Efficient Gaussian Mixture Filter for Hybrid Positioning

Simo Ali-Löytty
Department of Mathematics
Tampere University of Technology
Finland

Abstract—This paper presents a new way to apply Gaussian Mixture Filter (GMF) to hybrid positioning. The idea of this new GMF (Efficient Gaussian Mixture Filter, EGMF) is to split the state space into pieces using parallel planes and approximate posterior in every piece as Gaussian. EGMF outperforms the traditional single-component positioning filters, for example the Extended Kalman Filter and the Unscented Kalman Filter, in nonlinear hybrid positioning. Furthermore, EGMF has some advantages with respect to other GMF variants, for example EGMF gives the same or better performance than the Sigma Point Gaussian Mixture (SPGM) [1] with a smaller number of mixture components, i.e., smaller computational and memory requirements. If we consider only one time step, EGMF gives optimal results in the linear case, in the sense of mean and covariance, whereas other GMFs give suboptimal results.

I. INTRODUCTION

Positioning filters, such as GMF [2]–[4], are used to compute an estimate of the state using current and past measurement data. Usually, the mean of the posterior distribution is this estimate. A consistent filter also provides correct information on the accuracy of its state estimate, e.g., in the form of an estimated error covariance. Generally, GMF is a filter whose approximate prior and posterior densities are Gaussian Mixtures (GMs), a linear combination of Gaussian densities where weights are between 0 and 1. GMF is an extension of Kalman type filters. In particular, The Extended Kalman Filter (EKF) [5]–[8], Second Order Extended Kalman Filter (EKF2) [5], [6], [9], Unscented Kalman Filters (UKF) [10] and a bank of these filters, are special cases of GMF.

Hybrid positioning means that measurements used in positioning come from many different sources e.g. Global Navigation Satellite System, Inertial Measurement Unit, or local wireless networks such as a cellular network. Range, pseudo-range, delta range, altitude, base station sector and compass measurements are examples of typical measurements in hybrid positioning. In the hybrid positioning case, it is usual that measurements are nonlinear and because of that posterior density may have multiple peaks (multiple positioning solutions). In these cases, traditional single-component positioning filters, such as EKF, do not give good performance [9]. This is the reason for developing GMF for hybrid positioning [1]. Other possibility is to use a general nonlinear Bayesian filter, which is usually implemented as a particle filter or a point mass filter. These filters usually work correctly and give good positioning accuracy but require much computation time and memory.

An outline of the paper is as follows. In Section II, we glance at Bayesian filtering. In Section III, we study the basics of the GM and of the GMF. In Section IV, we present the new method, box GM approximation, to approximate Gaussian as GM. In Section V we apply the box GM approximation to the filtering framework and get the Box Gaussian Mixture Filter (BGMF). In Section V we also present Sigma Point Gaussian Mixture Filter (SPGMF) [1]. In Section VI, we develop BGMF so that it gives exact mean and covariance in one step linear case. We call that new filter the Efficient Gaussian Mixture Filter (EGMF). In Section VII, we compute one step comparison of EKF, SPGMF, BGMF and EGMF. Finally in Section VIII, we present simulation results where we compare different GMFs and a bootstrap particle filter [11].

II. BAYESIAN FILTERING

We consider the discrete-time non-linear non-Gaussian system

\[ x_k = f_{k-1}(x_{k-1}) + w_{k-1}, \]
\[ y_k = h_k(x_k) + v_k, \]

where the vectors \( x_k \in \mathbb{R}^n \) and \( y_k \in \mathbb{R}^m \) represent the state of the system and the measurement at time \( t_k, k \in \mathbb{N} \), respectively. We assume that errors \( w_{k} \) and \( v_{k} \) are white, mutually independent and independent of the initial state \( x_0 \). We denote the density functions of \( w_k \) and \( v_k \) by \( p_{w_k} \) and \( p_{v_k} \), respectively. The aim of filtering is to find the conditional probability density function (posterior)

\[ p(x_k | y_{1:k}), \]

where \( y_{1:k} \equiv y_1, \ldots, y_k \) are past and current measurements. The posterior can be determined recursively according to the following relations.

