g) = \int g(\boldsymbol{c})\mathcal{N}_{\boldsymbol{c}}(\boldsymbol{0}, \boldsymbol{I})d\boldsymbol{c} \approx \frac{1}{2n}\sum_{j=1}^{n}[g(\sqrt{n}\boldsymbol{e}_j) + g(-\sqrt{n}\boldsymbol{e}_j)], \tag{34}$$

where $\boldsymbol{e}_j$ is an n-dimensional unit vector in the jth-coordinate.

D. Weighting Factor

We now consider the evaluation of the weighting factor $p(s_k^{(i)}|\boldsymbol{Y}_k)$ and denote it with $\mu_k^{(i)}$; we note that

$$p(s_k^{(i)}|\boldsymbol{Y}_k) = \mu_k^{(i)} \propto p(\boldsymbol{y}_k|s_k^{(i)}, \boldsymbol{Y}_{k-1})p(s_k^{(i)}|\boldsymbol{Y}_{k-1}). \tag{35}$$

The second factor in (35), i.e., $p(s_k^{(i)}|\boldsymbol{Y}_{k-1})$ is expanded as follows:

$$p(s_k^{(i)}|\boldsymbol{Y}_{k-1}) = \sum_{j=1}^{2} p(s_k^{(i)}|s_{k-1}^{(j)}, \boldsymbol{Y}_{k-1})p(s_{k-1}^{(j)}|\boldsymbol{Y}_{k-1}) = \sum_{j=1}^{2}\pi_{ji}\mu_{k-1}^{(j)}. \tag{36}$$

Note that π_{ji} is known a priori and $\mu_{k-1}^{(j)}$ is available from the previous recursion. The first factor in (35), i.e., $p(\boldsymbol{y}_k|s_k^{(i)}, \boldsymbol{Y}_{k-1})$ can be written as

$$p(\boldsymbol{y}_k|s_k^{(i)}, \boldsymbol{Y}_{k-1}) = \int p(\boldsymbol{y}_k|\boldsymbol{x}_k, s_k^{(i)})p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1})\,d\boldsymbol{x}_k \tag{37}$$

Again, we evaluate $p(\boldsymbol{y}_k|s_k^{(i)}, \boldsymbol{Y}_{k-1})$ separately for $i = 1, 2$, in the following:

For s_k = s_1:

$$\begin{aligned} &p(\boldsymbol{y}_k|s_k^{(1)}, \boldsymbol{Y}_{k-1}) \\ &\approx \int \mathcal{N}_{\boldsymbol{y}_k}(\boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{c}_k; \boldsymbol{\Omega}_k^h + \boldsymbol{R}_k)\,\mathcal{N}_{\boldsymbol{x}_k}(\hat{\boldsymbol{x}}_{k|k-1}; \boldsymbol{P}_{k|k-1})\,d\boldsymbol{x}_k \\ &= \int \mathcal{N}_{\boldsymbol{y}_k}(\boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{c}_k; \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k + \boldsymbol{\Omega}_k^h) \\ &\qquad \mathcal{N}_{\boldsymbol{x}_k}(\hat{\boldsymbol{x}}_{k|k}; \boldsymbol{P}_{k|k})\,d\boldsymbol{x}_k \\ &= \mathcal{N}_{\boldsymbol{y}_k}(\boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{c}_k; \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k + \boldsymbol{\Omega}_k^h) = \Lambda_k^{(1)}, \end{aligned} \tag{38}$$

where the last equality in (38) is owing to GPT; also note that $\boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{c}_k = \bar{\boldsymbol{y}}_{k|k-1}$ and $\boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k + \boldsymbol{\Omega}_k^h = \boldsymbol{P}_k^{hh} + \boldsymbol{R}_k$, hence $\Lambda_k^{(1)} = \mathcal{N}_{\boldsymbol{y}_k}(\bar{\boldsymbol{y}}_{k|k-1}; \boldsymbol{P}_k^{hh} + \boldsymbol{R}_k)$.

