



NRL/MR/5344--19-9938

Consideration of Particle Flow Filter Implementations and Biases

DAVID F. CROUSE

*Surveillance Technology Branch
Radar Division*

CODIE LEWIS

*STEM Student Employment Program
Radar Division*

February 11, 2020

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 11-02-2020		2. REPORT TYPE NRL Memorandum Report		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE Consideration of Particle Flow Filter Implementations and Biases				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) David F. Crouse and Codie Lewis*				5d. PROJECT NUMBER 53-1J47-09	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER 1J47	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory 4555 Overlook Avenue, SW Washington, DC 20375-5320				8. PERFORMING ORGANIZATION REPORT NUMBER NRL/MR/5344--19-9938	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research One Liberty Center 875 North Randolph Street, Suite 1425 Arlington, VA 22203-1995				10. SPONSOR / MONITOR'S ACRONYM(S) ONR	
				11. SPONSOR / MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES *STEM Student Employment Program, 4555 Overlook Ave., S.W., Washington, DC 20375-5320.					
14. ABSTRACT Particle flow filters are appealing due to their potential resistance to particle collapse. However, common implementations exhibit undesirable biases or particle divergence. This paper shows that the explicit, incompressible, and diagonal flows, unlike the Gromov flow, are inherently biased. Another issue is errors in the numerical integration of the flow. The benefits of implicit stochastic-integration methods are demonstrated and a new adaptive step-size selection heuristic is presented.					
15. SUBJECT TERMS Particle filtering Particle flow Nonlinear estimation Gaussian approximation Biases					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Unclassified Unlimited	18. NUMBER OF PAGES 19	19a. NAME OF RESPONSIBLE PERSON David F. Crouse
a. REPORT Unclassified Unlimited	b. ABSTRACT Unclassified Unlimited	c. THIS PAGE Unclassified Unlimited			19b. TELEPHONE NUMBER (include area code) (202) 404-8106

This page intentionally left blank.

Consideration of Particle Flow Filter Implementations and Biases

David Frederic Crouse* and Codie Tyler Lewis*

Abstract

Particle flow filters are appealing due to their potential resistance to particle collapse. However, common implementations exhibit undesirable biases or particle divergence. This paper shows that the explicit, incompressible and diagonal flows, unlike the Gromov flow, are inherently biased. Another issue is errors in the numerical integration of the flow. The benefits of implicit stochastic-integration methods are demonstrated and a new adaptive step-size selection heuristic is presented. This report is an extended version of the conference paper [10].

I. INTRODUCTION

IN many estimation problems, the posterior probability density function (PDF) after updating with one or more measurements, has a form that is analytically intractable, necessitating the use of approximations. Whereas Gaussian approximations lead to elegant solutions, such as the Kalman filter for dynamic estimation [3, Ch. 3.5], [7], there are many instances where a Gaussian approximation does not suffice. One possible non-Gaussian estimation technique involves particle filters. This paper looks at improving homotopy integration methods for particle flow filters (PFFs) and explaining why certain flows for the filter do not work well. The focus is only on the measurement-update step. Propagation over time is simple: Given a continuous-time stochastic dynamic model, one can just integrate the stochastic differential equation for every particle using one of many formulae in [57].

Traditional particle filters [2], [43], [64] are sequential Monte-Carlo methods. They approximate a PDF as a set of points and weights. The downside of such methods is that measurements that are extremely accurate and/or that occur in unlikely regions (based on the prior distribution) can lead to “particle collapse.” The first practical particle filter, the bootstrap filter, also known as the sequential importance sampling filter, introduced in [46], uses techniques to reduce the likelihood of particle collapse. However, it is not actually immune to collapse. Indeed, its vulnerability to particle collapse has led to the development of many improved variants that utilize resampling, regularization, and other techniques to reduce the likelihood of particle collapse. A number of methods are given in the tutorial [2].

In contrast, the PFF, also known as the homotopy particle filter or the Daum-Huang particle filter, introduced in [12], is immune to particle collapse by definition of how it works. The PFF does not change the weights of the particles at all. Rather, it moves the particles. Thus, failure of the filter is represented by particles moving to regions of the estimation space that are not representative of the true distribution. Section II addresses the derivation of such filters.

The difference in robustness between traditional particle filters and PFFs is exemplified by Fig. 1. In that example, the prior distribution is Gaussian with mean $(0, 3 \text{ km})$ and covariance matrix $\mathbf{P} = \sigma^2 \mathbf{I}$, where $\mathbf{I}$ is the identity matrix and $\sigma = 1 \text{ km}$. One hundred particles with uniform weight are drawn from the prior distribution. The receiver is placed at $(0, 3 \text{ km})$, so the prior distribution essentially says that the target is somewhere near the receiver. A single bistatic measurement of range and polar angle is taken with the transmitter located at $(-3 \text{ km}; 0)$. Thus, the measurement is a nonlinear transformation of the state. In the measurement coordinate system, the measurement is corrupted with additive Gaussian noise with covariance matrix $\mathbf{R} = \text{diag}(\sigma_r^2, \sigma_\theta^2)$, where $\sigma_r = 200 \text{ m}$ and $\sigma_\theta = 10^{-3} \text{ rad}$.

In Fig. 1, the posterior PDF is almost identical to the measurement PDF, because the prior distribution is so bad. The red particles are the results of the homotopy particle filter implemented with the Gromov flow, exponential base-2 step-size selection, and a semi-implicit Euler-Maruyama method (these algorithms are subsequently discussed). In comparison, a traditional particle filter does not move the particles (though there could be a resampling step). Rather it changes the weights. Under double-precision arithmetic, in such an instance, all weight would numerically collapse onto the particle marked with a green “X.” Thus, the potential advantage of the PFF is clear.

However, many implementations of the PFF do not work well. Some authors report particles diverging, necessitating the use of heuristic techniques to redraw divergent particles based on other particles [6], [42], [47], [53], [55], [56]. This paper looks at some things that can be done to improve the performance of PFFs. The focus is on variants that have simple expressions for the flow and where in certain equations a Gaussian prior approximation is used to make the flows easy to implement. This is not the same Gaussian approximation that is used in Kalman-filtering algorithms. When a Gaussian prior approximation is necessary, it is assumed that the prior distribution is Gaussian with mean $\mathbf{x}_{\text{prior}}$ and covariance matrix $\mathbf{P}$. The mean and

*The authors are employed by the Naval Research Laboratory, Attn: Code 5344, 4555 Overlook Ave., SW, Washington DC 20375. (e-mail: david.crouse@nrl.navy.mil, codie.lewis@nrl.navy.mil) This research is supported by the Office of Naval Research (ONR) through the Naval Research Laboratory (NRL) 6.1 Base Program. This report is an expanded version of the conference paper [10]

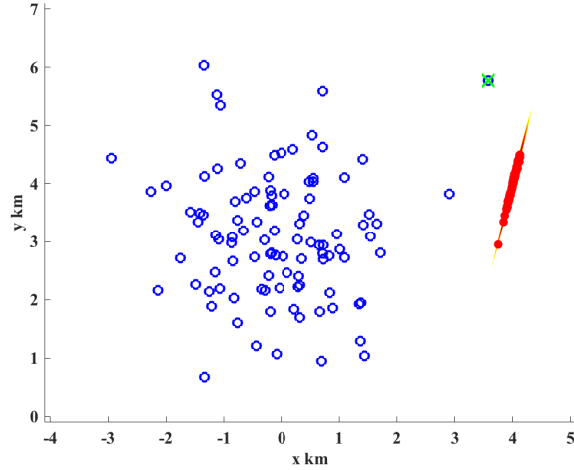


Fig. 1. An example of the collapse of a traditional particle filter given a “surprising” measurement. Open blue circles are the uniform-weighted prior particles. The filled red circles are the particles after updating using the PFF. The colors below the filled circles represent the logarithm of the true posterior PDF. The green X is the particle in a traditional particle filter to which all the weight would collapse after the measurement update.

covariance matrix to use can be obtained from the first two sample moments of the particles. Additionally, only measurements of the form

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v} \quad (1)$$

are considered where $\mathbf{h}$ is a nonlinear (vector) transformation of the target state $\mathbf{x}$ and $\mathbf{v}$ is zero-mean Gaussian noise with covariance matrix $\mathbf{R}$.

After discussing the basis of the filter in Sec. II, the various types of explicit flows considered here are reviewed in Section III. Section IV provides analytic solutions to the flows for a trivial test problem, revealing biases in the explicit and incompressible flows. Thus, only the Gromov flow is used in the rest of this paper. Section V then considers techniques to integrate the flows, including implicit methods. Section VI subsequently considers multiple step-size selection methods for integrating the flows. An new adaptive step-size selection method is introduced as an alternative to that of [61], which is not scale invariant. The importance of the integration algorithm and the step-size selection algorithm are also demonstrated in Section VII. The results are concluded in Sect. VIII.

II. THE NOTION BEHIND PARTICLE FLOW FILTERS

A. Stochastic Particle Filters

Consider the real d_x - and d_z -dimensional column vectors $\mathbf{x}$ and $\mathbf{z}$ to be the target state and a measurement. Define $p_x(\mathbf{x})$ to be the prior PDF of the state, $p_z(\mathbf{z}|\mathbf{x})$ to be the conditional distribution of the measurement given the state (the likelihood function) and $p_x(\mathbf{x}|\mathbf{z})$ to be the posterior distribution of the state given the measurement. Bayes’ theorem states that

$$p_x(\mathbf{x}|\mathbf{z}) = \frac{p_z(\mathbf{z}|\mathbf{x})p_x(\mathbf{x})}{p_z(\mathbf{z})} \quad (2)$$

where for a given measurement $\mathbf{z}$, $p_z(\mathbf{z})$ is effectively a normalizing constant:

$$p_z(\mathbf{z}) = \int_{\mathbf{x} \in \mathbb{R}^{d_x}} p_z(\mathbf{z}|\mathbf{x})p_x(\mathbf{x}) d\mathbf{x}. \quad (3)$$

To approximate (2) with general distributions, rather than discretizing the PDFs on a fixed grid of points where the weights represent the PDF values, particle filters represent the PDF as a set of N discrete points $\mathbf{x}_i$ with associated weights ω_i . Moments of the PDF can be approximated from the weighted samples of the points.

Traditional stochastic particle filters, which first became practicable in [46], as discussed in [2], [43], [64], adjust the weights of the particles. Given a fixed set of particles, one might adjust the i th weight as

$$\omega_i \propto p_z(\mathbf{z}|\mathbf{x}_i)\omega_{i,\text{prior}} \quad (4)$$

where we say $\propto$ rather than $=$ as it is assumed that the weights will all be normalized to sum to 1 after the multiplications. Unfortunately, this is a very bad approach to updating a set of particles. This is illustrated in Fig. 1.

B. Particle Flow Filters

Originally introduced by Daum and Huang [12] in 2007, PFFs differ from stochastic particle filters in two notable points:

- 1) During a measurement update, the weights of the particles never change.
- 2) The particles are moved based on deterministic or stochastic differential equations; they are not randomly resampled.

If the weights never change, particle collapse in the traditional sense is impossible. However, it is possible that the particles might move to regions in space that poorly represent the posterior PDF.

To understand the notion behind PFFs, we shall abbreviate the previous notation as:

$$g(\mathbf{x}) \triangleq p_x(\mathbf{x}) \quad m(\mathbf{x}) \triangleq p_z(\mathbf{z}|\mathbf{x}) \quad (5)$$

$$p(\mathbf{x}) \triangleq p_x(\mathbf{x}|\mathbf{z}) \quad K \triangleq p_z(\mathbf{z}) \quad (6)$$

and take the natural logarithm of both sides of Bayes' theorem from (2):

$$\log p(\mathbf{x}) = \log g(\mathbf{x}) + \log m(\mathbf{x}) - \log K. \quad (7)$$

This assumes that the distributions g and m are nowhere vanishing over a common region. Next, the particle flow defines a scalar homotopy parameter $0 \leq \lambda \leq 1$ and a log-homotopy of the form:

$$\log p(\mathbf{x}, \lambda) = \log g(\mathbf{x}) + \lambda \log m(\mathbf{x}) - \log K(\lambda) \quad (8)$$

where the normalization constant K has been parameterized by λ so that $\exp(\log p(\mathbf{x}, \lambda))$ is a valid PDF for all values of λ . Clearly, $p(\mathbf{x}, 0)$ is the prior distribution and $p(\mathbf{x}, 1)$ is the posterior distribution.