**Prediction (prior):**

\[ p(x_k | y_{1:k-1}) = \int p(x_k | x_{k-1})p(x_{k-1} | y_{1:k-1}) \, dx_{k-1}; \]

**Update (posterior):**

\[ p(x_k | y_{1:k}) = \frac{p(y_k | x_k)p(x_k | y_{1:k-1})}{\int p(y_k | x_k)p(x_k | y_{1:k-1}) \, dx_k}, \]

where the transition pdf is

\[ p(x_k | x_{k-1}) = p_{w_{k-1}}(x_k - f_{k-1}(x_{k-1})) \]
and the likelihood
\[ p(y_k|x_k) = p_{ak}(y_k - h_k(x_k)). \]
The initial condition for the recursion is given by the pdf of
the initial state \( p(x_0|y_{1:0}) = p(x_0) \). Knowledge of the posterior
distribution (4) enables one to compute an optimal state estimate
with respect to any criterion. For example, the minimum
mean-square error (MMSE) estimate is the conditional mean of
\( x_k \) [5], [12]. In general and in our case, the conditional
probability density function cannot be determined analytically.

III. GAUSSIAN MIXTURE FILTER

A. Gaussian Mixture

Definition 1 (Gaussian Mixture): an \( n \)-dimensional random
variable \( x \) is \( N \)-component Gaussian Mixture (GM) if its
characteristic function has the form
\[ \varphi_x(t) = \sum_{j=1}^{N} \alpha_j \exp \left( \frac{1}{2} t^T \Sigma_j t \right), \tag{5} \]
where \( \alpha_j \in \mathbb{R}^n, \Sigma_j \in \mathbb{R}^{n \times n} \) is symmetric positive semidefinite
(\( \Sigma_j \geq 0 \)), \( \alpha_j \geq 0 \) and \( \sum_{j=1}^{N} \alpha_j = 1 \). We use the abbreviation
\( x \sim M(\alpha_j, \mu_j, \Sigma_j) \).

If random variable \( x \) is GM, \( x \sim M(\alpha_j, \mu_j, \Sigma_j) \) (j) and
all matrices \( \Sigma_j \) are symmetric positive definite (\( \Sigma_j > 0 \)), then \( x \)
has a density function
\[ p_x(\xi) = \sum_{j=1}^{N} \alpha_j N_{\Sigma_j}^{\mu_j}(\xi), \tag{6} \]
where \( N_{\Sigma_j}^{\mu_j}(\xi) \) is the Gaussian density function with mean \( \mu_j \)
and covariance \( \Sigma_j \).

\[ N_{\Sigma_j}^{\mu_j}(\xi) = \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{\det \Sigma_j}} e^{-\frac{1}{2} (\xi - \mu_j)^T \Sigma_j^{-1} (\xi - \mu_j)}. \]

Theorem 2: Let \( x \sim M(\alpha_j, \mu_j, \Sigma_j) \). Then the mean of \( x \)
is
\[ E(x) = \sum_{j=1}^{N} \alpha_j \mu_j, \]
and the covariance of \( x \) is
\[ V(x) = \sum_{j=1}^{N} \alpha_j (\Sigma_j + (\mu_j - E(x))(\mu_j - E(x))^T). \]

Proof: Using the properties of characteristic function [13],
we get
\[ E(x) = \frac{1}{i} \text{Im}(\varphi_x'(t)|_{t=0}) = \sum_{j=1}^{N} \alpha_j \mu_j, \]
and
\[ V(x) = -\varphi_x''(t)|_{t=0} - E(x) E(x)^T \]
\[ = \sum_{j=1}^{N} \alpha_j (\mu_j \mu_j^T + \Sigma_j) - E(x) E(x)^T \]
\[ = \sum_{j=1}^{N} \alpha_j (\Sigma_j + (\mu_j - E(x))(\mu_j - E(x))^T). \]

Algorithm 1 Linearized GMF

for \( k = 1 \) to \( n \) do

1) Prediction step, prior at time \( t_k \) (see Thm. 3):
\[ M(\alpha_j, x_j^+, P_j^+), \]
where
\[ \alpha_j^+ = \frac{\alpha_j^0}{\alpha_j^0 + h_k(x_j^l_k) + \alpha_j^0 H_jk \Sigma_j H_k^T} \]
\[ x_j^+ = x_j^l_k + K_jk (y_k - h_k(x_j^l_k) - H_jk(x_j^l_k - \bar{x}_j,k)) \]
\[ P_j^+ = (I - K_jk H_jk) P_j^l_k \]
\[ K_jk = \frac{H_jk P_j^l_k H_k^T + R_k}{H_jk P_j^l_k H_k^T + R_k} \]

Here \( H_jk = \frac{\partial h_j}{\partial x} |_{x = \bar{x}_{j,k}} \) and \( \bar{x}_{j,k} \) are selected
linearization points, e.g. in EKF \( \bar{x}_{j,k} = \bar{x}_j,k \).

