PARTICLE FLOW PARTICLE FILTER FOR GAUSSIAN MIXTURE NOISE MODELS

Soumyasundar Pal and Mark Coates

Dept. of Electrical and Computer Engineering, McGill University, Montreal, Quebec, Canada

ABSTRACT

Particle filters has become a standard tool for state estimation in nonlinear systems. However, their performance usually deteriorates if the dimension of state space is high or the measurements are highly informative. A major challenge is to construct a proposal density that is well matched to the posterior distribution. Particle flow methods are a promising option for addressing this task. In this paper, we develop a particle flow particle filter algorithm to address the case where both the process noise and the measurement noise are distributed as mixtures of Gaussians. Numerical experiments are performed to explore when the proposed method offers advantages compared to existing techniques.

Index Terms— non-linear sequential state estimation, particle flow, Daum-Huang filter, particle filter, Gaussian mixture model, high-dimensional filtering.

I. INTRODUCTION

Particle filters [1] perform sequential importance sampling to solve the discrete-time nonlinear filtering task in a Bayesian framework. However, their performance can be poor if the measurements are highly informative or the state dimension is high [2], [3]. Although there have been multiple proposals to address this issue [2]–[7], the methods rely on the posterior having a special structure or are computationally expensive.

Particle flow filters [8]–[12] exhibit improved performance. Instead of sampling, particles are migrated from the prior to the posterior by identifying and solving differential equations that link these two distributions. The approximations needed to implement these filters can lead to particles not being a genuine sample from the posterior. To address this, several recent algorithms combine particle flow and particle filtering [12]–[16]. In particular, [16] introduces the particle flow particle filter (PFPF), which uses particle flow to construct the proposal distribution in a particle filtering framework. The theoretical guarantees for any particle filter still apply, while the flow ensures that a good proposal has been constructed.

The PFPF of [16] relies on a Gaussian approximation to the prior and posterior for construction of the flow. For many dynamic and measurement models, this approximation is poor. Two efforts have been made to expand the applicability of the particle flow filters to mixture models [17], [18], but neither of these approaches leads to a particle filter. The potentially mismatched modelling assumptions and approximations needed to construct and compute the particle flow can negatively impact performance. In this paper we develop a particle flow particle filter that addresses settings where the dynamic and measurement models can be approximated by a nonlinearity with additive noise distributed according to a Gaussian mixture.

The paper is organized as follows. Section II states the filtering task that we address. Section III reviews the exact particle flow filter [10], [19] and Section IV reviews the particle flow particle filters of [16] . Section V introduces the proposed particle flow particle filter for Gaussian mixture models and Section VI presents and discusses the results of numerical simulation experiments. The conclusion is provided in Section VII.

II. PROBLEM STATEMENT

We consider the discrete-time filtering problem where our goal is to track the marginal posterior distribution $p(x_k|z_{1:k})$ recursively with time k, starting from a initial probability density function $p(x_0)$. The dynamic and measurement models are specified as:

$$x_k = g_k(x_{k-1}) + v_k \quad \text{for } k \geq 1\,, \tag{1}$$

$$z_k = h_k(x_k) + w_k \quad \text{for } k \geq 1\,. \tag{2}$$

Here $g_k : \mathbb{R}^d \to \mathbb{R}^d$ is the state-transition function of the unobserved state $x_k \in \mathbb{R}^d$, the measurement $z_k \in \mathbb{R}^S$ is generated conditioned on the current state x_k through a potentially nonlinear measurement model $h_k : \mathbb{R}^d \to \mathbb{R}^S$. The process and measurement noises are denoted by $v_k \in \mathbb{R}^d$ and $w_k \in \mathbb{R}^S$ respectively. We assume that $p(x_0) = \mathcal{N}(x_0|\bar{\mu}_0, \bar{P}_0)$ is Gaussian whereas $v_k \sim \Sigma_{m=1}^{M} \alpha_{k,m}\mathcal{N}(\psi_{k,m}, Q_{k,m})$ and $w_k \sim \Sigma_{n=1}^{N} \beta_{k,n}\mathcal{N}(\zeta_{k,n}, R_{k,n})$ are distributed according to Gaussian mixtures.

