IMPLEMENTATION OF THE DAUM-HUANG EXACT-FLOW PARTICLE FILTER

Tao Ding and Mark J. Coates

McGill University
Department of Electrical and Computer Engineering
3480 University Street, Montreal, Quebec, Canada H3A 2A7

ABSTRACT

Several versions of the Daum-Huang (DH) filter have been introduced recently to address the task of discrete-time non-linear filtering. The filters propagate a particle set over time to track the system state, but, in contrast to conventional particle filters, there is no proposal density or importance sampling involved. Particles are smoothly migrated using a particle flow derived from a log-homotopy relating the prior and the posterior. Impressive performance has been demonstrated for a wide range of systems, but the implemented algorithms rely on an extended/unscented Kalman filter (EKF/UKF) that is executed in parallel. We illustrate through simulation that the performance of the exact flow DH filter can be compromised when the UKF and EKF fail. By introducing simple but important modifications to the exact flow DH filter implementation, the performance can be improved dramatically.

Index Terms**—** Daum-Huang filter, log-homotopy, particle filter, particle flow, exact flow

1. INTRODUCTION

Non-linear filtering for high-dimensional systems is an important problem that arises in a number of settings, including computational finance, meteorological system analysis, and multi-target tracking [1]. The performance of the particle filter can deteriorate significantly when the state dimension is large [2, 3]. Several mechanisms have been introduced to address this, including Markov Chain Monte Carlo (MCMC) steps in the particle sampling phase [4], accept/reject mechanisms [5], and the insertion of an annealing SMC sampler between filtering steps [3]. For some problems, these mechanisms can be very effective, but it is not clear that they can provide stable filtering solutions for general systems unless the number of particles (and hence computational cost) is allowed to grow exponentially with the state dimension [3].

Daum and Huang introduced an alternative non-linear filtering approach in [6–9]. In the Daum-Huang (DH) filters, a homotopy is introduced between the logarithms of the unnormalized prior and posterior densities at each time step. This homotopy defines a *particle flow*, described as the solution to a partial differential equation. The particle flow allows one to incrementally migrate a set of particles to regions where the posterior is large in value.

Impressive performance of the DH filters has been demonstrated for a variety of non-linear filtering systems [7, 8, 10]. When inspecting the implementation for the exact and incompressible flow DH filters, however, we observe that there is a reliance on an extended or unscented Kalman filter (EKF/UKF) that is executed in parallel [8, 10]. This limitation is probably eradicated in recent work introducing a Coulomb's law DH-filter [9]. We also note that it is not an intrinsic component of the exact flow filter of [8], but it is specified in [10], which provides more algorithmic detail.

We focus on the case where the system and observation noises are Gaussian, the system map is linear, but the observations are highly non-linear. We are interested in challenging filtering problems where the EKF and UKF fail. For such systems, the DH-filter implementation described in [10] performs relatively poorly. We introduce a modified implementation of the exact flow DH-filter of [8]. Our modifications are almost embarrassingly simple, but they have a dramatic impact on performance for some complex filtering problems. There are two significant modifications: (i) we introduce feedback between the Daum-Huang filter and the EKF/UKF; (ii) we linearize the system and calculate the migration parameters at each individual particle, rather than at a single representative state. The latter modification significantly increases the computational cost of the algorithm, but we find that it is essential for complex likelihood functions.

We provide simulation results for a multi-target tracking problem with state dimension 16 (four targets with position and velocity). The modified DH-filter implementation, using only 100 particles, significantly outperforms an MCMC particle filter (500 particles), a bootstrap particle filter (10^5 particles), and the unmodified DH-filter (500 particles).

In Section 2, we clarify the filtering problem that we address and provide a brief introduction to the Daum-Huang filter. In Section 3, we discuss the filter implementation and introduce our proposed modifications. Section 4 presents simulation results and in Section 5, we make concluding remarks.

2. PROBLEM STATEMENT AND BACKGROUND

We address a discrete-time non-linear filtering task for the case where the target dynamics and observations are described by the following Markovian state-space signal model:

$$x_k = f_k(x_{k-1}) + w_k \tag{1}$$
$$y_k = \gamma_k(x_k) + v_k. \tag{2}$$

In this model, x_k is a $d_x \times 1$ target state vector at time-step k, y_k is a $d_y \times 1$ measurement vector, w_k and v_k are system excitation and measurement noises, respectively, f_k is a nonlinear system map $f_k : \mathbb{R}^{d_x} \to \mathbb{R}^{d_x}$, and γ_k is a nonlinear measurement map $\gamma_k : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$. We focus on the case where w_k and v_k are drawn from Gaussian distributions and the system map f_k is linear, i.e. it can be described as $x_k = \Psi_k x_{k-1} + w_k$. Our goal is to recursively estimate the system state x_k at each time step k. We denote by $y_{1:k}$ the set of measurements from time-steps 1 to k.

