Gaussian Sum Particle Flow Filter

Soumyasundar Pal and Mark Coates
Department of Electrical and Computer Engineering
McGill University, 3480 University Street, Montreal, Quebec, Canada H3A2A7
Email: soumyasundar.pal@mail.mcgill.ca, mark.coates@mcgill.ca

***Abstract*—Particle flow filters provide an approach for state estimation in nonlinear systems. They can outperform many particle filter implementations when the state dimension is high or when the measurements are highly informative. Instead of employing importance sampling, the particles are migrated by numerically solving differential equations that describe a flow from the prior to the posterior at each time step. An analytical solution for the flow equation requires a Gaussian assumption for both the prior and the posterior. Recently Khan et al. [1] devised an approximate flow that could address the case when the prior is represented by a Gaussian Mixture Model (GMM) and the likelihood function is Gaussian. The solution involved inversion of a large matrix which made the computational requirements scale poorly with the state dimension. In this paper, we devise an approximate particle flow filter for the case when both the prior and the likelihood are modeled using Gaussian mixtures. We perform numerical experiments 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 offer a Monte Carlo based solution to the discrete-time nonlinear filtering task in a Bayesian framework. Weighted particles are propagated in time to approximate the filtering distribution sequentially. The bootstrap particle filter draws particles from the predictive prior distribution at each time step and updates the weights of the particles according to the likelihood of the measurements [2]. If the measurements are highly informative or the state dimension is high, after a few time steps, most of the particles lie in regions of state space where the likelihood is very low, leading to negligible weights for the majority of particles. This degeneracy in weights results in poor approximation of the posterior [3]–[7].

Multiple research avenues have been explored to develop filters that perform well in high dimensions. Some methods factorize or partition the state space [8]–[11]; others incorporate Markov Chain Monte Carlo (MCMC) methods within the particle filters [6], [7], [12]–[20]. Partitioning methods require that the posterior has a special structure, and the methods that incorporate MCMC are computationally demanding.

The "particle flow" filters, described in [21]–[30], can achieve impressive performance for a much reduced computational overhead. In these filters, particles are "migrated" to represent the posterior distribution by identifying and solving differential equations that link the prior and posterior distributions. Several recent algorithms combine particle flow and particle filtering [30]–[34].

Numerous variants of the particle flow filter have been developed, but most are intractable. When the measurement model is linear, and the prior and posterior distributions are Gaussian, the "exact" particle flow filter [24] can be expressed in terms of matrix computations. Recently, Khan et al. extended this to derive approximate flow equations for the case of a Gaussian mixture model prior and a linear Gaussian measurement model [1]. Although the solution is effective, the computation of the flow requires inversion of matrix whose size grows rapidly with respect to the state dimension.

In this paper, we develop a particle flow algorithm that combines exact particle flow with the extended Kalman filter implementation of the Gaussian sum filter to address the case where both the prior and likelihood can be modelled as Gaussian mixtures. The computational requirements of our proposed algorithm scale with respect to the state dimension in the same way as the exact particle flow filter.

The paper is organized as follows. In Section II we specify the problem that we address. Section III reviews the extended Gaussian sum filter [35], [36] and Section IV reviews the exact particle flow filter [24], [37]. Section V introduces the proposed Gaussian sum particle flow filter and Section VI presents and discusses the results of numerical simulation experiments. Section VII contains concluding remarks.

II. Problem Statement

The nonlinear filtering task we address involves tracking the marginal posterior distribution $p(x_k|z_{1:k})$, where x_k is the state of a system at time k and $z_{1:k} = \{z_1, \dots, z_k\}$ is a sequence of measurements collected up to time step k. The state evolution and measurements are described by the following model:

$$x_0 \sim p(x_0), \quad (1)$$

$$x_k = g_k(x_{k-1}) + v_k \quad \text{for } k \geq 1\,, \quad (2)$$

$$z_k = h_k(x_k) + w_k \quad \text{for } k \geq 1\,. \quad (3)$$

Here $p(x_0)$ is an initial probability density function, $g_k : \mathbb{R}^d \to \mathbb{R}^d$ is the state-transition function of the unobserved state $x_k \in \mathbb{R}^d$, $z_k \in \mathbb{R}^S$ is the measurement generated from the state x_k through a potentially nonlinear measurement model $h_k : \mathbb{R}^d \to \mathbb{R}^S$. $v_k \in \mathbb{R}^d$ is the process noise and $w_k \in \mathbb{R}^S$ is the measurement noise. We assume that $p(x_0) = \mathcal{N}(x_0|\bar{\mu}_0, \bar{P}_0)$ and $v_k \sim \mathcal{N}(0, Q_k)$ is Gaussian. We model $w_k \sim \Sigma_{j=1}^{N} \beta_{k,j} \mathcal{N}(\zeta_{k,j}, R_{k,j})$, as being distributed according to a Gaussian mixture.

III. Gaussian Sum Filter

Assume at time k that the predictive distribution is:

$$p(x_k|z_{1:k-1}) = \Sigma_{m=1}^{M}\alpha_{k,m}\mathcal{N}(x_k|\bar{\mu}_{k,m}, \bar{P}_{k,m})\,, \quad (4)$$

and

$$p(z_k|x_k) = \Sigma_{n=1}^{N}\beta_{k,n}\mathcal{N}(z_k|h_k(x_k) + \zeta_{k,n}, R_{k,n})\,. \quad (5)$$

Then the posterior approaches

$$p(x_k|z_{1:k}) \approx \Sigma_{m=1}^{M}\Sigma_{n=1}^{N}\gamma_{k,mn}\mathcal{N}(x_k|\mu_{k,mn}, P_{k,mn})\,, \quad (6)$$

uniformly in x_k as $\bar{P}_{k,m} \to 0$ [36]. From the interaction of the m-th component in the prior and the n-th component of the likelihood, we have the usual extended Kalman filter update:

$$\mu_{k,mn} = \bar{\mu}_{k,m} + K_{k,mn}(z_k - \zeta_{k,n} - h_k(\bar{\mu}_{k,m}))\,, \quad (7)$$

$$P_{k,mn} = (I - K_{k,mn}H_{k,m})\bar{P}_{k,m}\,, \quad (8)$$

$$H_{k,m} = \frac{\partial h_k(x)}{\partial x}|_{x=\bar{\mu}_{k,m}}\,, \quad (9)$$

$$K_{k,mn} = \bar{P}_{k,m}H_{k,m}^T(H_{k,m}\bar{P}_{k,m}H_{k,m}^T + R_{k,n})^{-1}\,. \quad (10)$$

The mixture proportion $\gamma_{k,mn}$ is given as follows:

$$\gamma_{k,mn} = \frac{\delta_{k,mn}}{\Sigma_{i=1}^{M}\Sigma_{j=1}^{N}\delta_{k,ij}}\,, \quad (11)$$

$$\delta_{k,ij} = \alpha_{k,i}\beta_{k,j}\mathcal{N}(z_k|h_k(\bar{\mu}_{k,i}) + \zeta_{k,j}, H_{k,i}\bar{P}_{k,i}H_{k,i}^T + R_{k,j}). \quad (12)$$

If the predictive prior and likelihood have M and N Gaussian components respectively, then the posterior will have MN components. We perform resampling of the components, as in [35], to reduce the number of components in the final representation to M. The posterior is then approximated as:

$$p(x_k|z_{1:k}) \approx \Sigma_{m=1}^{M}\hat{\gamma}_{k,m}\mathcal{N}(x_k|\hat{\mu}_{k,m}, \hat{P}_{k,m})\,. \quad (13)$$