We observe that this model can be alternatively expressed as a switching state space model. We introduce two unobserved scalar valued discrete random variables $d_k \in \{1, 2, ...M\}$ and $c_k \in \{1, 2, ...N\}$ such that $P(d_k = m) = \alpha_{k,m}$ and $P(c_k = n) = \beta_{k,n}$. The d_k and c_k variables are independent for different k and independent of each other. Let $p(x_k|x_{k-1}, d_k = m) = \mathcal{N}(x_k|g_k(x_{k-1}) + \psi_{k,m}, Q_{k,m}), \forall 1 \leq m \leq M$ and $p(z_k|x_k, c_k = n) =$

$\mathcal{N}(z_k|h_k(x_k)+\zeta_{k,n}, R_{k,n}), \forall 1 \le n \le N$. The state transition density is then:

$$\begin{aligned} p(x_k|x_{k-1}) &= \textstyle\sum_{m=1}^{M} \alpha_{k,m} \mathcal{N}(x_k|g_k(x_{k-1})+\psi_{k,m}, Q_{k,m}) \\ &= \textstyle\sum_{m=1}^{M} P(d_k=m) p(x_k|x_{k-1}, d_k=m) \quad (3) \end{aligned}$$

Similarly, the likelihood is

$$\begin{aligned} p(z_k|x_k) &= \textstyle\sum_{n=1}^{N} \beta_{k,n} \mathcal{N}(z_k|h_k(x_k)+\zeta_{k,n}, R_{k,n}) \\ &= \textstyle\sum_{n=1}^{N} P(c_k=n) p(z_k|x_k, c_k=n) \qquad (4) \end{aligned}$$

III. EXACT PARTICLE FLOW (SINGLE GAUSSIAN)

Suppose the posterior at time $k-1$ is approximated by a set of N_p unweighted particles $\{x_{k-1}^i\}_{i=1}^{N_p}$. Propagating the particles through the dynamic model yields $\{\tilde{x}_k^i\}_{i=1}^{N_p}$, which represent the predictive posterior at time k. The objective of particle flow methods is then to gradually migrate the particles towards the correct regions of state-space so that they approximate the posterior at time k, when the flow is complete. Particle flow can be modelled as a background stochastic process η_λ in a pseudo time interval $\lambda \in [0,1]$. The time index k is temporarily omitted to simplify notation, because the particle flow only concerns migration of particles within a single time step. η_λ^i denotes the stochastic process's i-th realization, and we initialize the flow by setting $\eta_0^i = \tilde{x}_k^i$, for $i = 1, 2, \ldots, N_p$.

The *zero diffusion* particle flow filters [9], [10] involve deterministic flows of particles. The trajectory of η_λ^i for realization i follows the ordinary differential equation (ODE):

$$\frac{d\eta_\lambda^i}{d\lambda} = f(\eta_\lambda^i, \lambda)\,, \qquad (5)$$

where $f : \mathbb{R}^d \to \mathbb{R}^d$ is governed by the Fokker-Planck equation and additional flow constraints [10].

An analytically tractable solution of equation (5) exists when the predictive posterior and the likelihood distributions are both Gaussian and the measurement model is linear, i.e., $\eta_0^i \sim \mathcal{N}(\bar{\eta}_0, \bar{P}), z = H\eta_\lambda^i + w \sim \mathcal{N}(H\eta_\lambda^i, R)$.

