Computational Statistics and Data Analysis 55 (2011) 280–290

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Nonlinear tracking in a diffusion process with a Bayesian filter and the finite element method

M.W. Pedersen [a,*], U.H. Thygesen [b], H. Madsen [a]

[a] Department for Informatics and Mathematical Modelling, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

[b] National Institute of Aquatic Resources, Technical University of Denmark, 2920 Charlottenlund, Denmark

ARTICLE INFO

Article history:
Received 17 February 2009
Received in revised form 18 March 2010
Accepted 25 April 2010
Available online 2 May 2010

Keywords:
Finite element method
Hidden Markov model
Nonlinear state estimation
Point-mass filter
Sequential Monte Carlo
Stochastic differential equation

ABSTRACT

A new approach to nonlinear state estimation and object tracking from indirect observations of a continuous time process is examined. Stochastic differential equations (SDEs) are employed to model the dynamics of the unobservable state. Tracking problems in the plane subject to boundaries on the state-space do not in general provide analytical solutions. A widely used numerical approach is the sequential Monte Carlo (SMC) method which relies on stochastic simulations to approximate state densities. For off-line analysis, however, accurate smoothed state density and parameter estimation can become complicated using SMC because Monte Carlo randomness is introduced. The finite element (FE) method solves the Kolmogorov equations of the SDE numerically on a triangular unstructured mesh for which boundary conditions to the state-space are simple to incorporate. The FE approach to nonlinear state estimation is suited for off-line data analysis because the computed smoothed state densities, maximum *a posteriori* parameter estimates and state sequence are deterministic conditional on the finite element mesh and the observations. The proposed method is conceptually similar to existing point-mass filtering methods, but is computationally more advanced and generally applicable. The performance of the FE estimators in relation to SMC and to the resolution of the spatial discretization is examined empirically through simulation. A real-data case study involving fish tracking is also analysed.

1. Introduction

Since the introduction of the Kalman filter (Kalman, 1960), state-space models (SSMs) have been widely used to solve object tracking problems and have undergone constant development. Initially, tracking problems were mostly related to radar observations (Pearson and Stear, 1974), but more recent applications now range from animal tracking, based on e.g. observations of daylight (Sibert et al., 2003), to object tracking in surveillance camera recordings (Comaniciu et al., 2003). The Kalman filter and its variants are suitable for linear and mildly nonlinear tracking problems owing to their computational simplicity. However, the majority of the Kalman filter variants build on a parametric representation of the probability density, thus limiting their use when problems become complex and non-Gaussian. Hence, for highly nonlinear problems, nonparametric Bayesian filtering techniques must be applied.

A common approach to filtering nonlinear SSMs is the use of simulation methods such as Markov chain Monte Carlo (Golightly and Wilkinson, 2008) and sequential Monte Carlo (SMC) (Creal, 2008; Ristic et al., 2004). The SMC methodology is intuitive and applies to virtually all SSMs regardless of order or type. One drawback of the method is that randomness is

* Corresponding author. Tel.: +45 4525 3095; fax: +45 4588 2673.
E-mail addresses: mwp@imm.dtu.dk, wpsgodd@gmail.com (M.W. Pedersen), uht@aqua.dtu.dk (U.H. Thygesen), hm@imm.dtu.dk (H. Madsen).

0167-9473/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2010.04.018

introduced into the filter, which can complicate matters of smoothing and parameter estimation for certain problems. For example, it can be difficult to obtain maximum likelihood estimates of model parameters because the randomness makes the likelihood function non-differentiable. Advanced algorithms have been devised to ameliorate this issue, although more work is still required in this respect (Doucet and Tadić, 2003). SMC methods are particularly appealing for high-dimensional problems because they suffer minimally from the curse of dimensionality when evaluating the integrals of Bayesian filters (Cappé et al., 2005).

Another approach to nonlinear state estimation is to deterministically discretize the continuous state-space and then solve the governing partial differential equations and related integrals numerically on this discretization (Bucy and Senne, 1971; Kitagawa, 1987). The method is feasible for SSMs with a low-dimensional state-space, up to dimension 3, say. This paper focuses on the case of a planar (two-dimensional) state-space, but the method is applicable to scalar and theoretically also to higher-dimensional situations although the required computational effort in such cases becomes substantial. Planar problems are common in many applications, for example when tracking individuals on the surface of the Earth, and therefore deserve special attention.