To get to the core of a PFF, we shall first consider a stochastic differential equation of the form

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, \lambda)d\lambda + \mathbf{B}(\mathbf{x}, \lambda)d\mathbf{w}_\lambda \quad (9)$$

where $\mathbf{f}$ is a $d_x \times 1$ drift function and $\mathbf{B}$ is a $d_x \times d_w$ diffusion matrix function. Both functions depend on $\mathbf{x}$ and λ . $d\mathbf{w}_\lambda$ is the $d_w \times 1$ differential of a d_w -dimensional Wiener process where λ is taken as a time parameter. If $\mathbf{B}$ is always zero, then (9) corresponds to the deterministic differential equation:

$$\frac{d\mathbf{x}}{d\lambda} = \mathbf{f}(\mathbf{x}, \lambda). \quad (10)$$

On the other hand, if $\mathbf{B}$ is not zero, then the evolution of a value $\mathbf{x}$ follows a random path.

Stochastic differential equations of the form in (9) are commonly used to model the random dynamics of targets, as discussed in the tutorial [8]. For a particular drift function and diffusion matrix, the Fokker-Planck equation is a partial differential equation that expresses how a PDF evolves over time. The Fokker-Planck equation is [51, Ch. 4.7]:

$$\frac{\partial p(\mathbf{x}, \lambda)}{\partial \lambda} = - \sum_{i=1}^{d_x} \frac{\partial \{p(\mathbf{x}, \lambda)\mathbf{f}(\mathbf{x}, \lambda)\}}{\partial x_i} + \frac{1}{2} \sum_{i=1}^{d_x} \sum_{j=1}^{d_x} \sum_{k=1}^{d_w} \frac{\partial^2 \{p(\mathbf{x}, \lambda) [\mathbf{B}(\mathbf{x}, \lambda)]_{i,k} [\mathbf{B}(\mathbf{x}, \lambda)]_{j,k}\}}{\partial x_i \partial x_j} \quad (11)$$

where $[\mathbf{B}(\mathbf{x}, \lambda)]_{i,k}$ denotes the element in row i and column k of the diffusion matrix.

In the following sections, it is convenient to suppress the parameters from the notation of the functions in (11). Similarly, one often abbreviates the elements of the outer product of $\mathbf{B}$ as

$$\mathbf{Q} \triangleq \mathbf{B}\mathbf{B}^T \quad [\mathbf{Q}]_{i,j} = \sum_{k=1}^{d_w} [\mathbf{B}]_{i,k} [\mathbf{B}]_{j,k} \quad (12)$$

in terms of the process noise covariance matrix $\mathbf{Q}$. This means that the Fokker-Planck equation of (11) is rewritten as:

$$\frac{\partial p}{\partial \lambda} = - \sum_{i=1}^{d_x} \frac{\partial \{p\mathbf{f}\}}{\partial x_i} + \frac{1}{2} \sum_{i=1}^{d_x} \sum_{j=1}^{d_x} \frac{\partial^2 \{p [\mathbf{Q}]_{i,j}\}}{\partial x_i \partial x_j}. \quad (13)$$

In vector notation, this can be written as

$$\frac{\partial p}{\partial \lambda} = -\text{div}(p\mathbf{f}) + \frac{1}{2} \nabla_{\mathbf{x}}^T (p\mathbf{Q}) \nabla_{\mathbf{x}} \quad (14)$$

where the gradient operator is defined as the partial-derivative column vector

$$\nabla_{\mathbf{x}} \triangleq \left[\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots, \frac{\partial}{\partial x_{d_x}} \right]^T \quad (15)$$

and the divergence of a generic vector function $\mathbf{a}$ is defined to be

$$\text{div}(\mathbf{a}) \triangleq \nabla_{\mathbf{x}}^T \mathbf{a}. \quad (16)$$

Given $p(\mathbf{x}, 0)$, the notion behind the PFF is that one chooses $\mathbf{f}$ and $\mathbf{B}$ such that the partial differential equation (9) in terms of the artificial parameter λ has a corresponding Fokker-Planck equation (13) such that integrating from $\lambda = 0$ to $\lambda = 1$ results in $p(\mathbf{x}, 1)$ being identical to the solution of Bayes' theorem in (2). Stochastic dynamic models that produce the desired solution to the Fokker-Planck equation are not unique. Hence, one strives to find models that are both simple and numerically stable when integrated. Note that $\mathbf{x}$ is not independent of λ . Indeed, $\mathbf{f}$ in (10) is the definition of the relationship. Misconceptions regarding the independence of $\mathbf{x}$ from λ and the nature of the Fokker-Planck equation have led authors to draw incorrect conclusions as in [60] and rebutted in [38].

The choice of the flow can affect the utility of the algorithm for different models. For example, the deterministic “incompressible flow” derived in the original papers [12], [13], though not deemed “incompressible” until [14], was noted in [5] to possess singularities. In [19], it is noted that such singularities tend to only be an issue when $\mathbf{x}$ is scalar, since in multiple dimensions the singularities are discrete points and the likelihood of the flow hitting it is rather low. This is illustrated in [18]. Nonetheless, other flows are not subject to the same types of singularities and can be better choices.

III. EXPLICIT FLOWS

Conditions on a flow such that it satisfies Bayes' theorem are given in [33] specifically for the zero-diffusion ($\mathbf{B} = \mathbf{0}$) case. The basic assumptions are that p and g are continuous and nowhere vanishing (except at asymptotically distant points). In practice, most particle flows are approximations so that one can have a computationally tractable solution. Beyond the “incompressible” flow that appeared in the original paper on the PFF [12], a great many other flows have been derived [11], [14]–[17], [20]–[32], [34], [36], [37], [39], [40]. This paper focusses only on flows that have simple, explicit expressions, particularly when using Gaussian noise or approximations in certain circumstances. All such particle flows considered avoid (or omit) the normalization constant. It has been shown that flows including the normalization constant as a term tend not to be as stable as those that avoid it [25].

Subsection III-A focusses on the so-called “Gromov” flow, which is a nonzero diffusion flow. The “geodesic” flow is also discussed as it is just the Gromov flow, neglecting the diffusion term. Subsections III-B and III-C then consider two zero-diffusion flows, the “incompressible” flow and the “exact” flow, which despite its name is an approximate flow.

A. The Geodesic and Gromov Flows

The first flow with nonzero diffusion, dubbed the geodesic flow, was introduced in [23], [25], [26], [29]. The notion of nonzero diffusion was used to simplify the drift function $\mathbf{f}$. However, the flow was integrated deterministically, completely neglecting the diffusion term $\mathbf{B}$, because no expression for it had been found. Without the diffusion term, the geodesic flow is, within a scale factor, equivalent to the zero-curvature particle flow [27]. General approaches for finding $\mathbf{Q}$ (the outer product of the diffusion matrix $\mathbf{B}$ with itself) were then given in [11] with an explicit solution assuming a Gaussian prior. A solution assuming a linear Gaussian-measurement model with a linear prior distribution is given in [36] based on the unpublished work [44]. The Gromov flow is the geodesic flow properly incorporating the diffusion term. The Gromov flow is generally better than the geodesic flow.

Given $\mathbf{Q}$, one has to take the square root¹ to obtain a $\mathbf{B}$ such that $\mathbf{Q} = \mathbf{B}\mathbf{B}^T$. If $\mathbf{Q}$ is not singular, this can be done using a lower-triangular Cholesky decomposition [45, Ch. 4.2.3]. However, the standard Cholesky decomposition algorithm will not work if $\mathbf{Q}$ is singular. In such an instance, a type of LDL decomposition can be used [45, Ch. 4.2.7, Ch. 4.2.8,] to obtain $\mathbf{Q} = \mathbf{L}\mathbf{T}\mathbf{L}^T$, where $\mathbf{T}$ is a diagonal matrix. Thus, $\mathbf{B} = \mathbf{L}\mathbf{T}^{\frac{1}{2}}$, where the square root is an element-by-element square root of the values in $\mathbf{T}$. The values in $\mathbf{T}$ will all be ≥ 0 , because $\mathbf{Q}$ is positive semidefinite. Such a general solution is implemented in the `cholSemiDef` function in the Tracker Component Library (TCL) [9], [67].

Using the explicit $\mathbf{Q}$ given in [36], Gromov's flow derives its names due to the relationship between the flow and a theorem of Gromov [36], [39]–[41]. A derivation of the flow based on [25] is given in Appendix A with the derivation of $\mathbf{Q}$ differing slightly from that of [44].

With the Gaussian approximations mentioned in Section I, $\mathbf{f}$ and $\mathbf{Q}$ are:

$$\mathbf{f} = -(\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1} (\mathbf{h} - \mathbf{z}) \quad (17)$$

$$\mathbf{Q} = (\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} (\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \quad (18)$$

where in practice [54], [58], [62], [63], [65], the matrix $\mathbf{H}$ is taken to be

$$\mathbf{H}^T \triangleq \nabla_{\mathbf{x}} \mathbf{h}^T. \quad (19)$$

¹It has been argued [45, Ch. 4.2.4] that such a matrix transformation should not be deemed a “square root.”

B. The Incompressible Flow

The first flow derived for PFFs is the incompressible flow of [12], [13]. This is a zero-diffusion ($\mathbf{B} = \mathbf{0}$) flow. This flow has been used in many papers including in the implementations of [6], [42]. The flow is based on an approximation, specifically neglecting the normalization constant. Appendix B provides a derivation of the incompressible flow based on the approach taken in [12]. The incompressible flow is:

$$\mathbf{f} = - \left(\frac{\nabla_{\mathbf{x}} \log p}{\|\nabla_{\mathbf{x}} \log p\|^2} \right) \log m. \quad (20)$$

In this format, the singularity issues raised in [5] become clear when the denominator in (20) is zero. Not all flows have this shortcoming. The incompressible flow is very similar to the variational flow derived in [14]. In [14], it is noted that the variation flow is essentially as good as the incompressible flow. Thus, we do not consider the variational flow.