III-A. The localized exact Daum and Huang filter

The localized exact Daum and Huang (LEDH) filter [19] uses linearized approximation of the measurement model to compute the drift term for each individual particle. For the i-th particle, the drift term is

$$f(\eta_\lambda^i, \lambda) = A^i(\lambda)\eta_\lambda^i + b^i(\lambda)\,, \qquad (6)$$

where

$$\begin{aligned} A^i(\lambda) =& -\frac{1}{2}\bar{P}H^i(\lambda)^T(\lambda H^i(\lambda)\bar{P}H^i(\lambda)^T + R)^{-1}H^i(\lambda), \\ b^i(\lambda) =& (I + 2\lambda A^i(\lambda))[(I+\lambda A^i(\lambda))\bar{P}H^i(\lambda)^T R^{-1}(z- \\ & e^i(\lambda)) + A^i(\lambda)\bar{\eta}_0]. \qquad (7) \end{aligned}$$

Here $H^i(\lambda) = \left.\frac{\partial h(\eta)}{\partial \eta}\right|_{\eta=\eta_\lambda^i}$ and $e^i(\lambda) = h(\eta_\lambda^i) - H^i(\lambda)\eta_\lambda^i$.

III-B. Numerical Implementation

The approximate solution of the ODE (5) is obtained numerically using Euler's method. We discretize the pseudo-time λ at N_λ positions $[\lambda_1, \lambda_2, \ldots, \lambda_{N_\lambda}]$, where $0 = \lambda_0 < \lambda_1 < \ldots < \lambda_{N_\lambda} = 1$. The step sizes $\epsilon_j = \lambda_j - \lambda_{j-1}$ for $j = 1, \ldots, N_\lambda$ satisfy $\sum_{j=1}^{N_\lambda} \epsilon_j = \lambda_{N_\lambda} - \lambda_0 = 1$. In the LEDH filter, we need to linearize $H^i(\lambda_j)$ to compute $A^i(\lambda_j)$ and this is performed at $\eta_{\lambda_{j-1}}^i$. The Euler update rule is

$$\eta_{\lambda_j}^i = \eta_{\lambda_{j-1}}^i + \epsilon_j(A^i(\lambda_j)\eta_{\lambda_{j-1}}^i + b^i(\lambda_j))\,.$$

IV. PARTICLE FLOW PARTICLE FILTER

Due to approximations in the flow implementation and mismatched modelling assumptions, the migrated particles after the particle flow process are not exactly distributed according to the posterior. A particle filter requires us to generate samples of the state from an importance distribution q and then calculate importance weights so that we can maintain a weighted particle approximation to the posterior as time evolves. η_1^i can be viewed as being drawn from a proposal distribution $q(\eta_1^i|x_{k-1}^i, z_k)$, which is possibly well matched to the posterior, because of the flow procedure. If the flow parameters, $(A^i(\lambda_p), b^i(\lambda_p))$ are computed based on linearization of the measurement function h at an auxiliary particle location $\bar{\eta}_{\lambda_p}^i$, starting from $\bar{\eta}_0^i = g_k(x_{k-1}^i)$, then under a mild smoothness condition on the measurement function h and with small enough step sizes ϵ_j, the discretized particle flow process introduces an invertible mapping $\eta_1^i = T(\eta_0^i; z_k, x_{k-1}^i)$, as shown in [16]. This property enables efficient evaluation of the importance density:

$$q(\eta_1^i|x_{k-1}^i, z_k) = \frac{p(\eta_0^i|x_{k-1}^i)}{|\det(\dot{T}(\eta_0^i; x_{k-1}^i, z_k))|}, \qquad (8)$$

where $\dot{T}(\cdot) \in \mathbb{R}^{d\times d}$ is the Jacobian function of the mapping $T(\cdot)$ and its determinant can be computed as

$$\det(\dot{T}(\eta_0^i; x_{k-1}^i, z_k)) = \prod_{p=1}^{N_\lambda} \det(I + \epsilon_p A^i(\lambda_p)) \qquad (9)$$