4) Reduce number of components: forgetting, merging and resampling [2], [15], [16].

end for

Algorithm 1 presents one version of GMF. Linearized GMF:
Algorithm 1 uses the following assumptions:
1) Initial state \( x_0 \) is non-singular GM, which means that \( x_0 \) has density function (6).

2) State model (1) is linear

\[
x_k = F_{k-1}x_{k-1} + w_{k-1}.
\]

3) Errors \( w_k \sim N(0, Q_k) \) and \( v_k \sim N(0, R_k) \) in (6) are non-singular.

Note that it is straightforward to extend Linearized GMF to cases where also the errors \( w_k \) and \( v_k \) are GMs.

IV. APPROXIMATE GAUSSIAN AS GAUSSIAN MIXTURE

As we know, EKF in hybrid positioning [9] has a consistency problem. The key reason for inconsistency is nonlinearity. Now we assume that we have only one measurement and our prior is Gaussian \( x \sim N(\hat{x}, P) \). The application to the general case is given in Section V.

One measure of nonlinearity is [1], [6], [9]

\[
Nonlinearity = \sqrt{\frac{\text{tr}(H_{e}P + H_{e}^T)}{R}} - 1, \tag{7}
\]

where \( H_{e} \) is Hessian matrix of scalar measurement \( b(x) \), \( R \) is a covariance of measurement error and \( P \) is a covariance of the state component. One possibility to overcome the nonlinearity problem (i.e. minimize Nonlinearity (7)) is to approximate Gaussian as GM whose components have smaller covariance matrices than the orginal Gaussian. One method for doing so is the Sigma Point Gaussian Mixture (SPGM) [1] (see Section IV-A). One drawback of SPGM is that SPGM splits one Gaussian to \( 2n_x + 1 \) components GM, regardless of our measurement equation and Hessian matrix \( H_{e} \). Because of this, we present a new method of approximating Gaussian as GM (see Section IV-B). We call this method as a Box GM, because it has connection to the "Box"-method [17].

A. Sigma Point GM approximation

The SPGM is given on Table I. SPGM has the same mean, covariance, and third moments as the original Gaussian distribution \( x \sim N(\hat{x}, P) \) [1].

<table>
<thead>
<tr>
<th>Index ( j )</th>
<th>( \alpha_j )</th>
<th>( \mu_j )</th>
<th>( \Sigma_j )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 0 )</td>
<td>( \frac{\hat{x}}{\sqrt{\kappa + \mu_j \sqrt{\kappa + \mu_j}} + \sqrt{\kappa + \mu_j \sqrt{\kappa + \mu_j}}} )</td>
<td>( \hat{x} )</td>
<td>(1 - ( \tau )^2) ( P )</td>
</tr>
<tr>
<td>( 1, \ldots, n_x )</td>
<td>( \frac{\hat{x} \tau + \hat{x} \tau + \sqrt{\kappa + \mu_j \sqrt{\kappa + \mu_j}} \tau}{\sqrt{\kappa + \mu_j \sqrt{\kappa + \mu_j}} + \sqrt{\kappa + \mu_j \sqrt{\kappa + \mu_j}}} )</td>
<td>( \hat{x} )</td>
<td>(1 - ( \tau )^2) ( P )</td>
</tr>
<tr>
<td>( n_x + 1, \ldots, 2n_x )</td>
<td>( \frac{\hat{x} \tau + \hat{x} \tau + \sqrt{\kappa + \mu_j \sqrt{\kappa + \mu_j}} \tau}{\sqrt{\kappa + \mu_j \sqrt{\kappa + \mu_j}} + \sqrt{\kappa + \mu_j \sqrt{\kappa + \mu_j}}} )</td>
<td>( \hat{x} )</td>
<td>(1 - ( \tau )^2) ( P )</td>
</tr>
</tbody>
</table>