With $p(x_k|z_{1:k})$ approximated as (13), the updated predictive distribution approaches

$$p(x_{k+1}|z_{1:k}) \approx \Sigma_{m=1}^{M}\alpha_{k+1,m}\mathcal{N}(x_{k+1}|\bar{\mu}_{k+1,m}, \bar{P}_{k+1,m})\,, \quad (14)$$

uniformly in x_k as $\hat{P}_{k,m} \to 0$, where,

$$\alpha_{k+1,m} = \hat{\gamma}_{k,m}\,, \quad (15)$$

$$\bar{\mu}_{k+1,m} = g_{k+1}(\hat{\mu}_{k,m})\,, \quad (16)$$

$$G_{k+1,m} = \frac{\partial g_{k+1}(x)}{\partial x}|_{x=\hat{\mu}_{k,m}}\,, \quad (17)$$

$$\bar{P}_{k+1,m} = G_{k+1,m}\hat{P}_{k,m}G_{k+1,m}^T + Q_{k+1}\,. \quad (18)$$

Each component of the predictive and posterior distribution follows extended Kalman filter (EKF) equations. We can recursively track the posterior by employing parallel EKFs. However, in higher dimensions, if g_k or h_k is highly non-linear, this approach breaks down. In this paper, we use particle flow to alleviate this problem. In the following section we review the exact particle flow equations for the case when the prior is represented using a single Gaussian and the measurement model is Gaussian and can be linearized.

IV. Exact Particle Flow (single Gaussian)

Suppose that we have a set of N_p particles $\{x_{k-1}^i\}_{i=1}^{N_p}$ approximating the posterior distribution at time $k-1$. After propagating particles using the dynamic model, we obtain particles $\{\tilde{x}_k^i\}_{i=1}^{N_p}$ that represent the predictive posterior distribution at time k. Particle flow then migrates the particles so that they approximate the posterior distribution at time k.

We can model the particle flow as a background stochastic process η_λ in a pseudo time interval $\lambda \in [0, 1]$. To simplify notation, we temporarily omit the time index k because the particle flow only concerns particle migration between two adjacent time steps. We denote by η_λ^i the stochastic process's i-th realization, and set $\eta_0^i = \tilde{x}_k^i$, for $i = 1, 2, \ldots, N_p$.

The *zero diffusion* particle flow filters [22]–[25], [38], [39] involve no random displacements of particles; the flows are deterministic. The trajectory of η_λ^i for realization i follows the ordinary differential equation (ODE):

$$\frac{d\eta_\lambda^i}{d\lambda} = f(\eta_\lambda^i, \lambda)\,, \quad (19)$$

where $f : \mathbb{R}^d \to \mathbb{R}^d$ is governed by the Fokker-Planck equation and additional flow constraints [25].

Equation (19) can lead to a variety of particle flow filters. The problem is analytically tractable when the predictive posterior and the likelihood distributions are both Gaussian and the measurement model is linear, i.e., $\eta_0^i \sim N(\bar{\eta}_0, \bar{P}), z = H\eta_\lambda^i + w \sim N(H\eta_\lambda^i, R)$. The predictive covariance $\bar{P}$ and the measurement covariance R are both positive definite, and H is the measurement matrix.

A. The localized exact Daum and Huang filter

The localized exact Daum and Huang filter (LEDH) [37] linearizes the system and updates the drift term for each individual particle. For the i-th particle, the drift term is

$$f(\eta_\lambda^i, \lambda) = A^i(\lambda)\eta_\lambda^i + b^i(\lambda)\,, \quad (20)$$

where

$$A^i(\lambda) = -\frac{1}{2}\bar{P}H^i(\lambda)^T(\lambda H^i(\lambda)\bar{P}H^i(\lambda)^T + R)^{-1}H^i(\lambda),$$
$$b^i(\lambda) = (I + 2\lambda A^i(\lambda))[(I + \lambda A^i(\lambda))\bar{P}H^i(\lambda)^T R^{-1}(z - e^i(\lambda)) + A^i(\lambda)\bar{\eta}_0]. \quad (21)$$