V. PARTICLE FLOW PARTICLE FILTER FOR GAUSSIAN MIXTURE MODELS

In this section we develop the novel particle flow particle filter. We consider the switching state representation of the Gaussian mixture model (1)-(2). We augment x_k with the unobserved discrete variables d_k and c_k and consider the target joint density to be $p(x_{0:k}, d_{1:k}, c_{1:k}|z_{1:k})$. We require that the importance distribution q factorizes:

$$\begin{aligned} q(x_{0:k}, d_{1:k}, c_{1:k}|z_{1:k}) &= q(x_{0:k-1}, d_{1:k-1}, c_{1:k-1}|z_{1:k-1}) \\ & q(x_k, d_k, c_k|x_{0:k-1}, d_{1:k-1}, c_{1:k-1}, z_{1:k}) \end{aligned} \qquad (10)$$

Samples $\{x^i_{0:k}, d^i_{1:k}, c^i_{1:k}\}^{N_p}_{i=1}$ are obtained by augmenting each existing sample,

$$(x^i_{0:k-1}, d^i_{1:k-1}, c^i_{1:k-1}) \sim q(x_{0:k-1}, d_{1:k-1}, c_{1:k-1}|z_{1:k-1})$$

with the new state

$$(x^i_k, d^i_k, c^i_k) \sim q(x_k, d_k, c_k|x^i_{0:k-1}, d^i_{1:k-1}, c^i_{1:k-1}, z_{1:k})\,.$$

The target joint density can be expressed as follows:

$$\begin{aligned}p(x_{0:k}, d_{1:k}, c_{1:k}|z_{1:k}) &\propto p(x_k, d_k, c_k|x_{k-1}, d_{k-1}, c_{k-1})\\ &\quad p(z_k|x_k, d_k, c_k)p(x_{0:k-1}, d_{1:k-1}, c_{1:k-1}|z_{1:k-1})\end{aligned}$$

We can calculate unnormalized importance weights as:

$$\begin{aligned}\omega^i_k &= \frac{p(x^i_{0:k}, d^i_{1:k}, c^i_{1:k}|z_{1:k})}{q(x^i_{0:k}, d^i_{1:k}, c^i_{1:k}|z_{1:k})},\\ &\propto \omega^i_{k-1}\frac{p(x^i_k, d^i_k, c^i_k|x^i_{k-1}, d^i_{k-1}, c^i_{k-1})p(z_k|x^i_k, d^i_k, c^i_k)}{q(x^i_k, d^i_k, c^i_k|x^i_{0:k-1}, d^i_{1:k-1}, c^i_{1:k-1}, z_{1:k})}\,,\end{aligned} \tag{11}$$

We design a proposal such that $q(x_k, d_k, c_k|x_{0:k-1}, d_{1:k-1}, c_{1:k-1}, z_{1:k}) = q(x_k, d_k, c_k|x_{k-1}, d_{k-1}, c_{k-1}, z_k)$ is satisfied, so that we only need to store the particles and weights from the previous time step, instead of the full trajectories. For the filtering problem, our focus is the marginal posterior, and we approximate this as

$$p(x_k|z_{1:k}) \approx \textstyle\sum_{i=1}^{N_p}\omega^i_k\delta(x_k - x^i_k) \tag{12}$$

where δ is the Dirac delta-function. From the dynamic model,

$$p(x^i_k, d^i_k, c^i_k|x^i_{k-1}, d^i_{k-1}, c^i_{k-1}) = P(d^i_k)P(c^i_k)p(x^i_k|x^i_{k-1}, d^i_k)\,.$$

We design q such that

$$\begin{aligned}q(x^i_k, d^i_k, c^i_k|x^i_{k-1}, d^i_{k-1}, c^i_{k-1}, z_k) &= Q(d^i_k)Q(c^i_k)\times\\ &\quad q(x^i_k|x^i_{k-1}, d^i_k, c^i_k, z_k)\end{aligned} \tag{13}$$

is satisfied. We choose $Q(d_k) = P(d_k)$, $Q(c_k) = P(c_k)$ and we calculate $q(x^i_k|x^i_{k-1}, d^i_k, c^i_k, z_k)$ based on the invertible mapping established by the flow [16].