For s_k = s_2:

Employing the approximation suggested in *Remark 1*, i.e., $p(\boldsymbol{y}_k|\boldsymbol{x}_k, s_k^{(2)}) \approx \mathcal{N}_{\boldsymbol{y}_k}\left(\boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{c}_k, \bar{\lambda}_k^{-1}\boldsymbol{\Sigma}_k + \boldsymbol{\Omega}_k^h\right)$, we can write

$$\begin{aligned} p(\boldsymbol{y}_k|s_k^{(2)}, \boldsymbol{Y}_{k-1}) &\approx \int \mathcal{N}_{\boldsymbol{y}_k}(\boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{c}_k; \boldsymbol{\Omega}_k^h + \bar{\lambda}_k^{-1}\boldsymbol{\Sigma}_k)\times \\ &\qquad \mathcal{N}_{\boldsymbol{x}_k}(\hat{\boldsymbol{x}}_{k|k-1}; \boldsymbol{P}_{k|k-1})\,d\boldsymbol{x}_k \\ &= \mathcal{N}_{\boldsymbol{y}_k}(\bar{\boldsymbol{y}}_{k|k-1}; \boldsymbol{P}_k^{hh} + \bar{\lambda}_k^{-1}\boldsymbol{\Sigma}_k) = \Lambda_k^{(2)}. \end{aligned} \tag{39}$$

Finally, an overall expression for $\mu_k^{(i)}$ may be written as

$$\mu_k^{(i)} = \frac{1}{c}\Lambda_k^{(i)}\sum_{j=1}^{2}\pi_{ji}\mu_{k-1}^{(j)}, \tag{40}$$

where $c = \sum_{l=1}^{2}\Lambda_k^{(l)}\sum_{m=1}^{2}\pi_{ml}\mu_{k-1}^{(m)}$ is the normalization constant.

TABLE I
STEPS INVOLVED IN THE PROPOSED FILTERING ALGORITHM

Initialize : $\widehat{\boldsymbol{x}}_{0|0} = \mathrm{E}[\boldsymbol{x}_0]$, $\boldsymbol{P}_{0|0} = \mathrm{E}[(\boldsymbol{x}_0 - \widehat{\boldsymbol{x}}_{0|0})^t(\boldsymbol{x}_0 - \widehat{\boldsymbol{x}}_{0|0})]$ and $\mu_k^{(i)} = \frac{1}{2}$, for $i = 1, 2$.

Predict : Find $\bar{\boldsymbol{x}}_{k|k-1}$ and $\boldsymbol{P}_{k-1}^{ff}$ using (7), and (11), set $\widehat{\boldsymbol{x}}_{k|k-1} = \bar{\boldsymbol{x}}_{k|k-1}$ and $\boldsymbol{P}_{k|k-1} = \boldsymbol{P}_{k-1}^{ff} + \boldsymbol{Q}_k$.

Update : Find $\bar{\boldsymbol{y}}_{k|k-1}, \boldsymbol{P}_k^{xh}$ and $\boldsymbol{P}_k^{hh}$, using (17), (18) and (19).
Find $\bar{\lambda}_k = \dfrac{\eta + m}{\boldsymbol{\epsilon}_k^T \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\epsilon}_k + \eta}, \boldsymbol{\epsilon}_k = \boldsymbol{y}_k - h(\widehat{\boldsymbol{x}}_{k|k-1})$, $m = \dim(\boldsymbol{y}_k)$.
Find $\boldsymbol{K}_k^{(1)}, \widehat{\boldsymbol{x}}_{k|k}^{(1)}$ and $\boldsymbol{P}_{k|k}^{(1)}$ using (22), (23) and (24).
Find $\boldsymbol{K}_k^{(2)}, \widehat{\boldsymbol{x}}_{k|k}^{(2)}$ and $\boldsymbol{P}_{k|k}^{(2)}$ using (33a), (33b) and (33c).