2.1. The Daum-Huang Filter

Multiple versions of the Daum-Huang filter have been introduced [6–9]. Our summary in this section is based on the material in these papers and in [10]. The key idea in all versions is the introduction of a particle flow to smoothly migrate particles from the prior at each time step to the posterior.

Bayes' rule can be used to identify an unnormalized marginal posterior density:

$$\tilde{p}(x_k|y_{1:k}) = p(y_k|x_k)p(x_k|y_{1:k-1}). \tag{3}$$

Denoting $h(x_k) = p(y_k|x_k)$ and $g(x_k) = p(x_k|y_{1:k-1})$, Daum and Huang introduce a homotopy function $\phi(x_k, \lambda)$:

$$\phi(x_k, \lambda) = \log g(x_k) + \lambda \log h(x_k), \tag{4}$$

where λ is a real valued parameter that varies from 0 to 1. The homotopy defines a continuous deformation from $\log g(x_k)$ (when $\lambda = 0$) to the log of the unnormalized posterior density $\log \tilde{p}(x_k|y_{1:k})$ (when $\lambda = 1$).

The homotopy is used to induce a particle flow. In the original incompressible flow filter [6, 7], the flow was developed by requiring that the homotopy function remain constant as λ evolved, i.e. $\frac{\mathrm{d}\phi}{\mathrm{d}\lambda} = 0$. This requirement leads to the partial differential equation:

$$\frac{\partial \phi}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}\lambda} + \frac{\partial \phi}{\partial \lambda} = 0. \tag{5}$$

The flow of particles is computed by solving (5). Daum and Huang identified the unique minimum norm solution using the generalized inverse (see [7] for details). The induced flow of particles is in the direction of the gradient of the log-homotopy, with speed proportional to $\log(h)$, and the flow stops when the gradient is zero.

Daum and Huang generalized and improved the filter in [8], introducing the *exact flow* DH filter. Suppose that the particle flow obeys the ordinary differential equation $\frac{\mathrm{d}x}{\mathrm{d}\lambda} = \psi(x, \lambda)$. Using the Fokker-Planck equation, Daum and Huang derived the following relationship [8]:

$$\frac{\partial \phi}{\partial x}\psi(x, \lambda) + \log(h) = -\mathrm{Tr}(\frac{\partial \psi}{\partial x}) \tag{6}$$

The solution to this equation specifies the exact flow of the probability density.

For problems in which $\log(h)$ and $\log(g)$ are polynomials in the components of x (Gaussian and other exponential distributions), a closed-form solution of $\frac{\mathrm{d}x}{\mathrm{d}\lambda}$ can be derived from (6). For the linear Gaussian model, let $\overline{x}$ denote the predicted value of x and denote by P the covariance matrix of the prediction error. Let H be the measurement matrix ($y_k = Hx_k + v_k$), and let R be the covariance matrix of the measurement noise. Daum and Huang derive the following solution for the linear Gaussian scenario [8]:

$$\frac{\mathrm{d}x}{\mathrm{d}\lambda} = A(\lambda)x + b(\lambda) \tag{7}$$

where

$$A = -\frac{1}{2}PH^T(\lambda HPH^T + R)^{-1}H \tag{8}$$
$$b = (I + 2\lambda A)[(I + \lambda A)PH^T R^{-1} z + A\overline{x}]. \tag{9}$$

For non-linear models, Taylor series expansions can be employed. An estimate of $\overline{x}$ can be formed from the particles and a linearization of the measurement model can be conducted to construct a measurement matrix $H_{\overline{x}}$.

3. DH EXACT FLOW FILTER IMPLEMENTATION

In this section we detail the primary contribution of the paper. Algorithm 2 provides pseudocode for the exact-flow particle filter, based on the presentation in [10]. In this algorithm, N is the number of particles and T is the number of timesteps. The EKF/UKF state and covariance matrix estimates are denoted by m and P. Particle migration is performed by calculating $\frac{\mathrm{d}x}{\mathrm{d}\lambda}$ at N_λ discrete values of λ (lines 7-16).