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

B. Numerical Implementation

The ODE is solved approximately using discretized pseudo-time integration. We identify a sequence of discrete steps at N_λ positions, $[\lambda_1, \lambda_2, \ldots, \lambda_{N_\lambda}]$, where $0 = \lambda_0 < \lambda_1 < \ldots < \lambda_{N_\lambda} = 1$. We allow the step size $\epsilon_j = \lambda_j - \lambda_{j-1}$ for $j = 1, \ldots, N_\lambda$ to vary but require that $\sum_{j=1}^{N_\lambda}\epsilon_j = \lambda_{N_\lambda} - \lambda_0 = 1$. In the LEDH, we need to linearize $H^i(\lambda_j)$ to update $A^i(\lambda_j)$ and this is performed at $\eta_{\lambda_{j-1}}^i$. The functional mapping becomes

$$\eta_{\lambda_j}^i = f_{\lambda_j}^i(\eta_{\lambda_{j-1}}^i) = \eta_{\lambda_{j-1}}^i + \epsilon_j(A^i(\lambda_j)\eta_{\lambda_{j-1}}^i + b^i(\lambda_j))\,. \quad (22)$$

V. Gaussian sum particle flow filter

The proposed filter combines exact particle flow with the Gaussian sum filter. Particles associated with each Gaussian in the mixture representing the predictive prior are migrated using particle flow to form one component of the mixture representing the posterior. Extended Kalman filters are run in parallel (one for each component of the mixture), but the means of these EKFs are updated using the component means calculated from the migrated particles. The particle flow filter and the parallel EKFs are thus intertwined, with the covariance matrices being computed by the EKF used for particle flow and the means being computed by particle flow used for EKF updates. The algorithm is summarized in Algorithm (1).

The filter is initialized at time $k = 0$, by sampling N_p particles from the initial density $\mathcal{N}(\bar{\mu}_0, \bar{P}_0)$. As there is only one Gaussian component in the initial density, we set $x^i_{0,1} = x^i_0$. At time $k = 1$, the number of Gaussian components in the predictive prior, $M = 1$. For successive time steps, we set $M = N$.

At each time step, the posterior is approximated by an M component Gaussian mixture, via resampling of Gaussian components (line 21 in Algorithm (1)). The particles associated with each of those M components are propagated through the Gaussian dynamic model to be distributed according to the corresponding components of the predictive prior for the next time step. The mixture proportions, means and covariances of components of predictive prior are calculated based on parallel EKF predictions (15), (16) and (18).

We loop over each component in the Gaussian mixture, applying particle flow to the particles that correspond to the component (lines 4-17 in Algorithm (1)). A separate flow is applied for each of the N components in the mixture representing the observation model, so at the end of this loop we have MN sets of particles, each representing a different component of the posterior. The mean of each component is estimated using the sample mean of the particles associated with that component. Parallel EKFs update the covariances and proportions of the mixture components (8), (11).

The most computationally demanding part of the algorithm is the inverse operations in calculating $A^i_{mn}(\lambda_p)$ and $b^i_{mn}(\lambda_p)$. Since individual flow parameters are calculated for each of N_p particles, and there are a total of MN separate flows at each time step with N_λ pseudo time steps, the total computational complexity of the matrix inverse operations is $O(MNN_pN_\lambda S^3)$, where S is the measurement dimension.

Algorithm 1 Particle flow for GMM predictive distribution and likelihood (PF-GMM).