$$q(x^i_k|x^i_{k-1}, d^i_k = m, c^i_k = n, z_k) = \frac{p(\eta^i_0|x^i_{k-1}, d^i_k = m)}{|\prod_{p=1}^{N_\lambda} det(I + \epsilon_p A^i_{mn}(\lambda_p))|}\,.$$

Then equation (11) can be rewritten as

$$\omega^i_k \propto \omega^i_{k-1}\frac{p(x^i_k|x^i_{k-1}, d^i_k)p(z_k|x^i_k, c^i_k)}{q(x^i_k|x^i_{k-1}, d^i_k, c^i_k, z_k)}\,. \tag{14}$$

The algorithm is summarized in Algorithm 1. The most computationally demanding parts of the algorithm are the matrix inverse operations needed to calculate $A^i_{mn}(\lambda_p)$ and $b^i_{mn}(\lambda_p)$. Since individual flow parameters are calculated for each of the N_p particles at each time step and there are a total of N_λ discrete pseudo time steps, the total computational complexity of the matrix inverse operations is $O(N_p N_\lambda S^3)$, where S is the measurement dimension.

Algorithm 1 PFPF-GMM

```
Initialization: Draw {x_0^i}_{i=1}^{N_p} from the prior p(x_0). Set x̂_0 to be the mean of p(x_0). Set {ω_0^i}_{i=1}^{N_p} = 1/N_p and {P_0^i}_{i=1}^{N_p} = P̄_0. Set λ_0 = 0.
for k = 1 to K do
    for i = 1, ..., N_p do
        Sample d_k^i = m ∈ {1, 2, ...M} with probability {α_{k,1}, α_{k,2}, ...α_{k,M}}
        Apply EKF/UKF prediction, {x_{k-1}^i, P_{k-1}^i} → {m_{k|k-1,m}^i, P_{k,m}^i}, using N(ψ_{k,m}, Q_{k,m})
        Calculate η̄_0^i = g_k(x_{k-1}^i) + ψ_{k,m}
        Propagate particle η_0^i = g_k(x_{k-1}^i) + v_{k,m}, where v_{k,m} ~ N(ψ_{k,m}, Q_{k,m})
        Sample c_k^i = n ∈ {1, 2, ...N} with probability {β_{k,1}, β_{k,2}, ...β_{k,N}}
        Set θ_{mn}^i = 1
        for p = 1, ..., N_λ do
            Set λ_p = λ_{p-1} + ε_p
            Calculate A_{mn}^i(λ_p) and b_{mn}^i(λ_p) from (7) with linearization performed at η̄_{λ_{p-1}}^i, and with z = z_k − ζ_{k,n}, R = R_{k,n}, η̄_0 = η̄_0^i and P̄ = P_{k,m}^i
            Migrate auxiliary particle: η̄_{λ_p}^i = η̄_{λ_{p-1}}^i + ε_p(A_{mn}^i(λ_p)η̄_{λ_{p-1}}^i + b_{mn}^i(λ_p))
            Migrate particle: η_{λ_p}^i = η_{λ_{p-1}}^i + ε_p(A_{mn}^i(λ_p)η_{λ_{p-1}}^i + b_{mn}^i(λ_p))
            θ_{mn}^i = θ_{mn}^i |det(I + ε_p A_{mn}^i(λ_p))|
        end for
        Set x_k^i = η_1^i
        Calculate importance weights:
            ω_k^i ∝ ω_{k-1}^i p(x_k^i|x_{k-1}^i, d_k^i = m)p(z_k|x_k^i, c_k^i = n) / (p(η_0^i|x_{k-1}^i, d_k^i = m)/θ_{mn}^i)
    end for
    for i = 1, ..., N_p do
        Normalize ω_k^i = ω_k^i / Σ_{s=1}^{N_p} ω_k^s
        Apply EKF/UKF update, {m_{k|k-1,m}^i, P_{k,m}^i} → {m_{k|k}^i, P_k^i} using N(ζ_{k,n}, R_{k,n})
    end for
    Estimate x̂_k = Σ_{i=1}^{N_p} ω_k^i x_k^i
    (Optional) resample particles : {x_k^i, P_k^i, ω_k^i}_{i=1}^{N_p} to obtain {x_k^i, P_k^i, 1/N_p}_{i=1}^{N_p}
end for
```