B. Box GM approximation

The idea of the Box GM approximation is that we split the state space using parallel planes and approximate the Gaussian inside every piece with one GM component using moment matching method. The Box GM approximation of the Gaussian \( x \sim N(\hat{x}, P) \), with \( P > 0 \), is

\[
A_j = \{ x | l_j < x^T (x - \hat{x}) \leq l_{j+1} \}, \quad \text{where} \quad a^T P a = 1 \quad \text{and} \quad l_i \quad \text{is monotonic increasing such that} \quad l_1 = -\infty \quad \text{and} \quad l_{\text{max}} = \infty \quad \text{so these sets constitute a partition of} \quad \mathbb{R}^{n_x}.
\]

1) Mean and covariance of the Box GM approximation: In this Section, we compute the mean and the covariance of the GM approximation \( x_N \) Eq. (8). First of all, because \( \Sigma_j > 0 \forall j \) and

\[
\alpha_j = \int_{A_j} \rho(x) \, dx = \Phi(l_j) - \Phi(l_{j-1}),
\]

\[
\mu_j = \int_{A_j} \xi \rho(x) \, dx = \hat{x} + P a_{ej},
\]

\[
\Sigma_j = \int_{A_j} (\xi - \mu_j)^T P \rho(x) (\xi - \mu_j) \, dx = \Phi(l_j) - \Phi(l_{j-1})^T P a_{ej} (\Phi(l_j) - \Phi(l_{j-1})) + \epsilon_j^2 \Phi(l_j) - \Phi(l_{j-1}),
\]

where \( \Phi \) is the standard normal cumulative density function and sets \( A_j \) have the following form

\[
A_j = \{ x | l_j < x^T (x - \hat{x}) \leq l_{j+1} \},
\]

\[
\text{where} \quad a^T P a = 1 \quad \text{and} \quad \text{vector} \quad l \quad \text{is monotonic increasing such that} \quad l_1 = -\infty \quad \text{and} \quad l_{\text{max}} = \infty \quad \text{so these sets constitute a partition of} \quad \mathbb{R}^{n_x}.
\]

2) Contour plot of the Box GM approximation: In Fig. 1 we compare the density function of the Gaussian distribution

\[
x \sim N \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 13 & -12 \\ -12 & 13 \end{bmatrix} \right). \tag{9}
\]

and the density function of its approximation by a Box GM with parameters \( \alpha \approx \begin{bmatrix} 0.2774 \\ 0 \end{bmatrix} \) and

\[
l = \begin{bmatrix} -\infty & -1.28 & 1.28 & \infty \end{bmatrix} \approx \Phi^{-1}(\begin{bmatrix} 0.0 \ 0.0 \ 0.9 \ 1 \end{bmatrix}).
\]

Fig. 1 shows the contour plots of the Gaussian and the Box GM density functions so that 50% of probability is inside the
innermost curve and 95% of probability is inside the outermost curve. We see that the density function of the Box GM is quite good approximation of the Gaussian density function Eq. (9).

V. BOX GAUSSIAN MIXTURE FILTER AND SIGMA POINT GAUSSIAN MIXTURE FILTER

Box Gaussian Mixture Filter (BGMF) is a straightforward application of the Box GM approximation to the Gaussian Mixture filtering framework. BGMF is a Linearized GMF (see Alg. 1), where the step 2 is the following. First we compute Nonlinearity (7) statistics for every component and measurement. We select the components that have at least one highly nonlinear measurement with Nonlinearity > 0. Every selected component we replace by its Box GM approximation (Section IV-B).

The SPGMF, which is presented in the paper [1] with name GMF$^d$, is also a Linearized GMF (see Alg. 1) and almost the same as BGMF. Only difference between SPGMF and BGMF is that SPGMF approximates selected Gaussians using SPGM approximation (see Table I), while BGMF uses the Box GM approximation (Section IV-B).