Averaging : Find $\Lambda_k^{(1)}$, and $\Lambda_k^{(2)}$, using (38) and (39).
Find $\mu_k^{(i)} = \dfrac{\Lambda_k^{(i)} \sum_{j=1}^{2} \pi_{ji} \mu_{k-1}^{(j)}}{\sum_{l=1}^{2} \Lambda_k^{(l)} \sum_{m=1}^{2} \pi_{ml} \mu_{k-1}^{(m)}}$, for $i = 1, 2$.

Outputs : $\widehat{\boldsymbol{x}}_{k|k} = \sum_{i=1}^{2} \widehat{\boldsymbol{x}}_{k|k}^{(i)} \mu_k^{(i)}$,
$\boldsymbol{P}_{k|k} = \sum_{i=1}^{2} \mu_k^{(i)} \{\boldsymbol{P}_{k|k}^{(i)} + (\widehat{\boldsymbol{x}}_{k|k}^{(i)} - \widehat{\boldsymbol{x}}_{k|k})(\widehat{\boldsymbol{x}}_{k|k}^{(i)} - \widehat{\boldsymbol{x}}_{k|k})^T\}$.

E. Approximation

From (3), we note that the required probability $p(\boldsymbol{x}_k|\boldsymbol{Y}_k)$ is actually a sum of two weighted densities. However, in the BMA framework, $p(\boldsymbol{x}_k|\boldsymbol{Y}_k)$ is approximated using a single Gaussian density at each instant k. Accordingly, we approximate $p(\boldsymbol{x}_k|\boldsymbol{Y}_k) \approx \mathcal{N}_{\boldsymbol{x}_k}(\hat{\boldsymbol{x}}_{k|k}; \boldsymbol{P}_{k|k})$, where $\hat{\boldsymbol{x}}_{k|k}$ and $\boldsymbol{P}_{k|k}$ are obtained by matching moments as follows:

$$\widehat{\boldsymbol{x}}_{k|k} = \sum_{i=1}^{2} \widehat{\boldsymbol{x}}_{k|k}^{(i)} \mu_k^{(i)},$$
$$\boldsymbol{P}_{k|k} = \sum_{i=1}^{2} \mu_k^{(i)} \{\boldsymbol{P}_{k|k}^{(i)} + (\widehat{\boldsymbol{x}}_{k|k}^{(i)} - \widehat{\boldsymbol{x}}_{k|k})(\widehat{\boldsymbol{x}}_{k|k}^{(i)} - \widehat{\boldsymbol{x}}_{k|k})^T\}. \tag{41}$$

This completes our derivation of the proposed filter; based on this derivation, an algorithm is summarized in Table I.

F. Computational Complexity

The asymptotic complexity of the various operations involved in the proposed algorithm is listed in Table II. Note that we have used $\mathcal{C}_f$ and $\mathcal{C}_h$ to denote the complexity of evaluating nonlinear functions $f(\cdot)$ and $h(\cdot)$, respectively. Also, $n = \dim\{\boldsymbol{x}_k\}$ and $m = \dim\{\boldsymbol{y}_k\}$. We note that, for large n, the time complexity will be dominated by $\mathcal{O}(n^3)$ operations. However, if either $\mathcal{C}_f$ or $\mathcal{C}_h$ is greater than $\mathcal{O}(n^2)$, then time complexity will chiefly depend upon function evaluations. Also note that, some operations such as $\boldsymbol{K}_k^{(i)}$, $\hat{\boldsymbol{x}}_{k|k}^{(i)}$ and $\boldsymbol{P}_{k|k}^{(i)}$ are evaluated for $i = 1, 2$ in the proposed method. Whereas, these operations are performed only once in a standard CKF. Also, the covariance mixing step in the output is owing to model averaging and is not required in standard filters. Hence, the complexity of the proposed algorithm is slightly greater than that of standard CKF. However, for large n, both scale according to either $\mathcal{O}(n^3)$. Also, in an iterative VB procedure, all of these operations (apart from covariance mixing) are performed N_{itr} times, where N_{itr} is the number of iterations required by the IVB filter to converge. Hence for $N_{\mathrm{itr}} > 2$, the proposed algorithm will always be computationally more efficient than its IVB based counterparts.