VI. NUMERICAL EXPERIMENTS AND RESULTS

We conduct numerical simulations for two scenarios. The first is a linear scenario, which allows us to compare the performance of the proposed filter with an (almost) optimal solution in the form of the Gaussian mixture model Kalman filter. The second nonlinear scenario requires a particle filter for accurate state estimates. We compare with an extended

Table I. Average and 5th and 95th sample percentiles of MSE and average execution time per step for the linear scenario of Section VI-A for 200 simulation trials.

Algorithm	No. of Particles	Avg. MSE	5th and 95th percentile MSE	Exec. time (s)
EKF-GMM	N/A	0.010	(0.009,0.010)	0.026
PF-GMM	50 per comp.	0.011	(0.010,0.011)	1.85
PFPF-GMM	200	0.012	(0.011,0.012)	1.77
GSPF	10^4 per comp.	78.93	(52.86,105.60)	2.02
UKF	N/A	1.99	(1.05,3.16)	0.019
LEDH	500	2.00	(1.05,3.20)	3.40
EDH	500	1.99	(1.06,3.16)	0.013
PFPF (LEDH)	500	0.20	(0.02,0.70)	4.53
PFPF (EDH)	10^5	0.033	(0.012,0.020)	1.97
BPF	10^6	8.14	(6.08,11.33)	4.21

Table II. Average and 5th and 95th sample percentiles of MSE and average execution time per step for the nonlinear scenario of Section VI-B for 200 simulation trials.

Algorithm	No. of Particles	Avg. MSE	5th and 95th percentile MSE	Exec. time (s)
EKF-GMM	N/A	4.64	(1.45,11.19)	0.031
PF-GMM	50 per comp.	0.77	(0.10,1.78)	2.75
PFPF-GMM	200	0.11	(0.06,0.32)	2.18
GSPF	10^4 per comp.	2.81	(1.39,4.67)	2.08
UKF	N/A	0.80	(0.51,1.23)	0.023
LEDH	500	1.41	(0.95,1.94)	3.63
EDH	500	4.40	(3.15,6.72)	0.018
PFPF (LEDH)	500	2.88	(1.36,5.61)	4.91
PFPF (EDH)	10^5	1.20	(0.10,2.36)	2.05
BPF	10^6	0.95	(0.67,1.42)	4.51

Kalman filter derived for Gaussian mixture models (EKF-GMM), the particle flow filter for Gaussian mixtures (PF-GMM) [18], the Gaussian Sum Particle Filter (GSPF) [20], an unscented Kalman filter (UKF), the exact Daum Huang (EDH) filter [10] and its localized version (LEDH) [19], the Particle Flow Particle Filters (PFPFs) based on EDH and LEDH [16], and a bootstrap particle filter using 1 million particles. All numerical simulations are executed using an Intel i7-4770K, 3.50GHz CPU and 32GB RAM.

VI-A. Linear Model

We first consider a linear dynamic and measurement model $x_k = \alpha x_{k-1} + v_k$ and $z_k = x_k + w_k$, where $x_k \in \mathbb{R}^d$ and $z_k \in \mathbb{R}^d$. We set $d = 64$ and $\alpha = 0.9$. The noise terms are drawn from GMMs: $v_k \sim \sum_{m=1}^{3} \frac{1}{3}\mathcal{N}(\mu_m \mathbf{1}_{d\times 1}, \sigma_v^2 I_{d\times d})$, where $\mu_1 = -1$, $\mu_2 = 0$, $\mu_3 = 1$ and $\sigma_v = 1$, $w_k \sim \sum_{n=1}^{3} \frac{1}{3}\mathcal{N}(\delta_n \mathbf{1}_{d\times 1}, \sigma_w^2 I_{d\times d})$, where $\delta_1 = -5$, $\delta_2 = 0$, $\delta_3 = 5$ and $\sigma_w = 0.1$. The true state starts with $x_0 = \mathbf{0}$. The experiment is executed 200 times for 50 time steps.