C. The Exact Flow

The exact flow is a zero-diffusion flow that has an explicit solution when neglecting the normalization term of the flow and approximating the prior distribution and measurement distribution as Gaussian. When the measurements are nonlinear transformations of the target state that have been corrupted with noise, one can use local linearization of the measurement function. In such instances, the filter has been referred to as the “localized exact Daum-Huang filter” [59]. The exact flow is discussed in the implementations of [4], [6], [42], [52], [53]. Originally derived in [14], the clearest derivation of the exact flow is given in [53, Appen. 3], and is the basis of the derivation provided in Appendix C, though the derivations are not identical. The exact flow is

$$\mathbf{f} = \mathbf{A}\mathbf{x} + \mathbf{b} \quad (21)$$

$$\mathbf{A} = -\frac{1}{2}\mathbf{P}\mathbf{H}^T (\lambda\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1} \mathbf{H} \quad (22)$$

$$\mathbf{b} = (\mathbf{I} + 2\lambda\mathbf{A}) (\mathbf{A}\hat{\mathbf{x}}_{\text{pred}} - (\mathbf{P}\mathbf{H}^T\mathbf{R}^{-1} + \lambda\mathbf{A}\mathbf{P}\mathbf{H}^T\mathbf{R}^{-1})\boldsymbol{\nu}) \quad (23)$$

$$\boldsymbol{\nu} = \mathbf{h}(\mathbf{x}) - \mathbf{z}. \quad (24)$$

D. The Diagonal Noise Flow

In [11], a flow that is a variant on the Gromov flow is constructed by assuming $\mathbf{Q} = \alpha\mathbf{I}$ where $\mathbf{I}$ is the identity matrix. This assumption leads to a flow of the form

$$\mathbf{f} \approx - \left\{ \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p \right\}^{-1} \left\{ \nabla_{\mathbf{x}} \log m + \nabla_{\mathbf{x}} \text{div}(\hat{\mathbf{f}}) + (\nabla_{\mathbf{x}} \hat{\mathbf{f}}^T) (\nabla_{\mathbf{x}} \log p) - \beta \right\}^T \quad (25)$$

where

$$\beta = \frac{\alpha}{2} \nabla_{\mathbf{x}} \left\{ \text{div}(\nabla_{\mathbf{x}} \log p) + (\nabla_{\mathbf{x}} \log p)^T (\nabla_{\mathbf{x}} \log p) \right\}. \quad (26)$$

Here, $\hat{\mathbf{f}}$ is usually taken to be a linearization of the flow $\mathbf{f}$. In other words, one can use the Exact Flow of Sec. III-C. A special case of the diagonal noise flow arises from the assumption that $\mathbf{f}$ is geodesic. This along with the assumption that $\hat{\mathbf{f}}$ is linear yields a unique expression for α :

$$\alpha \approx \frac{2 \left\| (\nabla_{\mathbf{x}} \hat{\mathbf{f}}^T) (\nabla_{\mathbf{x}} \log p) \right\|}{\left\| \nabla_{\mathbf{x}} \left\{ \text{div}(\nabla_{\mathbf{x}} \log p) + (\nabla_{\mathbf{x}} \log p)^T (\nabla_{\mathbf{x}} \log p) \right\} \right\|}. \quad (27)$$

A derivation of the diagonal noise flow is given in Appendix D.

Though the expression for $\mathbf{f}$ arises in parts of the derivation of (27) in Appendix D, the appendix also uses an unsimplified expression for $\mathbf{f}$ from the Gromov/geodesic flow in one portion. Thus, one could use (27) with either variant of the deterministic flow component.

IV. REVEALING BIASES IN THE FLOWS

Biases in the flows can be revealed analytically. Consider a 2D scenario where the measurement is one of the Cartesian components. That is:

$$h(\mathbf{x}) = \mathbf{H}\mathbf{x} \quad \mathbf{H} = [0, 1]. \quad (28)$$

In this instance, analytic expressions for the flow equations can be found. For the Gromov flow:

$$\mathbf{f} = \begin{bmatrix} 0 \\ -\frac{\sigma^2 \nu}{R + \lambda \sigma^2} \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{R \sigma^4}{(R + \lambda \sigma^2)^2} \end{bmatrix} \quad (29)$$

where the innovation for a particular particle is:

$$\nu = \mathbf{H}\mathbf{x} - z. \quad (30)$$

Under the same assumptions, the incompressible flow is

$$\mathbf{f} = \frac{\log m}{\frac{x^2}{\sigma^4} + \left(\frac{\nu\lambda}{R} + \frac{y}{\sigma^2}\right)^2} \begin{bmatrix} \frac{x}{\sigma^2} \\ \frac{\nu\lambda}{R} + \frac{y}{\sigma^2} \end{bmatrix} \quad (31)$$

and the explicit flow is

$$\mathbf{f} = \begin{bmatrix} 0 \\ -\frac{R\sigma^2(\nu - z) + \nu\lambda\sigma^4}{2(R + \lambda\sigma^2)^2} \end{bmatrix}. \quad (32)$$

Since the measurement equation only provides information for the y component, nothing related to the x component should be changed. However, the incompressible flow in (31) modifies the x component, thus introducing a bias. Consequently, one would not expect this to be an ideal flow to use in high-precision applications.

Considering the explicit flow in (32), suppose that the measurement agrees with the value of the particle, so $\nu = 0$. In this case, the flow is

$$\mathbf{f} = \begin{bmatrix} 0 \\ \frac{R\sigma^2 z}{2(R + \lambda\sigma^2)^2} \end{bmatrix}. \quad (33)$$

The particle flows in a direction that is determined by the value of the measurement z . By changing the origin of the coordinate system used, which changes z , we can make the flow arbitrarily large, thus introducing an arbitrary bias. Consequently, the explicit flow is not a good flow to use.

Finally, let us consider the optimized diagonal noise flow with the assumption that the deterministic part of the flow $\mathbf{f}$ is geodesic, which only leaves the stochastic portion of the flow as potentially biased. Since $\mathbf{Q} = \alpha\mathbf{I}$, the stochastic portion of the flow will clearly introduce biases in a multivariate context. However, is the flow unbiased if the state is univariate?

To answer the question, assume that $\nu = 0$, as was previously done. Then the equation for α becomes

$$\alpha = \frac{-\sigma^2 \left\| (\nabla_{\mathbf{x}} \hat{\mathbf{f}}^T) \mathbf{x} \right\|}{\|\mathbf{x}\|} \quad (34)$$

where $\hat{\mathbf{f}}$ is assumed to be a linearization of $\mathbf{f}$ and $\mathbf{x} = [x, y]^T$. We see here that the use of this optimized $\mathbf{Q} = \alpha\mathbf{I}$ is not independent of the coordinate system origin for the general case that $\nabla_{\mathbf{x}} \hat{\mathbf{f}}^T \neq C\mathbf{x}$ for some $C \in \mathbb{R}$. Therefore, while perhaps useful as a convenient initialization as suggested in [11], the geodesic flow with this $\mathbf{Q}$ would not be ideal. Similarly, this conclusion will hold true for the diagonal noise flow in general, as the lack of independence of the coordinate system origin will be introduced through $\nabla_{\mathbf{x}} \left\{ (\nabla_{\mathbf{x}} \log p)^T (\nabla_{\mathbf{x}} \log p) \right\} = \frac{2\mathbf{x}}{\sigma^4}$ in the expression for β .

These shortcomings of the explicit, incompressible, and diagonal noise flows have not been previously reported. On the other hand, these shortcomings are not exhibited by the Gromov flow. Thus, the Gromov flow is the only flow considered in the rest of this paper.

V. TECHNIQUES FOR INTEGRATING THE FLOW

In many papers on homotopy particle filtering, particularly when using the Gromov/geodesic (and similar) flows, it has been repeatedly noted that the resulting differential equation or stochastic differential equation is stiff [11], [31], [32], [34], [35], [40], [53], [55], [56], [61]. When considering the Gromov flow, only techniques for integrating stochastic differential equations are considered. In [31], the best methods for addressing such concerns are discussed. The primary solutions suggested are computing the flow in principal coordinates and the use of various step-size selection techniques. The use of implicit methods for solving the stochastic differential equations are mentioned, but their performance is not analyzed (it is listed as uncertain). Here, we consider a number of simple explicit and implicit techniques that will be compared in Section VII. We limit our consideration to low-order methods due to their simplicity and also because [40] reported very bad results with a fourth-order Runge-Kutta technique.

We consider four integration techniques (These and others are in the TCL [67]). Only the first has been previously considered for use with PFFs:

- 1) The explicit strong Euler-Maruyama method [57, Ch. 10.2]. An integration step to go from time λ to time $\lambda + \Delta$ is:

$$\mathbf{x}_{\lambda+\Delta} = \mathbf{x}_{\lambda} + \mathbf{f}(\mathbf{x}_{\lambda}, \lambda)\Delta + \mathbf{B}(\mathbf{x}_{\lambda}, \lambda)\tilde{\mathbf{w}} \quad (35)$$

where the arguments of $\mathbf{f}$ and $\mathbf{B}$ are explicitly written and $\tilde{\mathbf{w}}$ is a zero-mean Gaussian random variable with covariance matrix $\Delta \mathbf{I}$.

- 2) The strong semi-implicit Euler-Maruyama method [57, Ch. 12.2]:

$$\mathbf{x}_{\lambda+\Delta} = \mathbf{x}_\lambda + (\alpha \mathbf{f}(\mathbf{x}_{\lambda+\Delta}, \lambda + \Delta) + (1 - \alpha) \mathbf{f}(\mathbf{x}_\lambda, \lambda)) \Delta + \mathbf{B}(\mathbf{x}_\lambda, \lambda) \tilde{\mathbf{w}}. \quad (36)$$

The parameter α adjusts the level of implicitness. We only consider $\alpha = 1$.

- 3) The strong split-step backward Euler-Maruyama method [49]:

$$\tilde{\mathbf{x}}_{\lambda+\Delta} = \mathbf{x}_\lambda + \mathbf{f}(\tilde{\mathbf{x}}_{\lambda+\Delta}, \lambda + \Delta) \Delta \quad (37)$$

$$\mathbf{x}_{\lambda+\Delta} = \tilde{\mathbf{x}}_{\lambda+\Delta} + \mathbf{B}(\mathbf{x}_\lambda, \lambda) \tilde{\mathbf{w}}. \quad (38)$$

- 4) The strong error-corrected Euler-Maruyama method [66]:

$$\tilde{\mathbf{x}}_{\lambda+\Delta} = \mathbf{x}_\lambda + \mathbf{f}(\mathbf{x}_\lambda, \lambda) \Delta + \mathbf{B}(\mathbf{x}_\lambda, \lambda) \tilde{\mathbf{w}} \quad (39)$$

$$\mathbf{x}_{\lambda+\Delta} = \tilde{\mathbf{x}}_{\lambda+\Delta} + (\mathbf{I} - \nabla_{\mathbf{x}} \{ \mathbf{f}(\tilde{\mathbf{x}}_{\lambda+\Delta}, \lambda + \Delta)^T \} \Delta)^{-1} (\mathbf{f}(\tilde{\mathbf{x}}_{\lambda+\Delta}, \lambda + \Delta) - \mathbf{f}(\mathbf{x}_\lambda, \lambda)), \quad (40)$$

where the inverse can be a Moore-Penrose pseudoinverse [45, Ch. 5.5.2] if the matrix is singular.

For the implicit algorithms, (ones with $\mathbf{x}_{\lambda+\Delta}$ on both sides of the equation), the strong Euler-Maruyama method of (35) is used to get the initial estimate. One can then either perform a fixed-point iteration or iterate Newton's method to solve the equations. Fixed-point iteration involves repeatedly inserting the estimate into the right-hand side of the equation to get a new value of $\mathbf{x}_{\lambda+\Delta}$ and then repeating. For Newton's method, given an implicit equation of the form

$$\mathbf{c}_0 + c_1 \mathbf{a}(\mathbf{y}) - \mathbf{y} = \mathbf{0} \quad (41)$$

for some function $\mathbf{a}$ and constants $\mathbf{c}_0$ and c_1 , the k th iteration is

$$\mathbf{y}_{k+1} = \mathbf{y}_k - (\mathbf{I} - c_0 \nabla_{\mathbf{x}} \{ \mathbf{a}(\mathbf{y})^T \})^{-1} (\mathbf{y} - \mathbf{c}_0 - c_1 \mathbf{a}(\mathbf{y})). \quad (42)$$

VI. STEP-SIZE SELECTION FOR INTEGRATING FLOWS

We consider three techniques for step-size selection:

- 1) Use a uniform step size. For N steps,

$$\Delta = \frac{1}{N}. \quad (43)$$

- 2) Use an exponentially increasing step size, as suggested in [25], [31]. The k th step size is

$$\Delta_k = s_0 b^k \quad (44)$$

where b is the base to use, in this paper we choose $b = 2$, and s_0 is the initial step size. To take N exponential steps and end at $\lambda = 1$, s_0 must be:

$$s_0 = \frac{b - 1}{b^N - 1}. \quad (45)$$

- 3) Use an adaptive technique based on the notion that the distance traveled by a particle in a step should not be a significant fraction of the spread of the particles. This notion was presented in [61]. However, the resulting technique is not scale invariant making it completely unusable in many instances. For example, reproducing the scenario in [61], but multiply all distances by 10^6 (changing the units) completely ruins the results. Here, for the Gromov flow where $\mathbf{B}$ comes from a decomposition of a square-matrix $\mathbf{Q}$, we propose a modified, scale-invariant heuristic:

$$\Delta_k = \min \left\{ 1 - \lambda_k, \frac{\min_i \left\{ \sqrt{\frac{P_{i,i}}{\max_{l,j} \{ B_{i,l}(\mathbf{x}_j, \lambda)^2 + f_i(\mathbf{x}_j, \lambda)^2 \}}} \right\}}{N} \right\}. \quad (46)$$

Here N is the minimum number of steps to take. The scale of the problem is controlled by P . Note that this heuristic is invariant to a translation of the coordinate-system origin.