1: Initialization: Draw $\{x^i_0\}^{N_p}_{i=1}$ from the initial probability density $p(x_0)$. Set $\bar{\mu}_0$ and $\bar{P}_0$ to be the mean and covariance of $p(x_0)$, respectively. Set $\lambda_0 = 0$.
2: **for** $k = 1$ to K **do**
3: Apply parallel EKF prediction : $\{\hat{\gamma}_{k-1,m}, \hat{\mu}_{k-1,m}, \hat{P}_{k-1,m}\}^M_{m=1} \rightarrow \{\alpha_{k,m}, \bar{\mu}_{k,m}, \bar{P}_{k,m}\}^M_{m=1}$
4: **for** $m = 1, \ldots, M$ **do**
5: **for** $i = 1, \ldots, N_p$ **do**
6: Propagate particles through dynamical model $\eta^i_{0,m} = g_k(x^i_{k-1,m}) + v_k$
7: **for** $n = 1, \ldots, N$ **do**
8: Set $\eta^i_{\lambda_0,mn} = \eta^i_{\lambda_0,m}$
9: **for** $p = 1, \ldots, N_\lambda$ **do**
10: Set $\lambda_p = \lambda_{p-1} + \epsilon_p$
11: Calculate $A^i_{mn}(\lambda_p)$ and $b^i_{mn}(\lambda_p)$ from (21) with linearization performed at $\eta^i_{\lambda_{p-1},mn}$, and with $z = z_k - \zeta_{k,n}$ $\bar{P} = \bar{P}_{k,m}$ and $R = R_{k,n}$
12: Migrate particles: $\eta^i_{\lambda_p,mn} = \eta^i_{\lambda_{p-1},mn} + \epsilon_p(A^i_{mn}(\lambda_p)\eta^i_{\lambda_{p-1},mn} + b^i_{mn}(\lambda_p))$
13: **end for**
14: Set $x^i_{k,mn} = \eta^i_{1,mn}$
15: **end for**
16: **end for**
17: **end for**
18: Apply parallel EKF update: $\{\alpha_{k,m}, \bar{\mu}_{k,m}, \bar{P}_{k,m}\}^M_{m=1} \rightarrow \{\gamma_{k,mn}, \mu_{k,mn}, P_{k,mn}\}^{M,N}_{m=1,n=1}$
19: Set $\mu_{k,mn} = 1/N_p \sum^{N_p}_{i=1} x^i_{k,mn}$
20: Estimate $\hat{x}_k = \sum^M_{m=1} \sum^N_{n=1} \gamma_{k,mn}\mu_{k,mn}$
21: Resample Gaussian components: $\{\gamma_{k,mn}, \mu_{k,mn}, P_{k,mn}\}^{M,N}_{m=1,n=1} \rightarrow \{\hat{\gamma}_{k,m}, \hat{\mu}_{k,m}, \hat{P}_{k,m}\}^M_{m=1}$
22: Keep only the particles corresponding to the components retained after resampling $\{x^i_{k,m}\}^{M,N_p}_{m=1,i=1}$
23: (Optional) Redraw particles $\{x^i_{k,m}\}^{N_p}_{i=1} \sim \mathcal{N}(\hat{\mu}_{k,m}, \hat{P}_{k,m})$
24: **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 is a nonlinear scenario that requires the use of a particle filter to obtain accurate state estimates. We also compare our method with the Gaussian Sum Particle Filter (GSPF), proposed in [35].

1) Linear Model: We adapt the spatial sensor network setup proposed in [7]. There are d sensors deployed uniformly on a two-dimensional grid $\{1, 2, \ldots, \sqrt{d}\} \times \{1, 2, \ldots, \sqrt{d}\}$, and d is set to 64 in this example. Each sensor collects measurements, independently of the other sensors, about the underlying state at its physical location. Denote the state at the c-th sensor's position at time k by $x^c_k \in \mathbb{R}$, and its measurement as $z^c_k \in \mathbb{R}$. Then the full state at all sensor positions at time k is denoted by $x_k = [x^1_k, x^2_k, \ldots, x^d_k]' \in \mathbb{R}^d$, and all measurements at time k form the measurement vector $z_k = [z^1_k, z^2_k, \ldots, z^d_k]' \in \mathbb{R}^d$.