The discretized SSM can be solved by the framework of discrete hidden Markov models (HMMs) (Cappé et al., 2005). Historically, HMMs have been applied in cases where the state-space is discrete by nature, e.g. for digital signals, and therefore discretized continuous SSMs are rarely labeled as HMMs even though HMM theory applies. Other common names for methods that operate on the discretized problem are "point-mass" filters or direct Fokker–Planck-based methods (FPMs).

Previous studies that have considered discretized SSMs have opted for the simplest and most obvious discretization approach which is a uniform gridding of the state-space (Šimandl et al., 2006; Thygesen et al., 2009). As a result, the solution to the forward Kolmogorov equation or, synonymously, the Fokker–Planck equation can be obtained by finite differencing. The finite difference (FD) solution scheme is easy to implement yet gives powerful results. It comes, however, with some critical drawbacks such as lack of a standard procedure for imposing boundary conditions and non-conformity of the discretization to curvatures in the boundary geometry beyond one dimension. As a consequence of the rectangular uniform grid, the computational resources are distributed evenly over the state-space, which is rarely appropriate as some states will have close to zero probability. To this end there have been attempts to focus computer power to the relevant parts of the state-space by adaptively changing the discrete grid during the filtering process (Šimandl et al., 2006). This improvement reduces the computational burden, but severely complicates matters of smoothing and state sequence estimation and consequently limits the application of the method to filtering problems.

In this study we introduce an alternative non-uniform discretization of a continuous SSM and a method suited for obtaining smoothed state estimates and maximum *a posteriori* estimates of model parameters on this grid. In the two-dimensional case the discretization, that remains unchanged during computations, consists of a number of differently sized triangular shaped elements that are joined at the vertices to form a mesh. The lack of uniformity gives the mesh the ability to conform to curved boundary geometries that arise in many applications of geographical tracking, which is our primary focus. Also, the unstructured mesh allows for finer discretization of important parts of the state-space, thereby utilizing computer resources more efficiently. Within this framework the solutions to the Kolmogorov equations are provided by the finite element (FE) method, reviewed in Clough (1980). The performance in relation to SMC is illustrated by a simple Brownian bridge example and numerical properties are assessed through synthetic and real data sets.

The paper is composed such that Section 2 establishes the state-space model. Section 3 presents the details of the Bayesian filter while Section 4 details the discretization of the state-space. Section 5 examines parameter and state sequence estimation and Section 6 illustrates the application of the method through examples. Section 7 provides concluding remarks.

2. The stochastic model

The dynamic system that we are considering is described by a two-dimensional stochastic differential equation

$$\mathrm{d}\boldsymbol{X}_t = f(t, \boldsymbol{X}_t)\mathrm{d}t + g(t, \boldsymbol{X}_t)\mathrm{d}\boldsymbol{B}_t, \tag{1}$$

where $\boldsymbol{X}_t \in \mathbf{R}^2$ is the state at time t, $f : \mathbf{R}^{1+2} \to \mathbf{R}^2$, $\boldsymbol{B}_t$ is two-dimensional Brownian motion and $g : \mathbf{R}^{1+2} \to \mathbf{R}^{2\times 2}$. We assume $\boldsymbol{X}_t$ to be reflected at boundaries according to (10) in the Appendix. We have observations indexed by $k \in \{1, \ldots, N\}$ with the observation pertaining to the interval $[t_{k-1}, t_k]$ denoted by $\boldsymbol{z}_k \in \mathbf{R}^p$. We will refer to a set of observations as $\mathcal{Z}_k = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k)^T$. There are no restrictions on the type and nature of $\boldsymbol{z}_k$. The observation equation relates the noisy time-discrete observations to the continuous process

$$\boldsymbol{z}_k = h(t_k, \boldsymbol{X}_k, \boldsymbol{w}_k), \tag{2}$$

where $\boldsymbol{w}_k \in \mathbf{R}^q$ is a random perturbation and $\boldsymbol{X}_k$ is short for $\boldsymbol{X}_{t_k}$. No assumptions are made about the form of the possibly nonlinear mapping $h : \mathbf{R}^{1+2+q} \to \mathbf{R}^p$. The two-dimensional state-space can be written explicitly as $\boldsymbol{X}_k = (X_k, Y_k)^T$. The observation errors, $\boldsymbol{w}_k$, are assumed to be independent and identically distributed and can have any type of distribution.