VII. CONSIDERING THE EFFECT OF INTEGRATION AND STEP-SIZE SELECTION METHODS

To consider the effects of the choice of the integration and step-size selection algorithms on the performance of the PFF, we consider another carefully selected problem for which an analytic solution to the flow can be found. Specifically, we consider a scenario where the target state is a 2D Cartesian position and the prior distribution is zero-mean Gaussian ($\hat{\mathbf{x}}_{\text{pred}} = \mathbf{0}$) with

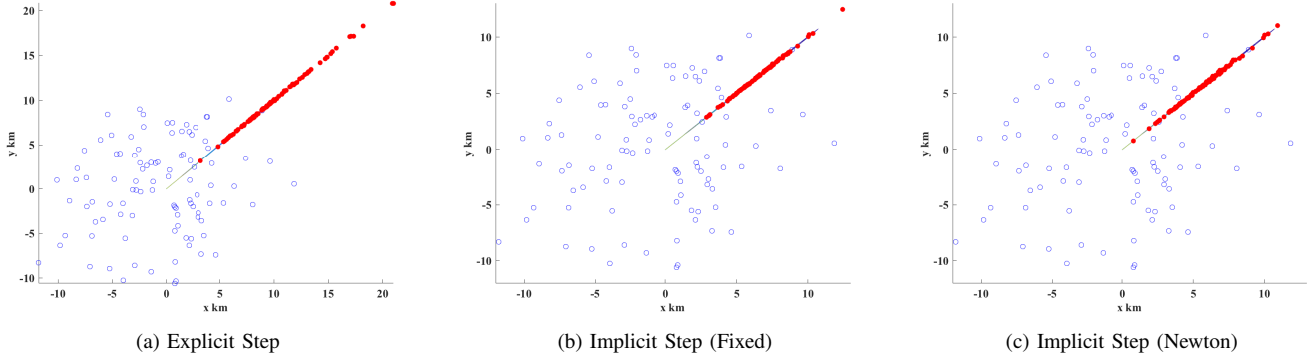


Fig. 2. The particles after a polar angle-only measurement, given a Gaussian prior using the Gromov flow and different integration techniques. Prior particles are blue circles; posterior particles are red-filled circles. A contour of the posterior PDF is the line. The same random seed and integration step sizes are used in all three plots. As described in the text, optimally, the measurement update will only rotate the particles. With an explicit Euler method, the particles are biased far away from the origin (note the different scale than the other two plots). With an implicit Euler method using two fixed-point iterations, the range bias is greatly reduced and with an implicit Euler method using Newton iterations, the bias is reduced further.

diagonal covariance matrix $\mathbf{P} = \sigma^2 \mathbf{I}$. The measurement is just a scalar direction of arrival (DOA). If the state is $\mathbf{x} = [x, y]^T$, then the measurement function and the partial derivatives going into the gradient are:

$$h(\mathbf{x}) = \text{atan2}(y, x) \quad \frac{\partial h}{\partial x} = -\frac{y}{\|\mathbf{x}\|^2} \quad \frac{\partial h}{\partial y} = \frac{x}{\|\mathbf{x}\|^2} \quad (47)$$

where atan2 is a four-quadrant inverse-tangent function. Given this model, substituting into the equations for the Gromov flow, one finds that the flow is:

$$\mathbf{f} = \left(\frac{\sigma^2 \nu}{R\|\mathbf{x}\|^2 + \lambda\sigma^2} \right) \begin{bmatrix} y \\ -x \end{bmatrix} \quad (48)$$

$$\mathbf{Q} = \left(\frac{R\sigma^4}{(R\|\mathbf{x}\|^2 + \lambda\sigma^2)^2} \right) \begin{bmatrix} y^2 & -xy \\ -xy & x^2 \end{bmatrix} \quad (49)$$

where z is the measured angle and ν is the innovation

$$\nu = h(\mathbf{x}) - z. \quad (50)$$

A lower-triangular Cholesky decomposition of $\mathbf{Q}$ that can provide the drift $\mathbf{B}$ from the stochastic differential equation is (9):

$$\mathbf{B} = \frac{R^{\frac{1}{2}}\sigma^2}{R\|\mathbf{x}\|^2 + \lambda\sigma^2} \begin{bmatrix} y & 0 \\ -x & 0 \end{bmatrix}. \quad (51)$$

The singularity of the matrix $\mathbf{Q}$ is clear from the all-zero column of $\mathbf{B}$.

The drift function $\mathbf{f}$ in (48) acts *orthogonally* to the state $\mathbf{x} = [x, y]^T$ (the radial direction). Similarly, regardless of the value of the differential Wiener process, the diffusion matrix in (51) ensures that the noise contribution acts orthogonally to the state. The consequence of this is that for any initial state, an exact integration of the diffusion equation in (9) will result in a pure rotation of the particles. Any change in $\|\mathbf{x}\|$ for any of the particles can be attributed to a shortcoming in the stochastic numeric-integration technique used. This drift in range can be used to assess the quality of the step-size selection and integration techniques.

The scenario considered here uses $\sigma = 5$ km and $R = 2 \times 10^{-3}$ rad and the target at $\mathbf{x} = [3 \text{ km}, 3 \text{ km}]^T$. One hundred particles are used. One Monte-Carlo run of the PFF with the Gromov flow considering two different integration methods and one alternate coordinate system is shown in Fig. 2. In all three instances, the step sizes are selected exponentially with base $b = 2$, and 20 steps are taken. The true posterior PDF is given by the line in the plots² in Fig. 2. In Fig. 2a, where an explicit strong Euler step is taken, one can see that the particles are severely biased away from the origin. In Fig. 2b, where an implicit Euler step is taken, performing two fixed-point iterations, one can see that the particles are biased, but to a much lesser extent and the bias becomes smaller in (2c) where two Newton iterations are used, with an outlier in the upper-right coming closer. In all instances, the filters were given the exact $\mathbf{P}$, not the value obtained from the particles. The diagonality of this $\mathbf{P}$ ensures that the ideal flow is orthogonal to the range.

²The true posterior PDF was obtained by discretizing the prior and posterior and performing normalized multiplication.

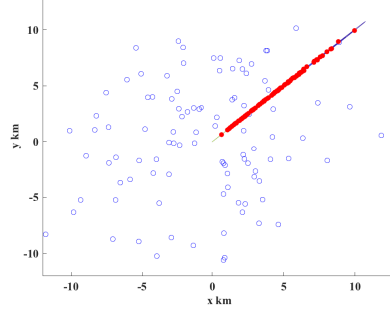


Fig. 3. The particles after a polar angle-only measurement, given a Gaussian prior, whereby the particles were converted to polar coordinates, the update was applied, and then the particles were converted back to Cartesian coordinates. Unlike those in Fig. 2, the bias is minimal.

TABLE I

THE AVERAGE MAXIMUM OFFSET IN RANGE (IN KM) OF INDIVIDUAL PARTICLES FOR THE GROMOV FLOW OF A DOA-ONLY MEASUREMENT WITH A CARTESIAN STATE TO THREE DIGITS. DIFFERENT VARIANTS OF THE INTEGRATION METHODS OF SECTION V AND THE DIFFERENT STEP-SIZE SELECTION ALGORITHMS OF SECTION VI ARE COMPARED. N DENOTES NEWTON ITERATION; F DENOTES FIXED-POINT ITERATION.

	Uniform	Exponential	Adaptive
Explicit	1.60×10^6	50.2	12.8
Semi-Implicit (N)	7.06×10^3	26.5	10.2
Semi-Implicit (F)	7.13×10^3	26.7	10.6
Split-Step Backward (N)	388	9.26	5.14
Split-Step Backward (F)	13.6	5.55	5.80
Error-Corrected	6.95×10^3	26.2	9.63

On the other hand, the performance of the algorithm can be greatly improved with a coordinate-system change during the update:

- 1) Convert the particles to polar coordinates.
- 2) Perform the (linear) measurement update in polar coordinates.
- 3) Convert the particles back to Cartesian coordinates.

In this instance, the distribution of the particles in polar coordinates is not Gaussian, but we use an approximate Gaussian prior, where the value of $\mathbf{P}$ is the sample covariance of the particles. The resulting plot is given in Fig. 3 and is similar to Fig. 2c. However, the maximum bias of the particles is 6.6 m (replacing the sample $\mathbf{P}$ in polar coordinates with a diagonal value makes the bias 0). This compares to maximum biases of 26.4733 km, 16.2975 km, and 14.2301 km for the scenarios in Figs. 2a, 2b, and 2c, respectively. The dramatic performance improvement by transforming to polar coordinates is consistent with similar results with other filters [1], [50].

Though the integration of the flow is dramatically worse when performing the measurement update using the state in Cartesian coordinates compared to polar coordinates, the polar scenario in Cartesian coordinates can be used as a test of how robust the integration and step-size selection techniques of Sections V and VI are. Table I compares the different techniques over 1,000 Monte Carlo runs when looking at the average of the maximum range offset of a particle. When using a uniform step size $N = 300$ steps were used; when using an exponential step size $N = 20$, and when using the adaptive step size $N = 2$ and the maximum number of steps was limited to 150.³ It can be seen that the explicit Euler-Maruyama method performed the worst of all the techniques and the uniform step size was the overall worst.

VIII. CONCLUSION

It was shown that the explicit, incompressible, and diagonal flows are biased with linear dynamic models, whereas the Gromov flow is unbiased. A new adaptive step-size selection heuristic for the Gromov flow was introduced. Unlike the heuristic of [61], the heuristic of this paper is scale invariant. It was demonstrated on a polar scenario that using implicit integration techniques on the Gromov flow outperforms explicit Euler integration. The utility of implicit techniques was deemed “uncertain” in [31]. However, when considering an update with a DOA measurement, for which an explicit expression is obtained using the expressions of [36], the improvement from implicit methods is dwarfed by the improvement in performing the update in a different coordinate system. This is consistent with the suggestion in [31] to use principal coordinates, which in this instance means using polar coordinates.

The PFF succeeds where traditional particle filters suffer from particle collapse. However, it cannot be used as a generic “black box” into which one can shove arbitrary measurement models. Rather, to avoid biased estimates, some level of design

³If more than 150 steps would normally be necessary, the final step size is set to $1 - \lambda$ to force it to step to the end.

must go into the coordinate system used for the measurement update as well as the technique used to integrate the homotopy. This paper has demonstrated the value in the use of implicit stochastic integration and has provided an improved step-size selection technique.

Derivations of the flows are given in the appendices. The derivation of the $\mathbf{Q}$ matrix of the Gromov flow in Appendix A is the first published instance of this derivation, because the prior literature just references the unpublished work in [44].

ACKNOWLEDGEMENTS

This research is supported by the Office of Naval Research (ONR) through the Naval Research Laboratory (NRL) Base Program. The author would also like to thank Fred Daum of the Raytheon company for answering questions on his algorithms and for sharing the notes from [44].