TABLE II
ASYMPTOTIC COMPUTATIONAL COMPLEXITY OF THE VARIOUS STEPS IN PROPOSED ALGORITHM.

Operations	Complexity $\mathcal{O}(\cdot)$
Sigma Points	n^3
Evaluation of $f(\cdot)$, $h(\cdot)$	$n\mathcal{C}_f$, $n\mathcal{C}_h$
$\widehat{\boldsymbol{x}}_{|k-1}$	n^2
$\bar{\boldsymbol{y}}_{k|k-1}$	nm
$\boldsymbol{P}_{k|k-1}$	n^3
$\boldsymbol{K}_k^{(i)}$	$nm^2 + m^3$
$\hat{\boldsymbol{x}}_{k|k}^{(i)}$	$n^2 m$
$\boldsymbol{P}_{k|k}^{(i)}$	$nm^2 + n^2 m$
$\boldsymbol{P}_{k|k}$	n^2

III. SIMULATION RESULTS

In this section, we the performance of the proposed algorithm is compared against a conventional CKF [17] and a robust (to outliers) CKF that utilizes iterative variational Bayes (IVB) technique to handle outliers [2, 5], using a simulation example that considers target tracking based on range and bearings measurements. We consider the cases of Gaussian-only as well as Gaussian-with-outliers observation noise models. The root-mean-square-error (RMSE) is used as a figure-of-merit to compare the performance of the various filters.

Let us consider a target that is moving with nearly constant velocity [20], i.e.,

$$\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{w}_k, \tag{42}$$

where $\boldsymbol{x}_k = [\zeta_k, \dot{\zeta}_k, \epsilon_k, \dot{\epsilon}_k]^T$, $\boldsymbol{F} = \boldsymbol{F}_1 \otimes \boldsymbol{I}_2$, $[\zeta_k, \epsilon_k]$ denote the position coordinates; whereas, $\dot{\zeta}_k$ and $\dot{\epsilon}_k$ denote velocities in ζ and ϵ directions, respectively. We have $\boldsymbol{F}_1 = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$ and the sampling time T is set to 0.5 sec. The uncertainty $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}; \boldsymbol{Q})$, where $\boldsymbol{Q} = (\boldsymbol{Q}_1 \otimes \boldsymbol{I}_2)\sigma_w^2$, $\boldsymbol{Q}_1 = \begin{bmatrix} T^4/4 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix}$ and $\sigma_w = 2$ m/s^2. The observation model is specified as

$$\boldsymbol{y}_k = \begin{bmatrix} \sqrt{\zeta_k^2 + \epsilon_k^2} \\ \tan^{-1}\left(\frac{\epsilon_k}{\zeta_k}\right) \end{bmatrix} + \boldsymbol{v}_k. \tag{43}$$

If there are no outliers in the observation noise then $\boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}; \boldsymbol{R}_k)$, where $\boldsymbol{R}_k = \mathrm{diag}\{[\sigma_r^2, \sigma_\theta^2]\}$ with $\sigma_r = \sqrt{1000}$ m and $\sigma_\theta = \sqrt{10}$ mrad for all k. To generate the effect of outliers, we use a clutter model that has been widely used in literature to simulate outliers [2, 3, 5, 6], i.e.,

$$\boldsymbol{v}_k \sim \begin{cases} \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k) & \text{with probability } 0.95 \\ \mathcal{N}(\boldsymbol{0}, 50\boldsymbol{R}_k) & \text{with probability } 0.05 \end{cases} \tag{44}$$