The dynamic model and the measurement model are:

$$x_k = \alpha x_{k-1} + v_k \,, \tag{23}$$

$$z_k = x_k + w_k \,, \tag{24}$$

where $\alpha = 0.9$, $v_k \sim \mathcal{N}(\mathbf{0}_{d\times 1}, \Sigma_{d\times d})$. The (i, j)-th entry of Σ is given as, $\Sigma_{i,j} = \alpha_0 e^{-\frac{||R^i - R^j||^2_2}{\beta}} + \alpha_1 \delta_{i,j}$, where $R^i \in \mathbb{R}^2$ is

TABLE I: Average, 5th and 95th percentile of MSE and average execution time per step for the linear scenario of Section (VI-1), among 100 simulation trials.

Algorithm	No. of Particles	Avg. MSE	5th and 95th percentile of MSE	Exec. time (s)
EKF-GMM	N/A	0.19	(0.007, 1.353)	0.016
PF-GMM	50 per comp.	0.19	(0.008, 1.383)	2.42
GSPF	10^4 per comp.	14.31	(8.15, 22.59)	1.78
UKF	N/A	2.04	(1.06, 3.46)	0.007
LEDH	500	2.06	(1.06, 3.55)	3.25
BPF	10^6	13.50	(7.28, 22.30)	3.55

TABLE II: Average, 5th and 95th percentile of MSE and average execution time per step for the nonlinear scenario of Section (VI-2), among 100 simulation trials.

Algorithm	No. of Particles	Avg. MSE	5th and 95th percentile of MSE	Exec. time (s)
EKF-GMM	N/A	103.64	(13.44, 255.35)	0.017
PF-GMM	50 per comp.	1.62	(0.17, 6.93)	2.75
GSPF	10^4 per comp.	6.91	(3.46, 11.70)	1.64
UKF	N/A	21.56	(4.45, 57.72)	0.007
LEDH	500	2.18	(0.35, 6.31)	3.35
BPF	10^6	6.31	(2.85, 10.90)	3.88

the physical position of sensor i and $\delta_{i,j}$ is the Kronecker delta symbol ($\delta_{i,i} = 1$ and $\delta_{i,j} = 0$ for $i \neq j$). Following [7], we set $\alpha_0 = 3, \alpha_1 = 0.01, \beta = 20$. $w_k \sim 1/3\mathcal{N}(5.\mathbf{1}_{d\times 1}, \sigma_z^2 \mathbb{I}_{d\times d}) + 1/3\mathcal{N}(\mathbf{0}_{d\times 1}, \sigma_z^2 \mathbb{I}_{d\times d}) + 1/3\mathcal{N}(-5.\mathbf{1}_{d\times 1}, \sigma_z^2 \mathbb{I}_{d\times d})$, is drawn from a GMM. We set $\sigma_z = 0.1$ to simulate informative measurements. The true state starts with $x_0 = \mathbf{0}$. For the proposed PF-GMM algorithm, we set $M = N = 3$. The experiment is executed 100 times for 50 time steps. Table [I] summarizes the results, reporting the mean-squared error (MSE) in the state estimation. From Table [I], our method performs as well as the EKF-GMM algorithm, which is optimal in the linear case except for the error introduced by component resampling after each measurement update. It outperforms the LEDH filter and the UKF, as they run based on an incorrect Gaussian likelihood approximation. The BPF and the GSPF suffer from severe weight degeneracy, even if many particles are employed.

2) Nonlinear Model: We 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:

$$g_k^c(x_{k-1}) = 0.5x_{k-1}^c + 8\cos(1.2(k-1)) + \begin{cases} 2.5\frac{x_{k-1}^{c+1}}{1+(x_{k-1}^c)^2} & \text{, if } c = 1 \\ 2.5\frac{x_{k-1}^{c+1}}{1+(x_{k-1}^{c-1})^2} & \text{, if } 1 < c < d \\ 2.5\frac{x_{k-1}^{c}}{1+(x_{k-1}^{c-1})^2} & \text{, if } c = d \end{cases} \quad (25)$$

We set $d = 64$ and $M = N = 3$. The distributions of v_k and w_k are the same as those for the linear model. The true state starts at $x_0 = \mathbf{0}$. The experiment is executed 100 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 fail. The BPF and the GSPF suffer from severe weight degeneracy. The LEDH filter, based on the incorrect approximation of a single Gaussian likelihood, performs reasonably, but the proposed PF-GMM algorithm achieves significantly better accuracy. It is also faster than the LEDH filter and the BPF. All numerical simulations are carried out using an Intel i7-4770K, 3.50GHz CPU and 32GB RAM.