VI. EFFICIENT GAUSSIAN MIXTURE FILTER

In this Section, we derive a new GMF, Efficient Gaussian Mixture Filter (EGMF). Prediction step (Eq. (3)) of EGMF is the same as prediction step of Linearized GMF (see Alg. 1). Now we consider update step (Eq. (4)). Assume that prior distribution is a GM

$$x \sim \text{M}(\beta^i, \hat{x}^i, P^i)_i$$

and measurement model is (see (2))

$$y = h(x) + v,$$

where $v \sim N(0,R)$. Now the posterior density function is

$$p(x|y) \propto \sum_{i=1}^{n_{\text{prior}}} \beta^i N_{P^i}(x) N^0_R(y - h(x))$$

$$= \sum_{i=1}^{n_{\text{prior}}} \beta^i \sum_{j=1}^{n_i} \chi_{A^i_j}(x) N_{P^i_j}(x) N^0_R(y - h(x)),$$

where $\alpha^T P^a T^a = 1 \forall i$, $A^i_j = \{x \mid l^i_j < \alpha^T (x - \hat{x}^i) \leq l^i_{j+1}\}$, vectors $l^i$ are monotonic increasing so that $l^i_1 = -\infty$ and $l^i_{n_i+1} = \infty$. So for all $i$ sets $A^i_j$ constitute a partition of $\mathbb{R}^n$.

BGMF (see Section V) approximates

$$\chi_{A^i_j}(x) N_{P^i_j}(x) \approx \alpha^i_j N_{\mu^i_j}(x) \text{ and}$$

$$N^0_R(y - h(x)) \approx N^0_R(y - h(\mu^i_j) - H_{\mu^i_j}(x - \mu^i_j))$$

where $\alpha^i_j$, $\mu^i_j$ and $\Sigma^i_j$ are computed using the Box GM algorithm (see Section IV-B) and $H_{\mu^i_j} = h'(\mu^i_j)$. So BGMF approximates both prior and likelihood before multiplying them. EGMF first approximates the likelihood as

$$N^0_R(y - h(x)) \approx N^0_R(y - h(\mu^i_j) - H_{\mu^i_j}(x - \mu^i_j))$$

and then multiplies it with the prior:

$$p(x|y) \propto \sum_{i=1}^{n_{\text{prior}}} \beta^i \sum_{j=1}^{n_i} \chi_{A^i_j}(x) N_{P^i_j}(x) N^0_R(y - h(\mu^i_j) - H_{\mu^i_j}(x - \mu^i_j))$$

$$= \sum_{i=1}^{n_{\text{prior}}} \beta^i \sum_{j=1}^{n_i} \chi_{A^i_j}(x) \gamma^i_j N_{P^i_j}(x)$$

(10)

where

$$\gamma^i_j = N_{H_{\mu^i_j} + H_{\mu^i_j}(\hat{x}^i - \mu^i_j)}(y)$$

$$\hat{x}^i = \hat{x} + K^i_j(y - h(\mu^i_j) - H_{\mu^i_j}(\hat{x}^i - \mu^i_j))$$

$$P^i_j = (I - K^i_jH_{\mu^i_j})P^i_j$$

$$K^i_j = P^i_jH_{\mu^i_j}(I + H_{\mu^i_j}P^i_jH_{\mu^i_j})^{-1}.$$
EGMF posterior, when prior distribution is posterior, SPGMF (Section V) posterior, BGMF posterior and example. This example presents a comparison between EKF included BGMF (Section V) and EGMF (Section VI) to this and we get one base station range measurement (2), where

\[ d = \sqrt{x^2 + y^2}. \]

Now we get that the posterior of EGMF is

\[ p(x|y) \approx \frac{\sum_{j=1}^{n_{max}} \sum_{i=1}^{n_{i}} a_j^i b^i \gamma_j^i \mathcal{N}(x)}{\sum_{j=1}^{n_{max}} \sum_{i=1}^{n_{i}} a_j^i b^i \gamma_j^i}. \]

If we consider the linear case and only one time step, EGMF gives a correct mean and a correct covariance, because linearity ensures that \* approximation in Eq. (10) is exact and the Box GM approximation maintains the mean and the covariance in Eq. (11) (see Section IV-B1). SPGMF and BGMF can give a wrong mean and a wrong covariance in one time step linear case.