APPENDIX A THE DERIVATION OF THE GROMOV FLOW

This appendix derives the Gromov flow in a manner similar but not identical to [36], [44]. To derive $\mathbf{f}$, begin by taking the partial derivative of the flow in (8)

$$\frac{\partial \log p}{\partial \lambda} = \log m - \frac{\partial \log K}{\partial \lambda}. \quad (52)$$

Next, use the identity of the partial derivative of the logarithm of a generic function a that

$$\frac{\partial \log a}{\partial \lambda} = \frac{1}{a} \frac{\partial a}{\partial \lambda} \quad (53)$$

to get

$$\frac{\partial p}{\partial \lambda} = p \left(\log m - \frac{\partial \log K}{\partial \lambda} \right). \quad (54)$$

Substituting (54) into (14), one gets

$$p \left(\log m - \frac{\partial \log K}{\partial \lambda} \right) = -\operatorname{div}(p\mathbf{f}) + \frac{1}{2} \nabla_{\mathbf{x}}^T (p\mathbf{Q}) \nabla_{\mathbf{x}}. \quad (55)$$

The chain rule can be used to expand the divergence term as

$$\operatorname{div}(p\mathbf{f}) = p \operatorname{div}(\mathbf{f}) + (\nabla_{\mathbf{x}} p)^T \mathbf{f}. \quad (56)$$

Substituting (56) into (55) and using (53), leads to

$$\log m - \frac{\partial \log K}{\partial \lambda} = -\operatorname{div}(\mathbf{f}) - (\nabla_{\mathbf{x}} \log p)^T \mathbf{f} + \frac{1}{2p} \nabla_{\mathbf{x}}^T (p\mathbf{Q}) \nabla_{\mathbf{x}}. \quad (57)$$

Next, take the gradient of both sides of (57), which eliminates the normalization constant, and making use of the chain rule, leads to

$$\nabla_{\mathbf{x}} \log m = -\nabla_{\mathbf{x}} \operatorname{div}(\mathbf{f}) - (\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p) \mathbf{f} - (\nabla_{\mathbf{x}} \mathbf{f}^T) (\nabla_{\mathbf{x}} \log p) + \nabla_{\mathbf{x}} \left\{ \frac{1}{2p} \nabla_{\mathbf{x}}^T (p\mathbf{Q}) \nabla_{\mathbf{x}} \right\}. \quad (58)$$

The matrix $\mathbf{Q}$ is specially chosen such that

$$-\nabla_{\mathbf{x}} \operatorname{div}(\mathbf{f}) - (\nabla_{\mathbf{x}} \mathbf{f}^T) (\nabla_{\mathbf{x}} \log p) + \nabla_{\mathbf{x}} \left\{ \frac{1}{2p} \nabla_{\mathbf{x}}^T (p\mathbf{Q}) \nabla_{\mathbf{x}} \right\} = \mathbf{0}. \quad (59)$$

Consequently, the flow drift function from (58) becomes

$$\mathbf{f} = -(\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p)^{-1} \nabla_{\mathbf{x}} \log m. \quad (60)$$

For the situations considered in this paper, the measurement model is taken to be a nonlinear transformation of the state with additive Gaussian noise:

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v} \quad (61)$$

where $\mathbf{v}$ is zero-mean Gaussian noise with $d_z \times d_z$ covariance matrix $\mathbf{R}$, $\mathbf{h}$ is a (typically nonlinear) function of the target state, and $\mathbf{z}$ is a $d_z \times 1$ vector. In such an instance, the logarithm of the measurement PDF is

$$\log m = -\frac{1}{2} \log(\|2\pi\mathbf{R}\|) - \frac{1}{2} (\mathbf{h} - \mathbf{z})^T \mathbf{R}^{-1} (\mathbf{h} - \mathbf{z}). \quad (62)$$

As in (19), define

$$\mathbf{H}^T \triangleq \nabla_{\mathbf{x}} \mathbf{h}^T \quad (63)$$

with the corresponding gradient and Hessian (omitting the argument of $\mathbf{h}$):

$$\nabla_{\mathbf{x}} \log m = -\mathbf{H}^T \mathbf{R}^{-1} (\mathbf{h} - \mathbf{z}) \quad (64)$$

$$\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log m = -\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} - \mathbf{C} \quad (65)$$

where column j of $\mathbf{C}$ is

$$\mathbf{C}_{:,j} = \frac{\partial}{\partial x_j} \{ \mathbf{H}^T \} \mathbf{R}^{-1} (\mathbf{h} - \mathbf{z}). \quad (66)$$

Further approximating the prior distribution g as Gaussian with mean $\hat{\mathbf{x}}_{\text{pred}}$ and covariance matrix $\mathbf{P}$, means that

$$\log g = -\frac{1}{2} \log (\|2\pi\mathbf{P}\|) - \frac{1}{2} (\mathbf{x} - \hat{\mathbf{x}}_{\text{pred}})^T \mathbf{P}^{-1} (\mathbf{x} - \hat{\mathbf{x}}_{\text{pred}}). \quad (67)$$

The gradient and Hessian are thus

$$\nabla_{\mathbf{x}} \log g = -\mathbf{P}^{-1} (\mathbf{x} - \hat{\mathbf{x}}_{\text{pred}}) \quad (68)$$

$$\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log g = -\mathbf{P}. \quad (69)$$

This means that the gradient and Hessian of the flow equation (8) are

$$\nabla_{\mathbf{x}} \log p = -\mathbf{P}^{-1} (\mathbf{x} - \hat{\mathbf{x}}_{\text{pred}}) - \lambda \mathbf{H}^T \mathbf{R}^{-1} (\mathbf{h} - \mathbf{z}) \quad (70)$$

$$\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p = -\mathbf{P}^{-1} - \lambda (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} + \mathbf{C}). \quad (71)$$

Consequently, the drift function of the flow from (60) simplifies to

$$\mathbf{f} = -(\mathbf{P}^{-1} + \lambda (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} + \mathbf{C}))^{-1} \mathbf{H}^T \mathbf{R}^{-1} (\mathbf{h} - \mathbf{z}). \quad (72)$$

When used with $\mathbf{Q} = \mathbf{0}$ (which does not satisfy (59), but lets one integrate (10) instead of (9)), the flow is deemed the geodesic flow. Next, the derivation of an explicit solution for $\mathbf{Q}$ is considered.

The derivation of $\mathbf{Q}$ starts with assumptions that the noise corrupting the measurement is Gaussian, the measurement is a linear transformation of the target state, and the prior distribution is Gaussian. This changes (61) to

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v}. \quad (73)$$

Consequently, $\mathbf{C}$ from (66) becomes $\mathbf{0}$, but all of the previous equations in the derivation of $\mathbf{f}$ in (72) used here still hold.

The matrix $\mathbf{Q}$ is specified by (59). The derivation starts by assuming that $\mathbf{Q}$ does not depend on $\mathbf{x}$. We start by performing simplifications that allow the last term in (59) to be expressed in a simple manner. Under the constant $\mathbf{Q}$ assumption

$$\frac{\partial^2 \{p[\mathbf{Q}]_{i,j}\}}{\partial x_i \partial x_j} = \frac{\partial^2 p}{\partial x_i \partial x_j} [\mathbf{Q}]_{i,j}. \quad (74)$$

Using the identity that

$$\sum_{i=1}^{d_x} \sum_{j=1}^{d_x} \frac{\partial^2 p}{\partial x_i \partial x_j} [\mathbf{Q}]_{i,j} = \text{Tr} \{ \mathbf{Q} (\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T p) \}, \quad (75)$$

multiply by $1/p$:

$$\frac{1}{p} \sum_{i=1}^{d_x} \sum_{j=1}^{d_x} \frac{\partial^2 p}{\partial x_i \partial x_j} [\mathbf{Q}]_{i,j} = \text{Tr} \left\{ \mathbf{Q} \left(\frac{1}{p} \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T p \right) \right\} \quad (76)$$

$$= \text{Tr} \left\{ \mathbf{Q} \left(\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p + (\nabla_{\mathbf{x}} \log p) (\nabla_{\mathbf{x}} \log p)^T \right) \right\} \quad (77)$$

and differentiate with respect to x_k :

$$\frac{\partial}{\partial x_k} \left\{ \frac{1}{p} \sum_{i=1}^{d_x} \sum_{j=1}^{d_x} \frac{\partial^2 p}{\partial x_i \partial x_j} [\mathbf{Q}]_{i,j} \right\} = \text{Tr} \left\{ \mathbf{Q} \left(\frac{\partial}{\partial x_k} \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p + \frac{\partial}{\partial x_k} \left\{ (\nabla_{\mathbf{x}} \log p) (\nabla_{\mathbf{x}} \log p)^T \right\} \right) \right\}. \quad (78)$$

Due to the linear, Gaussian approximations applied to m and g , third-order derivatives of $\log p$ are zero. Using this fact and the fact that matrix products in a trace can be reordered, results in:

$$\frac{\partial}{\partial x_k} \left\{ \frac{1}{p} \sum_{i=1}^{d_x} \sum_{j=1}^{d_x} \frac{\partial^2 p}{\partial x_i \partial x_j} [\mathbf{Q}]_{i,j} \right\} = \frac{\partial}{\partial x_k} \text{Tr} \left\{ \mathbf{Q} \left((\nabla_{\mathbf{x}} \log p) (\nabla_{\mathbf{x}} \log p)^T \right) \right\} \quad (79)$$

$$= \frac{\partial}{\partial x_k} \text{Tr} \left\{ (\nabla_{\mathbf{x}} \log p)^T \mathbf{Q} (\nabla_{\mathbf{x}} \log p) \right\} \quad (80)$$

$$= \frac{\partial}{\partial x_k} (\nabla_{\mathbf{x}} \log p)^T \mathbf{Q} (\nabla_{\mathbf{x}} \log p). \quad (81)$$

Thus, in vector form, the final term of the constraint on $\mathbf{Q}$ from (59) becomes:

$$\nabla_{\mathbf{x}} \left\{ \frac{1}{p} \sum_{i=1}^{d_x} \sum_{j=1}^{d_x} \frac{\partial^2 p}{\partial x_i \partial x_j} [\mathbf{Q}]_{i,j} \right\} = \nabla_{\mathbf{x}} \left\{ (\nabla_{\mathbf{x}} \log p)^T \mathbf{Q} (\nabla_{\mathbf{x}} \log p) \right\} \quad (82)$$

$$= 2 (\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p) \mathbf{Q} (\nabla_{\mathbf{x}} \log p). \quad (83)$$

Substituting into (59) and noting that under the linear Gaussian assumptions, $\nabla_{\mathbf{x}} \text{div}(\mathbf{f}) = \mathbf{0}$, results in the condition:

$$- (\nabla_{\mathbf{x}} \mathbf{f}^T) (\nabla_{\mathbf{x}} \log p) + (\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p) \mathbf{Q} (\nabla_{\mathbf{x}} \log p) = \mathbf{0}. \quad (84)$$

By equating common terms, it can be said that

$$\nabla_{\mathbf{x}} \mathbf{f}^T = (\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p) \mathbf{Q} \quad (85)$$

$$\mathbf{Q} = (\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p)^{-1} \nabla_{\mathbf{x}} \mathbf{f}^T. \quad (86)$$

Consequently, $\mathbf{f}$ and $\mathbf{Q}$ under a Gaussian prior approximation and a linear/Gaussian measurement approximation are given in (17) and (18).

APPENDIX B THE INCOMPRESSIBLE FLOW

This appendix derives the incompressible flow in a manner similar to [12]. The derivation starts by taking the total derivative of the left-hand of (8) with respect to λ , keeping in mind that $\mathbf{x}$ depends on λ . The incompressible condition used is that the likelihood will not change as λ changes ($\mathbf{x}$ moves to make that so). Thus, using the law of total derivatives,

$$\frac{d \log p}{d \lambda} = (\nabla_{\mathbf{x}}^T \log p) \mathbf{f} + \frac{\partial p}{\partial \lambda} = 0 \quad (87)$$

where $\mathbf{f} = \frac{\partial \mathbf{x}}{\partial \lambda}$ from (10). Except in the case of $\mathbf{x}$ being scalar, the solution of the flow $\mathbf{f}$ in (87) is not unique.