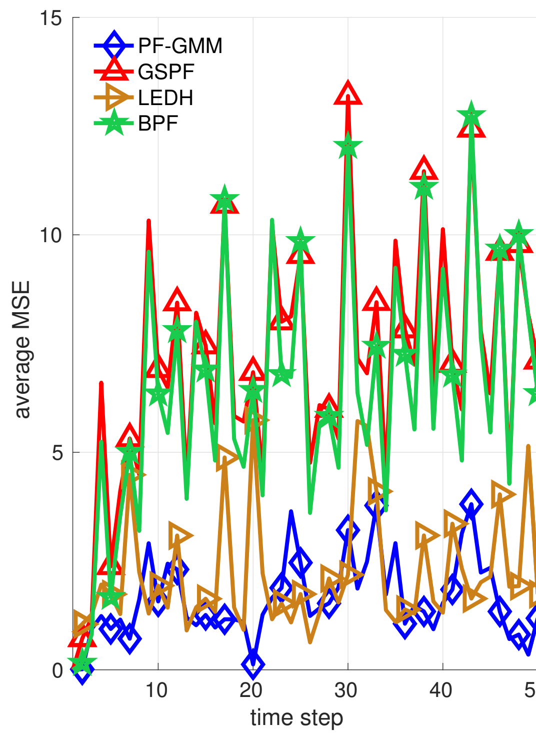

Fig. 1: MSE (averaged over 100 trials) vs time step for the nonlinear simulation scenario of Section VI-2.

VII. Conclusion

In this paper, we developed a Gaussian sum particle flow filter that can address the scenario when the prior and likelihood models are mixtures of Gaussians. The proposed method can be employed to address challenging high-dimensional filtering problems with multi-modal posteriors. Future research will investigate avenues for reducing the computational overhead and assess performance when the Gaussian mixture models are only approximations to the true behaviour of the system.

References

[1] 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.

[2] N. Gordon, D. Salmond, and A. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” in *IEE Proc. F Radar and Signal Processing*, vol. 140, no. 2, Apr. 1993, pp. 107–113.

[3] P. Bickel, B. Li, and T. Bengtsson, “Sharp failure rates for the bootstrap particle filter in high dimensions,” in *Pushing the limits of contemporary statistics: Contributions in honor of Jayanta K. Ghosh*. Institute of Mathematical Statistics, May 2008, pp. 318–329.

[4] T. Bengtsson, P. Bickel, and B. Li, “Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems,” in *Probability and Statistics: Essays in Honor of David A. Freedman*. Beachwood, OH, USA: Institute of Mathematical Statistics, Apr. 2008, vol. 2, pp. 316–334.

[5] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, “Obstacles to high-dimensional particle filtering,” *Mon. Weather Rev.*, vol. 136, no. 12, pp. 4629–4640, Dec. 2008.

[6] 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.

[7] 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.

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

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

[10] A. Beskos, D. Crisan, A. Jasra, K. Kamatani, and Y. Zhou, “A stable particle filter in high-dimensions,” *arXiv:1412.3501*, Dec. 2014.

[11] 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.

[12] C. Berzuini, N. G. Best, W. R. Gilks, and C. Larizza, “Dynamic conditional independence models and Markov chain Monte Carlo methods,” *J. Amer. Statist. Assoc.*, vol. 92, no. 440, pp. 1403–1412, Dec. 1997.

[13] W. R. Gilks and C. Berzuini, “Following a moving target–Monte Carlo inference for dynamic Bayesian models,” *J. R. Stat. Soc. B*, vol. 63, no. 1, pp. 127–146, Jan. 2002.