**VII. ONE STEP COMPARISON OF EKF, SPGMF, BGMF AND EGMF**

This is the same example as in paper [1], but we have included BGMF (Section V) and EGMF (Section VI) to this example. This example presents a comparison between EKF posterior, SPGMF (Section V) posterior, BGMF posterior and EGMF posterior, when prior distribution is

\[ x \sim N \left( \begin{bmatrix} d \\ 0 \end{bmatrix}, \begin{bmatrix} 100^2 & 0 \\ 0 & 300^2 \end{bmatrix} \right), \]

and we get one base station range measurement (2), where

\[ y = 1000, \quad h(x) = \|x\| \quad \text{and} \quad v \sim N(0, 100^2). \]

The base station is located in origin and so \( d \) is the distance between prior mean and the base station. "Exact" posterior density function \( p_{\text{exact}}(x) \) is computed using a point-mass filter, with \( 500^2 \) points [18]. Approximation of the exact density function is computed using EKF, SPGMF, BGMF and EGMF. The SPGMF uses parameters \( \kappa = 4 \) and \( \tau = \frac{1}{2} \) and the BGMF and EGMF uses parameters \( a = \left[ \begin{array}{c} 0 \\ \frac{1}{2} \end{array} \right] \) and

\[ l = \left[ \begin{array}{ccc} -\infty & -1.28 & 1.28 \\ -1.28 & 28 & \infty \\ 1.28 & 28 & \infty \end{array} \right] \]

Note that both BGMF and EGMF have only three GM components whereas SPGMF has five GM components.

Comparison between EKF, SPGMF, BGMF and EGMF contains two parts. First we compute the Mahalanobis distance between the mean of the exact posterior mean and means of approximations

\[ \sqrt{(\mu_{\text{exact}} - \mu_{\text{app}})^T \Sigma_{\text{exact}}^{-1} (\mu_{\text{exact}} - \mu_{\text{app}})}. \]

These results are shown in Fig. 2. The value of the **Nonlinearity** function (7) is also plotted in Fig. 2. We see that Mahalanobis distance between EKF mean and exact mean increases rapidly when nonlinearity becomes more significant. In that case SPGMF, BGMF and EGMF give much better results than EKF. EGMF has the same or smaller Mahalanobis distance than BGMF. Furthermore, SPGMF, BGMF and EGMF give always as good results as EKF even when there is no significant nonlinearity.

Second, we compute the first order Lissack-Fu distance

\[ \int |p_{\text{exact}}(x) - p_{\text{app}}(x)| dx, \]

between exact posterior and the approximations (also called a total variation norm). These results are in Fig. 3. The value of the **Nonlinearity** function (7) is also plotted. We see that SPGMF, BGMF and EGMF give smaller Lissack-Fu distance than EKF. Difference between EGMF Lissack-Fu distance and EKF Lissack-Fu distance increases when nonlinearity becomes more significant. EKF Lissack-Fu distance is almost 2 (maximum value) when \( d = 100 \), so the exact posterior and the EKF posterior approximation are almost totally separate.
Furthermore, EGMF has Lissack-Fu distance same as or smaller than BGMF.

Overall SPGMF, BGMF and EGMF work almost identically although EGMF and BGMF use only three mixture component versus five SPGMF mixture components. In this example, EGMF gives the best results compared to the other filters. Furthermore SPGMF, BGMF and EGMF all give much better results than EKF when nonlinearity is significant.

VIII. Simulations

In the simulations, we use the position-velocity model, so the state $x = \begin{bmatrix} r_u \\ v_u \end{bmatrix}$ consists of user position vector $r_u$ and user velocity vector $v_u$, which are in East-North-Up (ENU) coordinate system. In this model the user velocity is a random walk process [19]. Now the state-dynamic (1) is

$$x_k = \Phi_{k-1} x_{k-1} + w_{k-1},$$

where

$$\Phi_{k-1} = \begin{bmatrix} 1 & \Delta t_{k} I \\ 0 & 1 \end{bmatrix},$$

$\Delta t_{k} = t_k - t_{k-1},$ and $w_{k-1}$ is white, zero mean and Gaussian noise, with covariance

$$Q_{k-1} = \begin{bmatrix} \frac{\Delta t_k^4 \sigma_s^2}{4} & 0 & \frac{\Delta t_k^2 \sigma_s^2}{2} & 0 \\ 0 & \frac{\Delta t_k^4 \sigma_s^2}{4} & 0 & \frac{\Delta t_k^2 \sigma_s^2}{2} \\ \frac{\Delta t_k^4 \sigma_s^2}{4} & 0 & \frac{\Delta t_k^2 \sigma_s^2}{2} & 0 \\ 0 & \frac{\Delta t_k^4 \sigma_s^2}{4} & 0 & \frac{\Delta t_k^2 \sigma_s^2}{2} \end{bmatrix},$$

where $\sigma_s^2 = 2 \frac{\sigma_r^2}{\sigma_v^2}$ represents the velocity errors on the East-North plane and $\sigma_u^2 = 0.01 \frac{\sigma_r^2}{\sigma_v^2}$ represents the velocity errors in the vertical direction. [9], [20]