A UKF or EKF is run in parallel to the exact flow filter (lines 6 and 17). The UKF/EKF provides covariance matrix estimates that are used both when evaluating A and b for the particle migration (line 10) and when forming the estimates $\overline{x}_k$ and $\hat{x}_k$ (lines 15 and 18). The measurement matrix $H_{\overline{x}}$ is calculated by linearizing at the current estimate $\overline{x}_k$ (line 9). A and b are evaluated using $H_{\overline{x}}$; the same values of A and b are applied to calculate the update for every particle.

As discussed in the introduction, our modifications are simple. We replace the pseudocode in lines 7-19 of Algorithm 1 with the pseudocode in Algorithm 2. There are two major changes. For each particle, we perform the linearization of the measurement function at the particle location to

Algorithm 1: Original Exact Flow Daum-Huang Filter

1 Initialization : Draw $\{x_0^i\}_{i=1}^N$ from the prior $p(x_0)$;
2 Set $\hat{x}_0$ and m_0 as the mean; P_0 as the covariance matrix.
3 **for** $k = 1$ *to* T **do**
4   Propagate particles $x_{k-1}^i = f_k(x_{k-1}^i) + v_k$;
5   Calculate the mean value $\overline{x}_k$;
6   Apply UKF/EKF prediction: $(m_{k-1|k-1}, P_{k-1|k-1}) \rightarrow (m_{k|k-1}, P_{k|k-1})$;
7   **for** $j = 1, \ldots, N_\lambda$ **do**
8   Set $\lambda = j\Delta\lambda$;
9   Calculate $H_{\overline{x}}$ by linearizing $\gamma_k()$ at $\overline{x}_k$;
10   Calculate A and b from (8) and (9) using $P_{k|k-1}$, $\overline{x}$ and $H_{\overline{x}}$;
11   **for** $i = 1, \ldots, N$ **do**
12   Evaluate $\frac{\mathrm{d}x_k^i}{\mathrm{d}\lambda}$ for each particle from (7) ;
13   Migrate particles: $x_k^i = x_k^i + \Delta\lambda \cdot \frac{\mathrm{d}x_k^i}{\mathrm{d}\lambda}$;
14   **endfor**
15   Re-evaluate $\overline{x}_k$ using the updated particles x_k^i.
16   **endfor**
17   Apply UKF/EKF update: $(m_{k|k-1}, P_{k|k-1}) \rightarrow (m_{k|k}, P_{k|k})$;
18   Estimate $\hat{x}_k$ from the particles x_k^i using $P_{k|k}$;
19   Optional: redraw particles $x_k^i \sim N(\hat{x}_k, P_{k|k})$;
20 **endfor**

Algorithm 2: Modified Local Exact Flow Daum-Huang Filter (replaces lines 7-19 of Algorithm 1)

7 **for** $j = 1, \ldots, N_\lambda$ **do**
8   Set $\lambda = j\Delta\lambda$;
9   **for** $i = 1, \ldots, N$ **do**
10   Calculate H_{x_i} by linearizing $\gamma_k()$ at x^i ;
11   Calculate A^i and b^i from (8) and (9) using $P_{k|k-1}$, $\overline{x}$ and H_{x_i} ;
12   Evaluate $\frac{\mathrm{d}x_k^i}{\mathrm{d}\lambda}$ for each particle from (7) ;
13   Migrate particles: $x_k^i = x_k^i + \Delta\lambda \cdot \frac{\mathrm{d}x_k^i}{\mathrm{d}\lambda}$;
14   **endfor**
15   Re-evaluate $\overline{x}_k$ using updated particles x_k^i.
16 **endfor**
17 Apply UKF/EKF update: $(m_{k|k-1}, P_{k|k-1}) \rightarrow (m_{k|k}, P_{k|k})$;
18 Estimate $\hat{x}_k$ from the particles x_k^i using $P_{k|k}$;
19 Set $m_{k|k} = \hat{x}_k$;
20 Optional: redraw particles $x_k^i \sim N(\hat{x}_k, P_{k|k})$;

obtain H_i (line 10). We calculate individual values of A_i and b_i for each particle using the matrix H_i (line 11), and use these in (7) to calculate the migration step for x_k^i. In line 19, we replace the mean estimate from the UKF/EKF with the state estimate from the Daum-Huang filter.