[14] S. Godsill and T. Clapp, “Improvement strategies for Monte Carlo particle filters,” in *Sequential Monte Carlo Methods in Practice*. New York: Springer, 2001, ch. 7, pp. 139–158.

[15] C. Musso, N. Oudjane, and F. Le Gland, “Improving regularised particle filters,” in *Sequential Monte Carlo Methods in Practice*. New York: Springer, 2001, ch. 10, pp. 247–271.

[16] P. D. Moral, A. Doucet, and A. Jasra, “Sequential Monte Carlo samplers,” *J. R. Stat. Soc. Ser. B (Stat. Method.)*, vol. 68, no. 3, pp. 411–436, Jan. 2006.

[17] A. Golightly and D. J. Wilkinson, “Bayesian sequential inference for nonlinear multivariate diffusions,” *Stat. Comput.*, vol. 16, no. 4, pp. 323–338, May 2006.

[18] V. Maroulas and P. Stinis, “Improved particle filters for multi-target tracking,” *J. Computational Physics*, vol. 231, no. 2, pp. 602–611, Jan. 2012.

[19] K. Kang, V. Maroulas, and I. D. Schizas, “Drift homotopy particle filter for non-Gaussian multi-target tracking,” in *Proc. Intl. Conf. on Information Fusion*, Salamanca, Spain, July 2014, pp. 1–7.

[20] A. Brockwell, P. D. Moral, and A. Doucet, “Sequentially interacting Markov chain Monte Carlo methods,” *Ann. Stat.*, vol. 38, no. 6, pp. 3387–3411, Nov. 2010.

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

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

[23] ——, “Particle flow for nonlinear filters with log-homotopy,” in *Proc. SPIE Signal and Data Processing Small Targets*, Orlando, FL, USA, Apr. 2008, p. 696918.

[24] 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.

[25] F. Daum and J. Huang, “Exact particle flow for nonlinear filters: Seventeen dubious solutions to a first order linear underdetermined PDE,” in *Proc. Asilomar Conf. Signals, Systems and Computers (ASILOMAR)*, Pacific Grove, CA, USA, Nov. 2010, pp. 64–71.

[26] ——, “Small curvature particle flow for nonlinear filters,” in *Proc. SPIE Signal and Data Processing of Small Targets*, Baltimore, MD, USA, May 2012, p. 83930A.

[27] ——, “Particle flow with non-zero diffusion for nonlinear filters,” in *Proc. SPIE Conf. Signal Proc., Sensor Fusion, Target Recog.*, Baltimore, MD, USA, May 2013, p. 87450P.

[28] ——, “Seven dubious methods to mitigate stiffness in particle flow with non-zero diffusion for nonlinear filters, Bayesian decisions, and transport,” in *Proc. SPIE Conf. Signal Proc., Sensor Fusion, Target Recog.*, Baltimore, MD, USA, May 2014, p. 90920C.

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

[30] F. E. de Melo, S. Maskell, M. Fasiolo, and F. Daum, “Stochastic particle flow for nonlinear high-dimensional filtering problems,” *arXiv:1511.01448*, Nov. 2015.

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

[32] 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.

[33] J. Heng, A. Doucet, and Y. Pokern, “Gibbs flow for approximate transport with applications to Bayesian computation,” *arXiv:1509.08787*, Sep. 2015.

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

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

[36] B. Anderson and J. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[37] 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.

[38] F. Daum, J. Huang, A. Noushin, and M. Krichman, “Gradient estimation for particle flow induced by log-homotopy for nonlinear filters,” in *Proc. SPIE Conf. Signal Proc., Sensor Fusion, Target Recog.*, Orlando, FL, USA, Apr. 2009, p. 733602.

[39] F. Daum, J. Huang, and A. Noushin, “Coulomb’s law particle flow for nonlinear filters,” in *Proc. SPIE Conf. Signal Proc., Sensor Fusion, Target Recog.*, San Diego, CA, USA, Sep. 2011, p. 81370B.