TABLE III
COMPARISON OF RMSE VALUES AVERAGED OVER THE ENTIRE SIMULATION TIME

	RMSE	CKF	CKF-IVB	Proposed
No Outliers	Position (m)	12.7	13.4	12.7
	Velocity (m/s)	3.6	3.7	3.6
With Outliers	Position (m)	21.5	13.9	13.2
	Velocity (m/s)	4.6	3.7	3.6

The state vector is initialized with $\widehat{\boldsymbol{x}}_{0|0} = [100, 10, 100, 5]^T$ and the initial covariance is set to $\boldsymbol{P}_{0|0} = \text{diag}\{[100, 10, 100, 10]\}$. The parameter η is set to 4 (As suggested in [2]) and the total simulation time is 1 min. The transition probabilities π_{ji} required for the proposed filter are set as follows: $\pi_{11} = 0.9$, $\pi_{12} = 0.1$, $\pi_{21} = 0.9$ and $\pi_{22} = 0.1$[2]. The simulation results are averaged over a 1000 Monte-carlo runs.

In Figure 1 and 2, we depict the position and velocity RMSE values, respectively, of the three filters when the observation noise is sampled from Gaussian distribution only, i.e., when there are no outliers present. We note that the conventional CKF filter and the proposed filter have almost similar performances in this case; however, the robust CKF-IVB filter suffers performance degradation. It is owing to the reason that a CKF-IVB filter is based on the assumption of t-distributed observation noise and hence in the case of a Gaussian distributed noise, it does not perform as well as a conventional filter. On the other hand, the proposed filter incorporates both the Gaussian as well as t-distributed noise models in a BMA framework and hence performs as good as a conventional CKF. In Figure 3 and 4, we plot the RMSE values of the various filters for outliers contaminated observation noise generated using (44). Note that, owing to the effect of outliers, the CKF filter suffers a large degradation in performance. Whereas the proposed filter as well as the CKF-IVB filter show robust performance in the presence of outliers. Also, the proposed filter is performing better than the CKF-IVB filter even in the presence of outliers. In Table III, we depict the RMSE values averaged over the entire simulation time. We note that in the absence of outliers, the proposed filter shows a 5% improvement in position RMSE over the CKF-IVB filter; whereas, it shows a 38% improvement in position RMSE, over the standard CKF, in the presence of outliers.

Finally, to compare the computational costs of these methods, we run these methods 1000 times under the same conditions on MATLAB 2015a using a core-i7 2.6GHZ CPU with 16GB RAM. The average computational times for a single iteration are given in Table IV. Note that the proposed filter is approximately 2.5 times faster than the robust CKF filter based on IVB method; whereas, it takes about twice the computational time as that required by a standard CKF.

[2]The values of π_{ji} essentially model the fact that outliers occur only infrequently. We have set π_{ji} such that, irrespective of the previous noise sample, the probability that the next noise sample comes from a Gaussian distribution is 90%.

TABLE IV
COMPARISON OF AVERAGE COMPUTATIONAL TIME FOR ONE ITERATION OF EACH METHOD

Filter	CKF	CFK-IVB	Proposed
Time (msec)	25	109.2	43.7

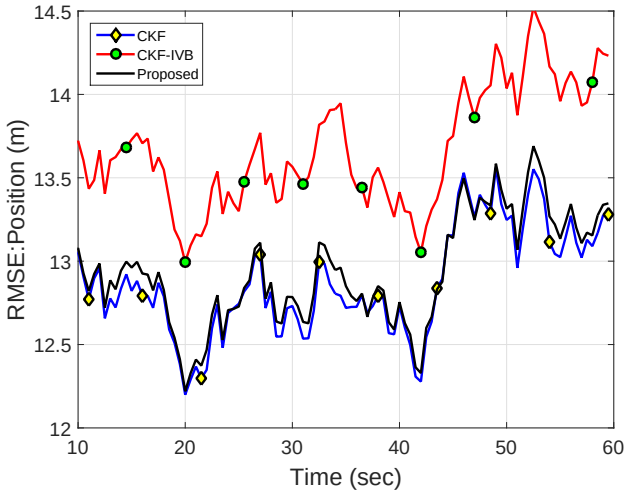

Fig. 1. Position RMSE values of CKF, CKF-IVB and the proposed filter, when there are no outliers.