4. SIMULATION EXPERIMENTS

We illustrate the importance of our modifications with a multi-target tracking problem[1], adapted from [11]. We model a wireless sensor network consisting of 25 acoustic amplitude sensor nodes deployed on a jittered grid within a rectangular region of size $40\text{m} \times 40\text{m}$. Four targets move independently in the x-y plane. The p-th target is represented by the state vector $\mathrm{x}_k^{(p)} = (x_k^{(p)}, y_k^{(p)}, \dot{x}_k^{(p)}, \dot{y}_k^{(p)})$, containing the position and velocity. The overall state vector is the concatenation of these four individual vectors (and thus has dimension 16).

Each target state evolves according to $x_k^{(p)} = G_p x_{k-1}^{(p)} + W_p u_k^{(p)}$ where $u_k^{(p)} \sim N(0, C_p)$ is an i.i.d. Gaussian driving noise. $\mathrm{G}_p \in \mathbb{R}^{4\times4}$ and $\mathrm{W}_p \in \mathbb{R}^{4\times2}$ are system matrices that will be specified below. Each target p emits a sound with a (root mean squared) amplitude A_p that is assumed constant and known. At the position of sensor j, denoted ξ_k^j, the sound amplitude due to target p is modeled as $A_p / \left(\left\| \rho_k^{(p)} - \xi_k^j \right\|^\kappa + d_0 \right)$, where $\rho_k^{(p)} = (x_k^{(p)}, y_k^{(p)})^T$ is the position of target p and κ is the path loss exponent, and d_0 is a threshold that determines the maximum measurable amplitude. The measurement y_k^j obtained by sensor j at time k is then:

$$y_k^j = \gamma_k^j(x_k) + v_k^j \tag{10}$$

with

$$\gamma_k^j(\mathrm{x}_k) = \sum_{p=1}^{P} \frac{A_p}{\left\| \rho_k^{(p)} - \xi_k^j \right\|^\kappa + d_0} \tag{11}$$

where $v_k^j \sim N(0, \sigma_v^2)$ are zero mean Gaussian variables of common variance σ_v^2.

In our simulations, the number of targets is $P = 4$. The system matrix G_p and W_p are identical for the four targets and correspond to a constant-velocity motion model with some additional independent position error.

$$\mathrm{G}_p = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \ \mathrm{W}_p = \begin{pmatrix} 0.5 & 0 & 0.2 & 0 \\ 0 & 0.5 & 0 & 0.2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

The diagonal elements of the driving noise covariance matrix C_p are set to 0.00035. Each of the target emits a sound of amplitude $A_p = 10$; the measurement noise variance is $\sigma_v^2 = 0.1$ and the path loss exponent is $\kappa = 1$. We set the threshold $d_0 = 0.1$. The targets are initialized

[1] We acknowledge that the problem we present here is well-suited to Rao-Blackwellization, which can effectively reduce the dimension from 16 to 8. Our purpose is to compare particle filters and DH filters operating on (relatively) high dimensional systems, so we do not employ the technique here. It is relatively straightforward to construct a slightly modified version of the system we analyze for which standard Rao-Blackwellization is not possible.

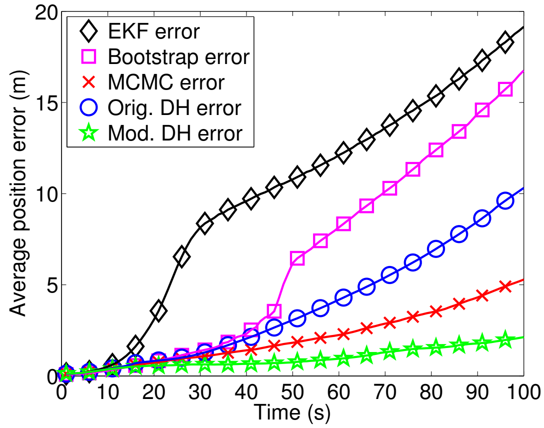

Fig. 1. Mean position error (averaged over the four targets and 50 trials) for the extended Kalman filter (EKF); the bootstrap particle filter (10^5 particles); the MCMC particle filter [4] (500 particles); the original Daum-Huang filter (500 particles); and the modified Daum-Huang filter (100 particles).

with states $[12, 6, 0.001, 0.002]$, $[32, 32, -0.001, -0.005]$, $[20, 13, -0.1, 0.01]$ and $[15, 35, 0.002, 0.002]$. The EKF is initialized by sampling m_0 from a Gaussian centred at the true initial states, with variance 0.1 for the position elements and 0.0005 for the velocity elements. Particles are then drawn from the same Gaussian, but centred at m_0.