Table I summarizes the results, reporting the mean-squared error (MSE) in the state estimation. The proposed particle flow particle filter for Gaussian mixture models (PFPF-GMM) performs only slightly worse than the EKF-GMM algorithm, which is close to optimal for this case. The PF-GMM also achieves close to the (almost) optimal performance. The PFPF-GMM outperforms the other particle-based methods significantly because they either use inaccurate Gaussian approximations or are ill-suited to the high-dimensional problem (e.g. GSPF, BPF).

VI-B. Nonlinear Model

We now consider a nonlinear dynamical model $g_k : \mathbb{R}^d \to \mathbb{R}^d$ and measurement function $h_k : \mathbb{R}^d \to \mathbb{R}^d$. The c-th element of the measurement vector is $h_k^c(x_k) = \frac{(x_k^c)^2}{20}$. Each element c of the state vector is defined as follows:

$$g_k^c(x_{k-1}) = 0.5x_{k-1}^c + 8\cos(1.2(k-1)) + \begin{cases} 2.5\frac{x_{k-1}^{c+1}}{1+(x_{k-1}^{c})^2} & \text{, if } c = 1 \\ 2.5\frac{x_{k-1}^{c+1}}{1+(x_{k-1}^{c-1})^2} & \text{, if } 1 < c < d \\ 2.5\frac{x_{k-1}^{c}}{1+(x_{k-1}^{c-1})^2} & \text{, if } c = d \end{cases} \quad (15)$$

We set $d = 64$. $v_k \sim \sum_{m=1}^{3} \frac{1}{3}\mathcal{N}(\mu_m \mathbf{1}_{d\times 1}, \sigma_v^2 I_{d\times d})$, where $\mu_1 = -1$, $\mu_2 = 0$, $\mu_3 = 1$ and $\sigma_v = 0.5$, $w_k \sim \sum_{n=1}^{3} \frac{1}{3}$ $\mathcal{N}(\delta_n \mathbf{1}_{d\times 1}, \sigma_w^2 I_{d\times d})$, where $\delta_1 = -3$, $\delta_2 = 0$, $\delta_3 = 3$ and $\sigma_w = 0.1$, are drawn from a GMM . The true state starts at $x_0 = \mathbf{0}$. The experiment is executed 200 times for 50 time steps. Table II summarizes the results, reporting the mean-squared error (MSE) in the state estimation.

For this challenging nonlinear problem, the EKF-GMM algorithm and the UKF struggle. The BPF and the GSPF suffer from significant weight degeneracy. The PF-GMM algorithm outperforms the LEDH filter as the latter is based on the incorrect approximation of a single Gaussian likelihood. The proposed PFPF-GMM has a considerably smaller MSE than all other filters.

VII. CONCLUSION

In this paper, we developed a particle flow particle filter that can address the scenario when both process and measurement noise are mixtures of Gaussians. The proposed method can be employed in challenging high-dimensional filtering problems with multi-modal posteriors. Future research will investigate ways to reduce the computational overhead and assess performance when the Gaussian mixture models only approximate the true behaviour of the system.

VIII. REFERENCES

[1] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian

state estimation," in *IEE Proc. F Radar and Signal Processing*, Apr. 1993, vol. 140, pp. 107–113.

[2] A. Beskos, D. Crisan, and A. Jasra, "On the stability of sequential Monte Carlo methods in high dimensions," *Ann. Appl. Probab.*, vol. 24, no. 4, pp. 1396–1445, Aug. 2014.

[3] F. Septier and G. W. Peters, "Langevin and Hamiltonian based sequential MCMC for efficient Bayesian filtering in high-dimensional spaces," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 2, pp. 312–327, Mar. 2016.