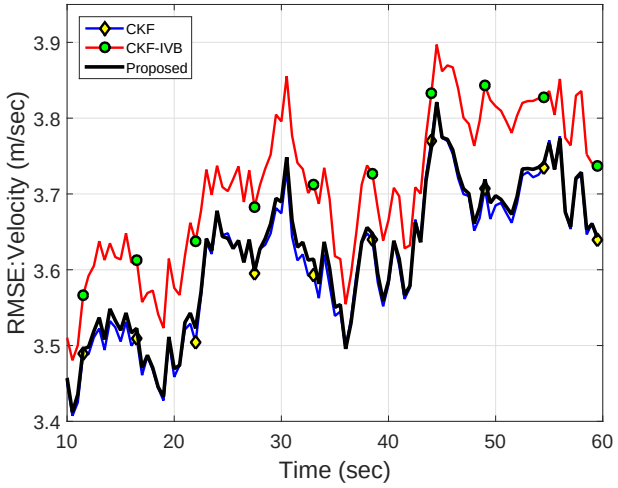

Fig. 2. Velocity RMSE values of CKF, CKF-IVB and the proposed filter, when there are no outliers.

IV. CONCLUSIONS

This paper proposes a novel robust-to-outliers Bayesian filtering approach that adopts both Gaussian and t-distributed densities to model the outliers contaminated observation noise. The proposed solution combines these models within the Bayesian Model Averaging framework, where the t-distribution is handled using a restricted variational Bayes approach. It was shown that, in the absence of outliers, the standard iterative variational Bayes (IVB) algorithm does not perform well. However, the proposed filter gives accurate estimates, comparable with the conventional Gaussian

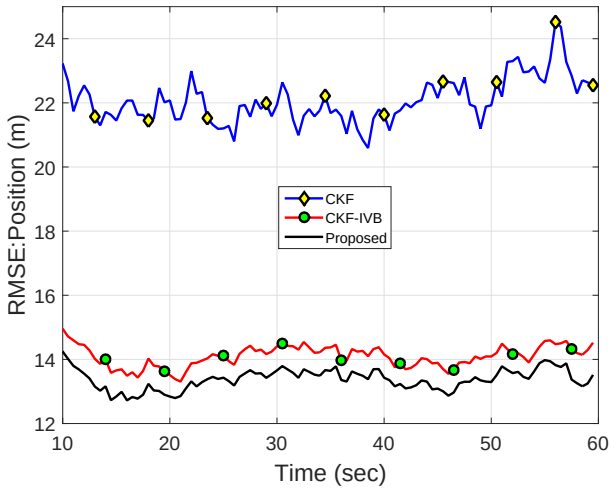

Fig. 3. Position RMSE values of CKF, CKF-IVB and the proposed filter, in the presence of outliers.

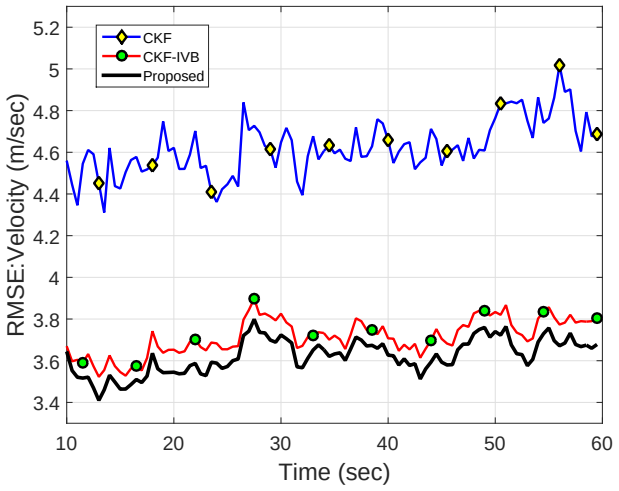

Fig. 4. Velocity RMSE values of CKF, CKF-IVB and the proposed filter, in the presence of outliers.