Fig. 1 compares the mean position errors for the modified and original DH filters, the MCMC particle filter from [4], and the bootstrap particle filter. The computation times per time step (Matlab code executed on a 1.8GHz Intel Core i7 dual core, 4GB machine) are approximately 0.015s for the original DH filter, 0.3s for the modified DH filter, 0.8s for the bootstrap filter, and 5.4s for the MCMC filter. These values should be interpreted with care, because they are implementation-dependent. The MCMC particle filter code is slower in Matlab because it is challenging to vectorize the Markov chain sampling. Fig. 2 shows box plots of the errors for the MCMC particle filter and for the modified DH particle filter. The median error of the DH filter is stable over the course of the simulation, with values under 1m (growth in the mean is due to a few lost tracks). By contrast, the median error of the MCMC particle filter grows over time.

5. CONCLUDING REMARKS

We introduced modifications to the implementation of the Daum-Huang exact-flow particle filter. We provide feedback from the DH filter to the parallel EKF/UKF filter and calculate linearizations of the measurement function at all particle locations instead of just at the current state estimate. This substantially increases the computational overhead but makes the filter much more robust. We demonstrated the value of these modifications for a multi-target tracking problem.

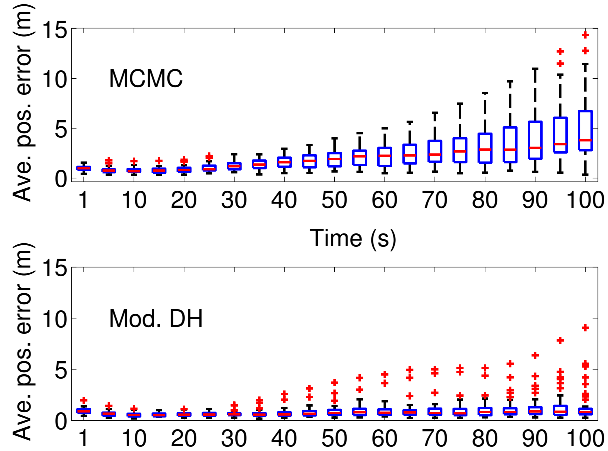

Fig. 2. Box plots of the position error (averaged over the four targets) for the MCMC particle filter (top) and the modified Daum-Huang filter (bottom).

6. REFERENCES

[1] D. Crisan and B. Rozovskii, Eds., *The Oxford Handbook of Nonlinear Filtering*, Oxford University Press, 2011.

[2] P.B. Quang, C. Musso, and F. Le Gland, "An insight into the issue of dimensionality in particle filtering," in *Proc. ISIF Int. Conf. Information Fusion*, Edinburgh, Scotland, July 2010.

[3] A. Beskos, D. Crisan, A. Jasra, and N. Whiteley, "Error bounds and normalizing constants for sequential Monte Carlo in high dimensions," arXiv:1112.1544v1.

[4] F. Septier, S.K. Pang, A. Carmi, and S. Godsill, "On MCMC-based particle methods for Bayesian filtering: Application to multitarget tracking," in *Proc. IEEE Int. Work. Comp. Adv. Multi-Sensor Adapt. Proc.*, Aruba, Dutch Antilles, Dec. 2009.

[5] P. Del Moral, *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*, Springer:New York, 2004.

[6] 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, Sept. 2007.

[7] F. Daum and J. Huang, "Nonlinear filters with particle flow induced by log-homotopy," in *Proc. SPIE Conf. Signal Proc., Sensor Fusion, Target Recog.*, Orlando, FL, USA, Apr. 2009.

[8] 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.

[9] F. Daum, J. Huang, and A. Noushin, "Coulomb's law particle flow for nonlinear filters," in *Proc. SPIE Conf. Signal and Data Proc.*, San Diego, CA, USA, Aug. 2011.

[10] S. Choi, P. Willett, F. Daum, and J. Huang, "Discussion and application of the homotopy filter," in *Proc. SPIE Conf. Sig. Proc., Sens. Fus., Targ. Recog.*, Orlando, FL, USA, Apr. 2011.

[11] O. Hlinka, O. Sluciak, F. Hlawatsch, P. M. Djuric, and M. Rupp, "Distributed Gaussian particle filtering using likelihood consensus," in *Proc. Int. Conf. Acoustics, Speech and Signal Proc.*, Prague, Czech Republic, May 2011.