3. The Bayesian filter and smoother

In this section we describe the filtering and smoothing recursions required for predicting, updating and smoothing the state estimates given data and the stochastic model. We assume here that the parameters of the model are known; in Section 5 we address estimation of these parameters.

3.1. Filtering

We apply a Bayesian filter to estimate the probability density of the state. The density conditional on $\mathcal{Z}_k$ is given by $\phi(t, \boldsymbol{x}_t|\mathcal{Z}_k)\mathrm{d}\boldsymbol{x}_t = P(\boldsymbol{X}_t \in \mathrm{d}\boldsymbol{x}_t|t, \mathcal{Z}_k)$ for $t \geq t_k$. The filter consists of two steps, the time update and the data update, that are performed recursively. The time update is related to the system Eq. (1), and has the purpose of predicting the evolution of $\phi(t, \boldsymbol{x}_t|\mathcal{Z}_k)$ throughout the elapsed time between observations. The time evolution of $\phi(t, \boldsymbol{x}_t|\mathcal{Z}_k)$ is described by the Kolmogorov forward equation (or Fokker–Planck equation) (Øksendal, 2007) in two dimensions

$$\dot{\phi} = -\nabla \cdot (u\phi - D\nabla\phi), \tag{3}$$

with $\phi = \phi(t, \boldsymbol{x}_t|\mathcal{Z}_k)$, $\dot{\phi}$ denoting $\frac{\partial\phi}{\partial t}$, $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$, and where $D = \frac{1}{2}gg^T$ and $u = f - \nabla D$ are the diffusion and advection parameters respectively. No-flux boundary conditions are imposed to ensure conservation of probability mass within the domain. The initial condition for (3) is $\phi(t_k, \boldsymbol{x}_k|\mathcal{Z}_k)$ and the solution, $\phi(t_{k+1}, \boldsymbol{x}_{k+1}|\mathcal{Z}_k)$, is obtained by solving the equation over the time period $[t_k, t_{k+1}]$.

The data update step consists of applying the information in $\boldsymbol{z}_k$ to $\phi(t_k, \boldsymbol{x}_k|\mathcal{Z}_{k-1})$ using Bayes' rule

$$\phi(t_k, \boldsymbol{x}_k|\mathcal{Z}_k) = \psi_k^{-1}\phi(t_k, \boldsymbol{x}_k|\mathcal{Z}_{k-1})L(\boldsymbol{z}_k|\boldsymbol{x}_k), \tag{4}$$

where $\psi_k = \int \phi(t_k, \boldsymbol{x}_k|\mathcal{Z}_{k-1})L(\boldsymbol{z}_k|\boldsymbol{x}_k)\mathrm{d}\boldsymbol{x}_k$, i.e. a normalization constant. The term $\phi(t_k, \boldsymbol{x}_k|\mathcal{Z}_{k-1})$ comes from solving (3) and the term $L(\boldsymbol{z}_k|\boldsymbol{x}_k)$ is the likelihood of $\boldsymbol{z}_k$ given $\boldsymbol{x}_k$. We will refer to $L(\boldsymbol{z}_k|\boldsymbol{x}_k)$ as the "data likelihood" to avoid confusion with the likelihood of model parameters introduced later.

By recursively solving Eqs. (3) and (4) the density $\phi(t_k, \boldsymbol{x}_k|\mathcal{Z}_k)$ is computed for all $k \in \{1, \ldots, N\}$. The filter is initialized by computing $\phi(t_1, \boldsymbol{x}_1|\boldsymbol{z}_1)$ from (4) with a prior distribution that reflects information available about the initial state.

3.2. The smoother

The smoothing recursions run in reverse time using the results of the filter and give the smoothed estimates, $\phi(t_k, \boldsymbol{x}_k|\mathcal{Z}_N)$, by assuming full knowledge of $\mathcal{Z}_N$ for all $k \in \{1, \ldots, N\}$. At t_k the unused observations in $\phi(t_k, \boldsymbol{x}_k|\mathcal{Z}_k)$ are $\mathcal{W}_{k+1} = (\boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_N)^T$. The information contained in these observations is given by

$$\Lambda(t_{k+1}, \mathcal{W}_{k+1}|\boldsymbol{x}_{k+1}) = \frac{\phi(t_{k+1}, \boldsymbol{x}_{k+1}|\mathcal{Z}_N)}{\phi(t_{k+1}, \boldsymbol{x}_{k+1}|\mathcal{Z}_k)}. \tag{5}$$