In our simulations, we use base station range measurements, altitude measurements, satellite pseudorange measurements and satellite delta range measurements (see Eq. (2)).

$$y^b = \|r_b - r_u\| + \epsilon_b,$$

$$y^p = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} r_u + \epsilon_a,$$

$$y^s = \begin{bmatrix} 1 & -1 & 0 \end{bmatrix} + b + \epsilon_v,$$

$$y^a = \begin{bmatrix} r_u - r_a \end{bmatrix}^T \left( \frac{r_k - r_u}{\|r_k - r_u\|} \right) + \hat{b} + \epsilon_v,$$

where $r_b$ is a base station position vector, $r_u$ is a satellite position vector, $b$ is clock bias, $\epsilon_a$ is a satellite velocity vector, $b$ is clock drift and $\epsilon_v$ are error terms. We use satellite measurements only when there is more than one satellite measurement available, so that bias can be eliminated. These are the same measurements equations as in the papers [11], [15], [17].

Simulations are made using Personal Navigation Filter Framework (PNaFF) [21]. PNaFF is a comprehensive simulation and filtering test bench that we are developing and using in the Personal Positioning Algorithms Research Group. PNaFF uses Earth Centered Earth Fixed (ECEF) coordinate system so we have converted our models from ENU to ECEF.

Table II: A summary of 200 different simulations with base station measurements.

<table>
<thead>
<tr>
<th>Solver</th>
<th>Time</th>
<th>Err. rms</th>
<th>95%</th>
<th>Ref</th>
<th>Inc. %</th>
</tr>
</thead>
<tbody>
<tr>
<td>EKF</td>
<td>10</td>
<td>236</td>
<td>465</td>
<td>83</td>
<td>6.6</td>
</tr>
<tr>
<td>EKF2</td>
<td>11</td>
<td>214</td>
<td>433</td>
<td>65</td>
<td>2.8</td>
</tr>
<tr>
<td>UKF</td>
<td>23</td>
<td>213</td>
<td>421</td>
<td>58</td>
<td>2.3</td>
</tr>
<tr>
<td>SPGMF</td>
<td>62</td>
<td>210</td>
<td>399</td>
<td>69</td>
<td>3.3</td>
</tr>
<tr>
<td>PF_2500</td>
<td>38</td>
<td>201</td>
<td>397</td>
<td>73</td>
<td>21.9</td>
</tr>
<tr>
<td>BGMF</td>
<td>32</td>
<td>194</td>
<td>371</td>
<td>58</td>
<td>2.8</td>
</tr>
<tr>
<td>EGMF</td>
<td>32</td>
<td>191</td>
<td>360</td>
<td>57</td>
<td>2.8</td>
</tr>
<tr>
<td>Ref</td>
<td>$\infty$</td>
<td>155</td>
<td>287</td>
<td>0</td>
<td>0.1</td>
</tr>
</tbody>
</table>

A. Summary of base station cases

On Table II, we have listed a summary of two hundred 120 second simulations, which use only base station measurements. This means that simulations use base station range measurements with variance $(80 \text{ m})^2$, very inaccurate altitude measurements with variance $(300 \text{ m})^2$ and restrictive information. Restrictive information is in our case base station $120^\circ$-sector and maximum range information. So when we have restrictive information we know that user is inside the particular area, which is restricted using sector and maximum range information. Restrictive information are used in the same way as in paper [17]. Summary consist of following columns: Time is computation time using Matlab in our implementation, scaled so that computation time of EKF is 10. This gives a rough idea of the relative time complexity of each algorithm. Err. rms is 3D root mean square position error. Err. 95% gives a radius containing 95 % of the 3D errors. Err. ref. is 3D error to reference posterior mean, which is computed using a particle filter with systematic resampling and $10^6$ particles [11]. Inc. % is a percentage of time where the filter is inconsistent with respect to the general inconsistency test, with risk level 5% [9].