assumption (GA) cubature Kalman filter. In the presence of outliers, the proposed algorithm outperformed both the GA filter as well as the robust IVB filter. Moreover, from the perspective of computational cost, the proposed algorithm was found to be approximately 2.5 times more efficient than the standard IVB based robust solution. Consequently, the proposed filter appears to be an admissible substitute for its traditional counterparts.

ACKNOWLEDGMENT

This work was supported in part by funding from the Higher Education Commission (HEC), Government of Pakistan.

REFERENCES

[1] A. J. Haug, *Bayesian estimation and tracking: a practical guide*. John Wiley & Sons, 2012.

[2] R. Piché, S. Särkkä, and J. Hartikainen, "Recursive outlier-robust filtering and smoothing for nonlinear systems using the multivariate student-t distribution," in *Proc. from the IEEE International Workshp on Machine Learning for Signal Processing (MLSP)*. IEEE, 2012, pp. 1–6.

[3] S. S. Khalid, N. U. Rehman, and S. Abrar, "Robust stochastic integration filtering for nonlinear systems under multivariate t-distributed uncertainties," *Signal Processing*, vol. 140, pp. 53–59, 2017.

[4] M. Roth, E. Özkan, and F. Gustafsson, "A Student t-filter for heavy tailed process and measurement noise," in *IEEE Intl. Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 5770–5774.

[5] Y. Huang, Y. Zhang, N. Li, and J. Chambers, "A robust Gaussian approximate filter for nonlinear systems with heavy tailed measurement noises," in *Proc. from the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4209–4213.

[6] ——, "A robust Gaussian approximate fixed-interval smoother for nonlinear systems with heavy-tailed process and measurement noises," *IEEE Signal Processing Letters*, vol. 23, no. 4, pp. 468–472, 2016.

[7] B. Liu, "Robust particle filter by dynamic averaging of multiple noise models," in *Proc. from the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4034–4038.

[8] Y. Huang and Y. Zhang, "Robust students t-based stochastic cubature filter for nonlinear systems with heavy-tailed process and measurement noises," *IEEE Access*, vol. 5, pp. 7964–7974, 2017.

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

[10] A. Kiring, "Tracking in wireless sensor networks with correlated and sparse measurements," Ph.D. dissertation, Department of Automatic Control and Systems Engineering, The University of Sheffield, Western Bank, Sheffield, S10 2TN, 2017.

[11] S. Kotz and S. Nadarajah, *Multivariate t-distributions and their applications*. Cambridge University Press, 2004.

[12] I. Arasaratnam, S. Haykin, and R. J. Elliott, "Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 953–977, 2007.

[13] Á. F. García-Fernández, L. Svensson, M. R. Morelande, and S. Särkkä, "Posterior linearization filter: Principles and implementation using sigma points," *IEEE transactions on signal processing*, vol. 63, no. 20, pp. 5561–5573, 2015.

[14] S. Challa, *Fundamentals of object tracking*. Cambridge University Press, 2011.

[15] V. Šmídl and A. Quinn, *The variational Bayes method in signal processing*. Springer Science & Business Media, 2006.

[16] G. C. Goodwin and K. S. Sin, *Adaptive filtering prediction and control*. Courier Corporation, 2014.

[17] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Transactions on automatic control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[18] Y. Wu, D. Hu, M. Wu, and X. Hu, "A numerical-integration perspective on Gaussian filters," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 2910–2921, 2006.

[19] B. Jia, M. Xin, and Y. Cheng, "High-degree cubature Kalman filter," *Automatica*, vol. 49, no. 2, pp. 510–518, 2013.

[20] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.