Smoothing involves the use of the Kolmogorov backward equation (Øksendal, 2007) which describes the reverse time evolution of $\Lambda(t, \mathcal{W}_{k+1}|\boldsymbol{x}_{k+1})$ according to

$$-\dot{\Lambda} = u\nabla\Lambda + \nabla \cdot (D\nabla\Lambda), \tag{6}$$

where $\Lambda = \Lambda(t, \mathcal{W}_{k+1}|\boldsymbol{x}_{k+1})$ and with the Neumann boundary condition $n \cdot \nabla\Lambda = 0$, where n is a vector normal to the boundary of the state-space. The result of solving (6) with $\Lambda(t_{k+1}, \mathcal{W}_{k+1}|\boldsymbol{x}_{k+1})$ as the terminal condition over the time period $[t_k, t_{k+1}]$ we denote $\Lambda(t_k, \mathcal{W}_{k+1}|\boldsymbol{x}_k)$, which can be interpreted as the data likelihood of the observations $\mathcal{W}_{k+1}$ at t_k. This is perhaps clearer when it is realized that

$$\Lambda(t_k, \mathcal{W}_k|\boldsymbol{x}_k) = \Lambda(t_k, \mathcal{W}_{k+1}|\boldsymbol{x}_k)L(\boldsymbol{z}_k|\boldsymbol{x}_k) \tag{7}$$

and therefore that $\Lambda(t_k, \mathcal{W}_{k+1}|\boldsymbol{x}_k)$ for any k can be calculated from a recursion of (6) and (7). Finally the smoothed estimate satisfies

$$\phi(t_k, \boldsymbol{x}_k|\mathcal{Z}_N) = \phi(t_k, \boldsymbol{x}_k|\mathcal{Z}_k)\Lambda(t_k, \mathcal{W}_{k+1}|\boldsymbol{x}_k).$$

The smoothing recursions are initialized with the final filter estimate $\phi(t_N, \boldsymbol{x}_N|\mathcal{Z}_N)$ which is also a smoothed estimate.

4. Discretization of the state-space

The primary computational burden of the smoothing problem is the need to solve (3) and (6). The partial differential equations can be solved numerically with the finite element (FE) method. The FE method uses an unstructured discretization of the state-space, referred to as the mesh, defined by $M < \infty$ nodes (vertices) with states (locations) $\boldsymbol{x}^{(j)}$ for $j \in \{1, \ldots, M\}$. Associated with the nodes are basis functions $v_j(\boldsymbol{x})$ that act as interpolants via

$$\phi(t, \boldsymbol{x}) \approx \sum_{j=1}^{M} \beta_j(t)v_j(\boldsymbol{x}), \tag{8}$$

where $\beta_j(t) = \phi(t, \boldsymbol{x}^{(j)})$. The shape and form of the basis function depend on the mesh element type as explained by Cook et al. (2001). We use a three-node triangular element which is the simplest two-dimensional element. The technical details of the FE algorithm for this application are reviewed in the Appendix. For a more in depth coverage see e.g. Cook et al. (2001).

The mesh can be created and modified by importing the geometry of the state-space, e.g. a landscape, into mesh generating software. We used the open-source meshing tool Triangle (Shewchuk, 1996). The initial mesh should be coarse with uniformly sized elements to achieve a quick but rough estimate of the posterior distribution. The mesh can then be refined in important regions and coarsened in unimportant regions according to the posterior distribution. In Section 6.2 we examine empirically how the mesh fineness influences the accuracy of results.

Solving (3) requires $T = (t_{k+1} - t_k)/\delta$ FE iterations, where δ is the FE time step. The computational complexity of (3) is therefore $O(TM^2)$ since the solution is obtained by a simple matrix vector multiplication. However, matrix sparsity algorithms can be employed that significantly reduce computation time and memory requirements. For a band matrix with bandwidth K the complexity becomes $O(TKM)$ where $TK \ll M$. For SMC methods (e.g. the particle filter of Cappé et al., 2007) the complexity is $O(M_*)$ for the time update, where M_* is the number of particles. The FE smoothing recursion has complexity $O(TKM)$ as for the filtering recursion. Marginal smoothing for SMC methods requires $O(M_*^2)$ per recursion (Doucet et al., 2000); hence for smoothing problems, when accurate density estimation is required, i.e. when M and M_* become large, SMC methods suffer from a significant increase in computation time compared to the FE approach.