The Moore-Penrose pseudoinverse of an arbitrary $m \times n$ matrix $\mathbf{A}$ is the $n \times m$ matrix $\mathbf{A}^\dagger$ such that [45, Ch. 5.5.2]

$$\mathbf{A}^\dagger = \arg \min_{\mathbf{X} \in \mathbb{R}^{m \times n}} \|\mathbf{A}\mathbf{X} - \mathbf{I}\|_F \quad (88)$$

where the norm taken is the Frobenius norm (square root of the sum of the squares of all of the elements), and $\mathbf{I}$ is an $m \times m$ identity matrix. The Moore-Penrose pseudoinverse can be found with the `pinv` command in Matlab and can be implemented in a number of ways [45, Ch. 5.5]. It is worth noting that if $\mathbf{A}$ is rank n , then the pseudoinverse of a real matrix is [45, Ch. 5.5.2]

$$\mathbf{A}^\dagger = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \quad (89)$$

and when $\mathbf{A}$ is a vector, $\mathbf{a}$, (89) becomes

$$\mathbf{a}^\dagger = \frac{\mathbf{a}^T}{\|\mathbf{a}\|^2}. \quad (90)$$

In the incompressible flow, the Moore-Penrose pseudoinverse is chosen as a unique solution to solve for $\mathbf{f}$ resulting in

$$\mathbf{f} = - (\nabla_{\mathbf{x}}^T \log p)^\dagger \frac{\partial p}{\partial \lambda} \quad (91)$$

or using (90)

$$\mathbf{f} = - \left(\frac{\nabla_{\mathbf{x}} \log p}{\|\nabla_{\mathbf{x}} \log p\|^2} \right) \frac{\partial p}{\partial \lambda}. \quad (92)$$

Note that $\nabla_{\mathbf{x}}^T \log p$ is the prior distribution, and should thus be known. The unknown quantity $\frac{\partial p}{\partial \lambda}$ is approximated by taking the partial derivative of (8) with respect to λ , *neglecting the normalizing constant*. Thus,

$$\frac{\partial p}{\partial \lambda} \approx \log m. \quad (93)$$

All together, the incompressible flow is given in (20).

APPENDIX C THE EXACT FLOW

The derivation begins with the assumption that the flow takes the form

$$\mathbf{f} = \mathbf{A}(\lambda)\mathbf{x} + \mathbf{b}(\lambda) \quad (94)$$

where the λ arguments of $\mathbf{A}$ and $\mathbf{b}$ will be suppressed in further equations. Next, one uses (57) with $\mathbf{Q} = \mathbf{0}$ omitting the derivative of the normalizing constant K with respect to λ . Solving for the divergence of the flow leads to

$$\text{div}(\mathbf{f}) = -\log m - (\nabla_{\mathbf{x}} \log p)^T \mathbf{f}. \quad (95)$$

Given (94),

$$\text{div}(\mathbf{f}) = \text{Tr} \{ \mathbf{A} \}. \quad (96)$$

The prior distribution is assumed to be Gaussian as in (67) and the measurement is assumed to be a nonlinear transformation of the state corrupted with Gaussian noise as in (61), and (62). Substituting (96), (62), (94), and the gradient of p from (70) into (95) leads to

$$\text{Tr} \{ \mathbf{A} \} = \frac{1}{2} \log (\|2\pi\mathbf{R}\|) + \frac{1}{2} (\mathbf{h} - \mathbf{z})^T \mathbf{R}^{-1} (\mathbf{h} - \mathbf{z}) + (\mathbf{Ax} + \mathbf{b})^T (\mathbf{P}^{-1} (\mathbf{x} - \hat{\mathbf{x}}_{\text{pred}}) + \lambda \mathbf{H}^T \mathbf{R}^{-1} (\mathbf{h} - \mathbf{z})). \quad (97)$$

Next, for nonlinear functions, a first-order Taylor series expansions of $\mathbf{h}$ is taken about a point $\mathbf{x}_0$. In practice with particle, $\mathbf{x}_0$ will be the initial value of the particle prior to starting the flow. Thus:

$$\mathbf{h}(\mathbf{x}) \approx \mathbf{h}(\mathbf{x}_0) + \mathbf{H}(\mathbf{x} - \mathbf{x}_0) \quad (98)$$

$$= \mathbf{H}\mathbf{x} + \underbrace{(\mathbf{h}(\mathbf{x}_0) - \mathbf{H}\mathbf{x}_0)}_{\gamma_0}. \quad (99)$$

Defining

$$\tilde{\mathbf{z}} \triangleq \mathbf{z} - \gamma_0 \quad (100)$$

and using the first-order Taylor series approximation, Eq. (97) becomes

$$\text{Tr} \{ \mathbf{A} \} = \frac{1}{2} \log (\|2\pi\mathbf{R}\|) + \frac{1}{2} (\mathbf{H}\mathbf{x} - \tilde{\mathbf{z}})^T \mathbf{R}^{-1} (\mathbf{H}\mathbf{x} - \tilde{\mathbf{z}}) + (\mathbf{Ax} + \mathbf{b})^T (\mathbf{P}^{-1} (\mathbf{x} - \hat{\mathbf{x}}_{\text{pred}}) + \lambda \mathbf{H}^T \mathbf{R}^{-1} (\mathbf{H}\mathbf{x} - \tilde{\mathbf{z}})). \quad (101)$$

Expanding the terms

$$\begin{aligned} \text{Tr} \{ \mathbf{A} \} = & \frac{1}{2} \log (\|2\pi\mathbf{R}\|) + \frac{1}{2} \mathbf{x}^T \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{x} - \frac{1}{2} \mathbf{x}^T \mathbf{H}^T \mathbf{R}^{-1} \tilde{\mathbf{z}} - \frac{1}{2} \tilde{\mathbf{z}}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{x} + \frac{1}{2} \tilde{\mathbf{z}}^T \mathbf{R}^{-1} \tilde{\mathbf{z}} \\ & + \mathbf{x}^T \mathbf{A}^T \mathbf{P}^{-1} \mathbf{x} - \mathbf{x}^T \mathbf{A}^T \mathbf{P}^{-1} \hat{\mathbf{x}}_{\text{pred}} + \mathbf{b}^T \mathbf{P}^{-1} \mathbf{x} - \mathbf{b}^T \mathbf{P}^{-1} \hat{\mathbf{x}}_{\text{pred}} \\ & + \lambda \mathbf{x}^T \mathbf{A}^T \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{x} + \lambda \mathbf{b}^T \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{x} - \lambda \mathbf{x}^T \mathbf{A}^T \mathbf{H}^T \mathbf{R}^{-1} \tilde{\mathbf{z}} - \lambda \mathbf{b}^T \mathbf{H}^T \mathbf{R}^{-1} \tilde{\mathbf{z}}. \end{aligned} \quad (102)$$

Simplifying,

$$\begin{aligned} & \lambda \mathbf{b}^T \mathbf{H}^T \mathbf{R}^{-1} \tilde{\mathbf{z}} + \mathbf{b}^T \mathbf{P}^{-1} \hat{\mathbf{x}}_{\text{pred}} - \frac{1}{2} \tilde{\mathbf{z}}^T \mathbf{R}^{-1} \tilde{\mathbf{z}} + \text{Tr} \{ \mathbf{A} \} - \frac{1}{2} \log (\|2\pi\mathbf{R}\|) = \\ & \mathbf{x}^T \left(\mathbf{A}^T \mathbf{P}^{-1} + \lambda \mathbf{A}^T \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} + \frac{1}{2} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \right) \mathbf{x} \\ & + \left(\lambda \mathbf{b}^T \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} - \hat{\mathbf{x}}_{\text{pred}}^T \mathbf{P}^{-1} \mathbf{A} + \mathbf{b}^T \mathbf{P}^{-1} - \tilde{\mathbf{z}}^T \mathbf{R}^{-1} \mathbf{H} - \lambda \tilde{\mathbf{z}}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{A} \right) \mathbf{x}. \end{aligned} \quad (103)$$

The left side of (103) is a constant; it does not change as $\mathbf{x}$ changes. Thus, for the right-hand side of (103) to be a constant, the linear and the quadratic terms must be zero.

Setting the quadratic term in the right side of (103) to zero, results in

$$\mathbf{A}^T \mathbf{P}^{-1} + \lambda \mathbf{A}^T \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} + \frac{1}{2} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} = \mathbf{0}. \quad (104)$$

Since the quadratic term in (103) should still be valid if some of the inner terms are transposed, transposing two of the terms in (104) results in

$$\mathbf{P}^{-1} \mathbf{A} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{A} + \frac{1}{2} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} = \mathbf{0}. \quad (105)$$

Solving for $\mathbf{A}$,

$$\mathbf{A} = -\frac{1}{2} (\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}. \quad (106)$$

Next, $\mathbf{A}$ is expressed in a manner that will make identifying a relation between $\mathbf{A}$ and $\mathbf{P}$ simpler. A matrix identity for a

given set of matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ is [48]

$$(\mathbf{A} + \mathbf{BCD})^{-1} \mathbf{BC} = \mathbf{A}^{-1} \mathbf{B} (\mathbf{C}^{-1} + \mathbf{DA}^{-1} \mathbf{B})^{-1}. \quad (107)$$

In this problem at hand, $\mathbf{A} = \mathbf{P}^{-1}$, $\mathbf{B} = \mathbf{H}^T$, $\mathbf{C} = \mathbf{R}^{-1}$, and $\mathbf{D} = \lambda \mathbf{H}$. Consequently,

$$\mathbf{A} = -\frac{1}{2} \mathbf{P} \mathbf{H}^T (\lambda \mathbf{H} \mathbf{P} \mathbf{H}^T + \mathbf{R})^{-1} \mathbf{H}. \quad (108)$$

In this form, noting that $\mathbf{P}$ is a covariance matrix and is thus symmetric, it is clear that

$$\mathbf{P} \mathbf{A}^T = \mathbf{A} \mathbf{P}. \quad (109)$$

This fact will be useful when determining $\mathbf{b}$.