Solver are sorted so that rms errors are in descending order. PF\_N indicates particle filter with systematic resampling and $N$ particles [11], so reference solution is the same as PF\_100. On these simulations BGMF splits, when highly nonlinearity exists, one GM component into four components with equal weights. In that case the parameter $l$ of the Box GM approximation (see Section IV-B) is

$$l = \Phi^{-1} \left( \begin{bmatrix} 0 & 1/4 & 1/4 & 3/4 & 1 \end{bmatrix} \right).$$

EGMF also uses the same $l$ parameter (see Section VI).

In Table II, the results are the realization of random variables and if we run these simulations again we possibly get a slightly different result. The following conclusions can be drawn based on simulations and theory.

- Both BGMF and EGMF give a better result (in all listed criteria) than SPGMF or PF\_2500.
- Computation time of BGMF and EGMF is approximately half of computation time of SPGMF.
- EGMF gives much better results than traditional EKF (EKF\_no res.).
• PF$_{2500}$ has a serious inconsistency problem, i.e. it under estimates the state covariance.

B. Summary of mixed cases

In Table III, we have listed a summary of one thousand 120 second simulations, which use both base station and satellite measurements with varying parameters. Parameters are following: variance of base station range measurement $\approx (30 \text{ m})^2$, variance of satellite pseudorange measurement $\approx (3 \text{ m})^2$ and variance of delta range measurement $\approx (0.1 \text{ m})^2$. We use restrictive information in the same way as simulations on Section VIII-A. Also notations and filters are the same as on simulations on Section VIII-A.

The following conclusions can be drawn based on simulations.

• Order of filter in Table III is almost the same as in Table II.
• Differences between different filters are smaller than in Table II, because there are also satellite measurements (very accurate linear measurements).
• Computation time of BGMF and EGMF is approximately half of computation time of SPGMF.
• EGMF gives much better results than traditional EKF (EKF$_{\text{no res.}}$).

<table>
<thead>
<tr>
<th>Solver</th>
<th>Time $\mu$s</th>
<th>Err.</th>
<th>Err. $95%$</th>
<th>Err.</th>
<th>Inc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>EKF$_{\text{no res.}}$</td>
<td>6</td>
<td>115</td>
<td>215</td>
<td>23</td>
<td>3.0</td>
</tr>
<tr>
<td>EKF</td>
<td>10</td>
<td>101</td>
<td>184</td>
<td>16</td>
<td>2.1</td>
</tr>
<tr>
<td>UKF</td>
<td>28</td>
<td>95</td>
<td>174</td>
<td>11</td>
<td>1.1</td>
</tr>
<tr>
<td>SPGMF</td>
<td>38</td>
<td>95</td>
<td>173</td>
<td>13</td>
<td>1.2</td>
</tr>
<tr>
<td>PF$_{2500}$</td>
<td>43</td>
<td>95</td>
<td>160</td>
<td>12</td>
<td>1.8</td>
</tr>
<tr>
<td>EKF2</td>
<td>11</td>
<td>94</td>
<td>176</td>
<td>12</td>
<td>1.2</td>
</tr>
<tr>
<td>BGMF</td>
<td>20</td>
<td>93</td>
<td>170</td>
<td>12</td>
<td>1.0</td>
</tr>
<tr>
<td>EGMF</td>
<td>20</td>
<td>92</td>
<td>169</td>
<td>12</td>
<td>1.0</td>
</tr>
<tr>
<td>Ref</td>
<td>$\infty$</td>
<td>83</td>
<td>156</td>
<td>0</td>
<td>1.2</td>
</tr>
</tbody>
</table>

**IX. CONCLUSION**

In this article, we have presented two new Gaussian Mixture Filters for the hybrid positioning: the Box GMF and the Efficient GMF. BGMF and EGMF are almost the same filter, because of this their performances are almost the same. Nevertheless EGMF gives slightly better results than BGMF. Both filters outperform the Sigma Point GMF, which outperforms the traditional single-component Kalman type filters such as EKF, UKF and EKF2. EGMF and BGMF also outperform particle filter, when number of particles is selected so that particle filter uses about the same time of computation as EGMF. GMFs and EKF works equivalently if we have only linear measurements, but the more nonlinearity occurs the better result GMFs give compared to EKF.

**ACKNOWLEDGMENT**

The author would like to thank Niilo Sirola and Robert Piché for their comments and suggestions. This study was partly funded by Nokia Corporation. The author acknowledges the financial support of the Nokia Foundation and the Tampere Graduate School in Information Science and Engineering.

**REFERENCES**