[4] P. M. Djurić, T. Lu, and M. F. Bugallo, "Multiple particle filtering," in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Proc.*, Honolulu, HI, USA, Apr. 2007, vol. 3, pp. 1181–1184.

[5] P. M. Djurić and M. F. Bugallo, "Particle filtering for high-dimensional systems," in *Proc. IEEE Intl. Workshop Computational Advances in Multi-Sensor Adaptive Processing*, St. Martin, France, Dec. 2013, pp. 352–355.

[6] A. Beskos, D. Crisan, A. Jasra, K. Kamatani, and Y. Zhou, "A stable particle filter for a class of high-dimensional state-space models," *Advances in Applied Probability*, vol. 49, no. 1, pp. 24–48, Mar. 2017.

[7] P. Rebeschini and R. van Handel, "Can local particle filters beat the curse of dimensionality?," *Ann. Appl. Probab.*, vol. 25, no. 5, pp. 2809–2866, Oct. 2015.

[8] U. D. Hanebeck, K. Briechle, and A. Rauh, "Progressive Bayes: a new framework for nonlinear state estimation," in *Proc. SPIE Conf. Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, Orlando, FL, USA, Apr. 2003, vol. 5099, pp. 256–267.

[9] F. Daum and J. Huang, "Nonlinear filters with log-homotopy," in *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, San Diego, CA, USA, Sep. 2007, p. 669918.

[10] F. Daum, J. Huang, and A. Noushin, "Exact particle flow for nonlinear filters," in *Proc. SPIE Conf. Signal Proc., Sensor Fusion, Target Recog.*, Orlando, FL, USA, Apr. 2010, p. 769704.

[11] M. A. Khan and M. Ulmke, "Non-linear and non-Gaussian state estimation using log-homotopy based particle flow filters," in *Proc. Intl. Workshop Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Bonn, Germany, Oct. 2014, pp. 1–6.

[12] F. Eler De Melo, S. Maskell, M. Fasiolo, and F. Daum, "Stochastic particle flow for nonlinear high-dimensional filtering problems," *ArXiv eprints arXiv:1511.01448*, Nov. 2015.

[13] Sebastian Reich, "A guided sequential Monte Carlo method for the assimilation of data into stochastic dynamical systems," in *Recent Trends in Dynamical Systems*, vol. 35, pp. 205–220. Springer, Basel, Switzerland, Jul. 2013.

[14] P. Bunch and S. Godsill, "Approximations of the optimal importance density using Gaussian particle flow importance sampling," *J. Amer. Statist. Assoc.*, vol. 111, no. 514, pp. 748–762, Aug. 2016.

[15] J. Heng, A. Doucet, and Y. Pokern, "Gibbs flow for approximate transport with applications to Bayesian computation," *ArXiv eprints arXiv:1509.08787*, Sep. 2015.

[16] Y. Li and M. Coates, "Particle filtering with invertible particle flow," *IEEE Trans. Signal Processing*, vol. 65, no. 15, pp. 4102–4116, Aug. 2017.

[17] M. A. Khan, M. Ulmke, and W. Koch, "A log homotopy based particle flow solution for mixture of Gaussian prior densities," in *Proc. IEEE Int. Conf. Multisensor Fusion and Integration for Intelligent Systems*, Munich, Germany, Sep. 2016, pp. 546–551.

[18] S. Pal and M. Coates, "Gaussian sum particle flow filter," in *Proc. IEEE Intl. Workshop Computational Advances in Multi-Sensor Adaptive Processing*, Curacao, The Netherlands, Dec. 2017, Accepted.

[19] T. Ding and M. J. Coates, "Implementation of the Daum-Huang exact-flow particle filter," in *Proc. IEEE Statistical Signal Processing Workshop (SSP)*, Ann Arbor, MI, USA, Aug. 2012, pp. 257–260.

[20] J. H. Kotecha and P. M. Djurić, "Gaussian sum particle filtering," *IEEE Trans. Signal Processing*, vol. 51, no. 10, pp. 2602–2612, Oct 2003.