Next, set the coefficient of the linear term in (103) to zero:

$$\lambda \mathbf{b}^T \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} - \hat{\mathbf{x}}_{\text{pred}}^T \mathbf{P}^{-1} \mathbf{A} + \mathbf{b}^T \mathbf{P}^{-1} - \tilde{\mathbf{z}}^T \mathbf{R}^{-1} \mathbf{H} - \lambda \tilde{\mathbf{z}}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{A} = 0. \quad (110)$$

Combining the $\mathbf{b}$ terms,

$$\mathbf{b}^T (\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}) = \hat{\mathbf{x}}_{\text{pred}}^T \mathbf{P}^{-1} \mathbf{A} + \tilde{\mathbf{z}}^T \mathbf{R}^{-1} \mathbf{H} + \lambda \tilde{\mathbf{z}}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{A}. \quad (111)$$

Taking the transpose of both sides results in

$$(\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}) \mathbf{b} = \mathbf{A}^T \mathbf{P}^{-1} \hat{\mathbf{x}}_{\text{pred}} + (\mathbf{H}^T \mathbf{R}^{-1} + \lambda \mathbf{A}^T \mathbf{H}^T \mathbf{R}^{-1}) \tilde{\mathbf{z}}. \quad (112)$$

Using the identity from (109) and multiplying both sides by $\mathbf{P}$, leads to

$$(\mathbf{I} + \lambda \mathbf{P} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}) \mathbf{b} = \mathbf{A} \hat{\mathbf{x}}_{\text{pred}} + (\mathbf{P} \mathbf{H}^T \mathbf{R}^{-1} + \lambda \mathbf{A} \mathbf{P} \mathbf{H}^T \mathbf{R}^{-1}) \tilde{\mathbf{z}} \quad (113)$$

and thus

$$\mathbf{b} = (\mathbf{I} + \lambda \mathbf{P} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} (\mathbf{A} \hat{\mathbf{x}}_{\text{pred}} + (\mathbf{P} \mathbf{H}^T \mathbf{R}^{-1} + \lambda \mathbf{A} \mathbf{P} \mathbf{H}^T \mathbf{R}^{-1}) \tilde{\mathbf{z}}). \quad (114)$$

The matrix inversion can be eliminated using the following matrix identity for a given set of matrices, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ [48]

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{B} (\mathbf{C}^{-1} + \mathbf{DA}^{-1} \mathbf{B})^{-1} \mathbf{DA}^{-1}. \quad (115)$$

In this case, $\mathbf{A} = \mathbf{I}$, $\mathbf{B} = \lambda \mathbf{P} \mathbf{H}^T$, $\mathbf{C} = \mathbf{R}^{-1}$, and $\mathbf{D} = \mathbf{H}$. Applying this to the inverse term in (114) and using (108) for the $\mathbf{A}$ matrix in the flow leads to

$$(\mathbf{I} + \lambda \mathbf{P} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} = \mathbf{I} - \lambda \mathbf{P} \mathbf{H}^T (\lambda \mathbf{H} \mathbf{P} \mathbf{H}^T + \mathbf{R})^{-1} \mathbf{H} \quad (116)$$

$$= \mathbf{I} + 2\lambda \mathbf{A}. \quad (117)$$

Substituting (117) into (114) leads to the following expression for $\mathbf{b}$

$$\mathbf{b} = (\mathbf{I} + 2\lambda \mathbf{A}) (\mathbf{A} \hat{\mathbf{x}}_{\text{pred}} + (\mathbf{P} \mathbf{H}^T \mathbf{R}^{-1} + \lambda \mathbf{A} \mathbf{P} \mathbf{H}^T \mathbf{R}^{-1}) \tilde{\mathbf{z}}). \quad (118)$$

Given $\mathbf{A}$ and $\mathbf{b}$, we can look again at (103) and though we have zeroed the right side, it is clear that nothing on the left side can cancel the $\frac{1}{2} \log(\|2\pi \mathbf{R}\|)$ term to zero it. Consequently, the final approximation in the derivation of the exact flow is to neglect those terms.

In practice, the expansion in (98) is typically taken at $\mathbf{x}_0 = \mathbf{x}$, which is the current point under consideration. Thus, $\tilde{\mathbf{z}}$ in (100) is the negative innovation:

$$\tilde{\mathbf{z}} = -\boldsymbol{\nu} = \mathbf{z} - \mathbf{h}(\mathbf{x}). \quad (119)$$

All together, the exact flow is given by (21), (22), (23), and (24). When the exact flow is used with a nonlinear measurement model, whereby $\mathbf{H}$ is taken as a gradient as in (19), the flow has been called the localized exact flow [59].

APPENDIX D THE DIAGONAL NOISE FLOW

This appendix derives the diagonal noise flow in a manner similar to that suggested in [11]. If we let the covariance matrix $\mathbf{Q}$ in (12) be of the form $\mathbf{Q} = \alpha \mathbf{I}$, where $\mathbf{I}$ is the identity matrix, then we can derive a parametric equation which allows for optimization over α . Recall equation (58) and suppose that we can choose a $\hat{\mathbf{f}}$ such that $\nabla_{\mathbf{x}} \text{div}(\hat{\mathbf{f}}) \approx \nabla_{\mathbf{x}} \text{div}(\mathbf{f})$ and $\frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{x}} \approx \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$. Then by substitution and rearranging we get

$$-(\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p) \mathbf{f} \approx \nabla_{\mathbf{x}} \log m + \nabla_{\mathbf{x}} \text{div}(\hat{\mathbf{f}}) + (\nabla_{\mathbf{x}} \hat{\mathbf{f}}^T) (\nabla_{\mathbf{x}} \log p) - \nabla_{\mathbf{x}} \left\{ \frac{1}{2p} \nabla_{\mathbf{x}}^T (p \mathbf{Q}) \nabla_{\mathbf{x}} \right\} \quad (120)$$

$$\mathbf{f} \approx -\left\{ \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p \right\}^{-1} \left\{ \nabla_{\mathbf{x}} \log m + \nabla_{\mathbf{x}} \text{div}(\hat{\mathbf{f}}) + (\nabla_{\mathbf{x}} \hat{\mathbf{f}}^T) (\nabla_{\mathbf{x}} \log p) - \nabla_{\mathbf{x}} \left\{ \frac{1}{2p} \nabla_{\mathbf{x}}^T (p \mathbf{Q}) \nabla_{\mathbf{x}} \right\} \right\}^T. \quad (121)$$

Using the simplification of the $\mathbf{Q}$ term in (121) in (77) (which relies on a constant $\mathbf{Q}$ assumption, and the identity that the trace of a sum of matrices is the sum of the traces of individual matrices:

$$\nabla_{\mathbf{x}} \left\{ \frac{1}{p} \nabla_{\mathbf{x}}^T (p\mathbf{Q}) \nabla_{\mathbf{x}} \right\} = \frac{1}{p} \sum_{i=1}^{d_x} \sum_{j=1}^{d_x} \frac{\partial^2 p}{\partial x_i \partial x_j} [\mathbf{Q}]_{i,j} = \text{Tr} \left\{ \mathbf{Q} (\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p) \right\} + \text{Tr} \left\{ \mathbf{Q} (\nabla_{\mathbf{x}} \log p) (\nabla_{\mathbf{x}} \log p)^T \right\}. \quad (122)$$

Assuming that $\mathbf{Q} = \alpha \mathbf{I}$, where $\alpha \in \mathbb{R}_{\geq 0}$ and $\mathbf{I}$ is the identity matrix, and using the properties that the ordering of the terms of a matrix product in a trace can be rotated and that the trace of a matrix equals the trace of its transpose:

$$\frac{1}{p} \sum_{i=1}^{d_x} \sum_{j=1}^{d_x} \frac{\partial^2 p}{\partial x_i \partial x_j} [\mathbf{Q}]_{i,j} = \alpha \left(\text{Tr} \left\{ \nabla_{\mathbf{x}}^T \nabla_{\mathbf{x}} \log p \right\} + \text{Tr} \left\{ (\nabla_{\mathbf{x}} \log p)^T (\nabla_{\mathbf{x}} \log p) \right\} \right). \quad (123)$$

The first trace term in (123) can be identified as $\text{div}(\nabla_{\mathbf{x}} \log p)$. Thus, by substituting $\mathbf{Q} = \alpha \mathbf{I}$, it can be seen that we have effectively applied the identity that

$$\frac{1}{p} \text{div}(\nabla_{\mathbf{x}} p) = \text{div}(\nabla_{\mathbf{x}} \log p) + (\nabla_{\mathbf{x}} \log p)^T (\nabla_{\mathbf{x}} \log p). \quad (124)$$

Substituting (123) back into (121), and solving for $\mathbf{f}$ leads to

$$\mathbf{f} = - \left\{ \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p \right\}^{-1} \left\{ \nabla_{\mathbf{x}} \log m + \nabla_{\mathbf{x}} \text{div}(\hat{\mathbf{f}}) + (\nabla_{\mathbf{x}} \hat{\mathbf{f}}^T) (\nabla_{\mathbf{x}} \log p) - \beta \right\}^T \quad (125)$$

where

$$\beta = \frac{\alpha}{2} \nabla_{\mathbf{x}} \left\{ \text{div}(\nabla_{\mathbf{x}} \log p) + (\nabla_{\mathbf{x}} \log p)^T (\nabla_{\mathbf{x}} \log p) \right\}. \quad (126)$$

This provides a parametric equation which can be optimized over α in a number of computationally efficient ways. However, to go a littler further, we can assume $\hat{\mathbf{f}}$ is a linear function (so that $\nabla_{\mathbf{x}} \text{div}(\hat{\mathbf{f}}) = \mathbf{0}$) and that $\mathbf{f}$ is a geodesic flow:

$$\mathbf{f} = - \left\{ \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p \right\}^{-1} \left\{ \nabla_{\mathbf{x}} \log m \right\}^T. \quad (127)$$

Plugging (127) into (125), assuming that $\left\{ \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p \right\}^{-1}$ is positive definite, and simplifying yields

$$- \left\{ \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p \right\}^{-1} \left\{ \nabla_{\mathbf{x}} \log m \right\}^T \approx - \left\{ \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log p \right\}^{-1} \left\{ \nabla_{\mathbf{x}} \log m + (\nabla_{\mathbf{x}} \hat{\mathbf{f}}^T) (\nabla_{\mathbf{x}} \log p) - \beta \right\}^T \quad (128)$$

$$\alpha \nabla_{\mathbf{x}} \left\{ \text{div}(\nabla_{\mathbf{x}} \log p) + (\nabla_{\mathbf{x}} \log p)^T (\nabla_{\mathbf{x}} \log p) \right\} = 2(\nabla_{\mathbf{x}} \hat{\mathbf{f}}^T) (\nabla_{\mathbf{x}} \log p) \quad (129)$$

The expression in (129) is actually an overdetermined set of equations for α . One could obtain an approximate solution using a pseudoinverse or by taking the trace of both side of the equation and dividing. However, the approach chosen in [11] is to take the l_2 norm of both sides and divide, which leads to the expression of (27) for α .

REFERENCES

- [1] V. J. Aidala and S. E. Hammel, "Utilization of modified polar coordinates for bearings-only tracking," *IEEE Transactions on Automatic Control*, vol. 28, no. 3, pp. 283–294, Mar. 1983.
- [2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: John Wiley and Sons, Inc, 2001.
- [4] K. L. Bell and L. D. Stone, "Implementation of the homotopy particle filter in the JPDA and MAP-PF multi-target tracking algorithms," in *Proceedings of the 17th International Conference on Information Fusion*, Salamanca, Spain, 7–10 Jul. 2014.
- [5] L. Chen and R. K. Mehra, "A study of "nonlinear filters with particle flow induced by log-homotopy"," in *Proceedings of SPIE: Signal Processing, Sensor Fusion, and Target Recognition XIX*, vol. 7697, Orlando, FL, 27 Apr. 2010.
- [6] S. Choi, P. Willett, F. Daum, and J. Huang, "Discussion and application of the homotopy filter," in *Proceedings of SPIE: Signal Processing, Sensor Fusion, and Target Recognition XX*, vol. 8050, Orlando, FL, 25 Apr. 2011.
- [7] D. F. Crouse, "Basic tracking using nonlinear 3d monostatic and bistatic measurements," *IEEE Aerospace and Electronic Systems Magazine*, vol. 29, no. 8, Part II, pp. 4–53, Aug. 2014.
- [8] —, "Basic tracking using nonlinear continuous-time dynamic models," *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 2, Part II, pp. 4–41, Feb. 2015.
- [9] —, "The tracker component library: Free routines for rapid prototyping," *IEEE Aerospace and Electronic Systems Magazine*, vol. 32, no. 5, pp. 18–27, May 2017.
- [10] —, "Particle flow filters: Biases and bias avoidance," in *Proceedings of the 22nd International Conference on Information Fusion*, Ottawa, Canada, 2–5 Jul. 2019.
- [11] F. Daum, "Seven dubious methods to compute optimal q for Bayesian stochastic particle flow," in *Proceedings of the 19th International Conference on Information Fusion*, Heidelberg, Germany, 5–8 Jul. 2016.
- [12] F. Daum and J. Huang, "Nonlinear filters with log-homotopy," in *Proceedings of SPIE: Signal and Data Processing of Small Targets*, vol. 6699, San Diego, CA, 26 Aug. 2007.
- [13] —, "Particle flow for nonlinear filters with log-homotopy," in *Proceedings of SPIE: Signal and Data Processing of Small Targets*, vol. 6969, Orlando, FL, 16 Apr. 2008.
- [14] —, "Exact particle flow for nonlinear filters," in *Proceedings of SPIE: Signal Processing, Sensor Fusion, and Target Recognition XIX*, vol. 7697, Orlando, FL, 27 Apr. 2010.

- [15] —, “Exact particle flow for nonlinear filters: Seventeen dubious solutions to a first order linear underdetermined PDE,” in *Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, 7-10 Nov. 2010, pp. 64–71.
- [16] —, “Generalized particle flow for nonlinear filters,” in *Proceedings of SPIE: Signal and Data Processing of Small Targets*, vol. 7698, Orlando, FL, 15 Apr. 2010.
- [17] —, “Bayesian big bang,” in *Proceedings of SPIE: Signal and Data Processing of Small Targets*, vol. 8137, San Diego, CA, 16 Sep. 2011.
- [18] —, “Hollywood log-homotopy: Movies of particle flow for nonlinear filters,” in *Proceedings of SPIE: Signal Processing, Sensor Fusion, and Target Recognition XX*, vol. 8050, Orlando, FL, 2 May 2011.
- [19] —, “Friendly rebuttal to Chen and Mehra: Incompressible particle flow for nonlinear filters,” in *Proceedings of SPIE: Signal Processing, Sensor Fusion, and Target Recognition XXI*, vol. 8392, 17 May 2012.
- [20] —, “Particle flow and Monge-Kantorovich transport,” in *Proceedings of the 15th International Conference on Information Fusion*, Singapore, 9–12 Jul. 2012, pp. 135–142.
- [21] —, “Small curvature particle flow for nonlinear filters,” in *Proceedings of SPIE: Signal and Data Processing of Small Targets*, vol. 8393, Baltimore, MD, 15 May 2012.
- [22] —, “Fourier transform particle flow for nonlinear filters,” in *Proceedings of SPIE: Signal Processing, Sensor Fusion, and Target Recognition XXII*, vol. 8745, Baltimore, MD, 23 May 2013.
- [23] —, “Particle flow for nonlinear filters, Bayesian decisions and transport,” in *Proceedings of the 16th International Conference on Information Fusion*, Istanbul, Turkey, 9–12 Jul. 2013.
- [24] —, “Particle flow inspired by Knothe-Rosenblatt transport for nonlinear filter,” in *Proceedings of SPIE: Signal Processing, Sensor Fusion, and Target Recognition XXII*, vol. 8745, Baltimore, MD, 23 May 2013.
- [25] —, “Particle flow with non-zero diffusion for nonlinear filters,” in *Proceedings of SPIE: Signal Processing, Sensor Fusion, and Target Recognition XXII*, vol. 8745, Baltimore, MD, 23 May 2013.
- [26] —, “Particle flow with non-zero diffusion for nonlinear filters, Bayesian decisions and transport,” in *Proceedings of SPIE: Signal and Data Processing of Small Targets*, vol. 8857, San Diego, CA, 30 Sep. 2013.
- [27] —, “Zero curvature particle flow for nonlinear filters,” in *Proceedings of SPIE: Signal Processing, Sensor Fusion, and target Recognition XXII*, vol. 8745, Baltimore, MD, 23 May 2013.
- [28] —, “How to avoid normalization of particle flow for nonlinear filters,” in *Proceedings of SPIE: Signal and Data Processing of Small Targets*, vol. 9092, Baltimore, MD, 13 Jun. 2014.
- [29] —, “How to avoid normalization of particle flow for nonlinear filters, Bayesian decisions, and transport,” in *Proceedings of SPIE: Signal and Data Processing of Small Targets*, vol. 9092, Baltimore, MD, 13 Jun. 2014.
- [30] —, “Renormalization group flow and other ideas inspired by physics for nonlinear filters, Bayesian decisions, and transport,” in *Proceedings of SPIE: Signal Processing, Sensor/Information Fusion and Target Recognition XXIII*, vol. 9091, Baltimore, MD, 20 Jun. 2014.
- [31] —, “Seven dubious methods to mitigate stiffness in particle flow with non-zero diffusion for nonlinear filters, Bayesian decisions, and transport,” in *Proceedings of SPIE: Signal and Data Processing of Small Targets*, vol. 9092, Baltimore, MD, 13 Jun. 2014.
- [32] —, “A baker’s dozen of new particle flows for nonlinear filters Bayesian decisions and transport,” in *Proceedings of SPIE: Signal Processing, Sensor Fusion, and Target Recognition XXIV*, vol. 9474, Baltimore, MD, 21 Apr. 2015.
- [33] —, “Proof that particle flow corresponds to Bayes’ rule: Necessary and sufficient conditions,” in *Proceedings of SPIE: Signal Processing, Sensor/Information Fusion and Target Recognition XXIV*, vol. 9474, Baltimore, MD, 20 Apr. 2015.
- [34] —, “Renormalization group flow in K-space for nonlinear filters, Bayesian decisions and transport,” in *Proceedings of the 18th International Conference on Information Fusion*, Washington, DC, 6–9 Jul. 2015, pp. 1617–1624.
- [35] —, “A plethora of open problems in particle flow research for nonlinear filters,” in *Proceedings of SPIE: Signal Processing, Sensor Fusion, and Target Recognition XXV*, vol. 9842, Baltimore, MD, 17 May 2016.
- [36] F. Daum, J. Huang, and A. Noushin, “Gromov’s method for Bayesian stochastic particle flow: A simple exact formula for Q_t ,” in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Baden-Baden, Germany, 19–21 Sep. 2016, pp. 540–545.
- [37] —, “Coulomb’s law particle flow for nonlinear filters,” in *Proceedings of SPIE: Signal and Data Processing of Small Targets*, vol. 8137, San Diego, CA, 16 Sep. 2011.
- [38] —, “A friendly rebuttal to Mallick and Sindhu on particle flow for Bayes’ rule,” in *Proceedings of SPIE: Signal Processing, Sensor/Information Fusion, and Target Recognition*, Baltimore, MD, 17 May 2016.
- [39] —, “Generalized Gromov method for stochastic particle flow filters,” in *Proceedings of SPIE: Signal Processing, Sensor/Information Fusion and Target Recognition XXVI*, vol. 10200, Anaheim, CA, 2 May 2017.
- [40] —, “New theory and numerical results for Gromov’s method for stochastic particle flow filters,” in *Proceedings of the 21st International Conference on Information Fusion*, Cambridge, UK, 10–13 Jul. 2018, pp. 108–115.
- [41] F. Daum, A. Noushin, and J. Huang, “Numerical experiments for Gromov’s stochastic particle flow filters,” in *Proceedings of SPIE: Signal Processing, Sensor/Information Fusion and Target Recognition XXVI*, vol. 10200, Anaheim, CA, 2 May 2017.
- [42] T. Ding and M. J. Coates, “Implementation of the Daum-Huang exact-flow particle filter,” in *Proceedings of IEEE Statistical Signal Processing Workshop*, Ann Arbor, MI, 5–8 Aug. 2012, pp. 257–260.
- [43] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York: Springer, 2010.
- [44] R. Fuentes and E. Blake, “Unpublished notes,” Jul. 2016.
- [45] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. Baltimore: The Johns Hopkins University Press, 2013.
- [46] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proceedings-F*, vol. 140, no. 2, pp. 109–113, Apr. 1993.
- [47] D. Gupta, Syamantak, J. Y. Yu, M. Mallick, M. Coates, and M. Moreland, “Comparison of angle-only filtering algorithms in 3D using EKF, UKF, PF, PFF, and ensemble KF,” in *Proceedings of the 18th International Conference on Information Fusion*, Washington, DC, 6–9 Jul. 2015, pp. 1649–1656.
- [48] H. V. Henderson and S. R. Searle, “On deriving the inverse of a sum of matrices,” *SIAM Review*, vol. 23, no. 1, pp. 53–60, Jan. 1981.
- [49] D. J. Higham, X. Mao, and A. M. Stuart, “Strong convergence of Euler-type methods for nonlinear stochastic differential equations,” *SIAM Journal on Numerical Analysis*, vol. 40, no. 3, pp. 1041–1063, 2002.
- [50] H. D. Hoelzer, G. W. Johnson, A. O. Cohen, and K. R. Brown, “Modified polar coordinates – the key to well behaved bearings only ranging,” IBM Federal Systems Division, Manassas, VA, Tech. Rep. IR&D Report 78-M19-0001A, 31 Aug. 1978.
- [51] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York: Academic Press, 1970.
- [52] V. P. Jilkov, J. Wu, and H. Chen, “Performance comparison of GPU-accelerated particle flow and particle filters,” in *Proceedings of the 16th International Conference on Information Fusion*, Istanbul, Turkey, 9–12 Jul. 2013.
- [53] M. A. A. Khan, “Nonlinear filtering based on log-homotopy particle flow,” Ph.D. dissertation, Rheinischen Friedrich-Wilhelms-Universität Bonn, Bonn, Germany, Aug. 2018.
- [54] M. A. Khan, A. De Freitas, L. Mihaylova, M. Ulmke, and W. Koch, “Bayesian processing of big data using log homotopy based particle flow filters,” in *Proceedings of Sensor Data Fusion: Trends, Solutions, Applications*, Bonn, Germany, 10–12 Oct. 2017.
- [55] M. A. Khan and M. Ulmke, “Improvements in the implementation of log-homotopy based particle flow filters,” in *Proceedings of the 18th International Conference on Information Fusion*, Washington, DC, 6–9 Jul. 2015, pp. 74–81.

- [56] M. A. Khan, M. Ulmke, and W. Koch, "Analysis of log-homotopy based particle flow filters," *Journal of Advances in Information Fusion*, vol. 12, no. 1, pp. 73–94, Jun. 2017.
- [57] P. E. Kloeden and P. Eckhard, *Numerical Solution of Stochastic Differential Equations*. Berlin: Springer, 1999.
- [58] C. Kreucher and K. Bell, "A geodesic flow particle filter for non-thresholded radar tracking," *IEEE transactions on Aerospace and Electronic Systems*, vol. 54, no. 6, pp. 3169–3175, Dec. 2018.
- [59] Y. Li and M. Coates, "Particle filtering with invertible particle flow," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 4102–4116, 1 Aug. 2017.
- [60] M. Mallick and B. Sindhu, "Critical analysis of the particle flow filter," in *Proceedings of the International Conference on Control, Automation and Information Sciences*, Changshu, China, 29–31 Oct. 2015, pp. 512–517.
- [61] S. Mori, F. Daum, and J. Douglas, "Adaptive step size approach to homotopy-based particle filtering Bayesian update," in *Proceedings of the 19th International Conference on Information Fusion*, Heidelberg, Germany, 5–8 Jul. 2016.
- [62] N. Moshtag, J. D. Chan, and M. W. Chan, "Homotopy particle filter for ground-based tracking of satellites at GEO," in *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, Wailea, HI, 20–23 Sep. 2016.
- [63] N. Moshtag and M. W. Chan, "Multisensor fusion using homotopy particle filter," in *Proceedings of the 18th International Conference on Information Fusion*, Washington, DC, 6–9 Jul. 2015, pp. 1641–1648.
- [64] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston: Artech House, 2004.
- [65] X. Wang and W. Ni, "An unbiased homotopy particle filter and its application to the INS/GPS integrated navigation system," in *Proceedings of the 20th International Conference on Information Fusion*, Xi'an, China, 10–13 Jul. 2017.
- [66] Z. Yin and S. Gan, "An error-corrected Euler-Maruyama method for stiff stochastic differential equations," *Applied Mathematics and Computation*, vol. 25, pp. 630–641, 1 Apr. 2015.
- [67] The tracker component library. [Online]. Available: <https://github.com/USNavalResearchLaboratory/TrackerComponentLibrary>