5. Parameter and state estimation

In Section 3 the parameters of the model were assumed to be known. In practice this assumption rarely holds and parameters must therefore be estimated. In this section we describe how maximum likelihood (ML) or maximum *a posteriori* (MAP) parameter estimates can be obtained and we discuss estimation of the optimal state sequence given the posterior distribution.

5.1. Parameter estimation

We focus on off-line parameter estimation and let $\boldsymbol{\theta}$ denote the parameters that we wish to estimate. In filter-based likelihood inference the recursions of Section 3.1 executed with a given set of parameters provide the likelihood function evaluated at those parameters (Ljung, 1997). Specifically, the log-likelihood function of $\boldsymbol{\theta}$ is given by the joint density of the observations

$$l(\boldsymbol{\theta}|\mathcal{Z}_N) = \log\left\{\phi(z_1)\prod_{k=2}^{N}\phi(z_k|\mathcal{Z}_{k-1})\right\}.$$

By storing $\psi_k = \phi(z_k|\mathcal{Z}_{k-1})$ given by (4) for all $k \in \{1, \ldots, N\}$ we obtain a log-likelihood value of $\boldsymbol{\theta}$. Assuming that we have *a priori* information about $\boldsymbol{\theta}$ quantified by the prior density $\pi(\boldsymbol{\theta})$ the MAP estimate of $\boldsymbol{\theta}$ is obtained by maximizing the posterior density, i.e.

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}}\{l(\boldsymbol{\theta}|\mathcal{Z}_N)\pi(\boldsymbol{\theta})\}. \tag{9}$$

Note that (9) gives the ML estimate of $\boldsymbol{\theta}$ if $\pi(\boldsymbol{\theta})$ is uninformative. The log-likelihood function is a deterministic function of $\boldsymbol{\theta}$ conditional on $\mathcal{Z}_N$ and the mesh. In this study we use the fmincon function in the Matlab optimization toolbox to find $\widehat{\boldsymbol{\theta}}$. The optimizer evaluates the gradients numerically which can be computationally cumbersome, so optimization using analytically derived gradients is an important subject of future research.

5.2. State sequence estimation

For tracking problems we are typically more interested in a state sequence or synonymously a track, supplemented by the probability density of the state, rather than the density alone. Consider the restricted FE state-space where we have the discrete states $\boldsymbol{\xi}_k$ for $\boldsymbol{\xi}_k \in \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M)}\}$. We define a track on the mesh nodes $\boldsymbol{\Xi} = (\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N)$ and express the posterior mean $\bar{\boldsymbol{\Xi}} = (\bar{\boldsymbol{\xi}}_1, \ldots, \bar{\boldsymbol{\xi}}_N)$ by its marginals

$$\bar{\boldsymbol{\xi}}_k = \sum_{\boldsymbol{\xi}} \boldsymbol{\xi}\phi(t_k, \boldsymbol{\xi}|\mathcal{Z}_N).$$

The posterior mean is a robust and L^2 optimal estimator of $\boldsymbol{\Xi}$ but cannot be guaranteed to lie within the state-space, e.g. when the state-space is not convex. An alternative is to calculate the posterior mode

$$\widehat{\boldsymbol{\Xi}} = (\widehat{\boldsymbol{\xi}}_1, \ldots, \widehat{\boldsymbol{\xi}}_N) = \arg\max_{\boldsymbol{\Xi}} L(\boldsymbol{\Xi}),$$

where the track likelihood is given by

$$L(\boldsymbol{\Xi}) = L(\boldsymbol{z}_1|\boldsymbol{\xi}_1)\prod_{k=2}^{N} P(\boldsymbol{X}_k = \boldsymbol{\xi}_k|\boldsymbol{X}_{k-1} = \boldsymbol{\xi}_{k-1})L(\boldsymbol{z}_k|\boldsymbol{\xi}_k).$$

The Viterbi algorithm (Viterbi, 2006) is an efficient way of computing the posterior mode. For skewed posterior distributions the mode does not necessarily capture the overall trends in the posterior distribution because it is based on

Table 1
Metrics describing the quality of the estimates of the Brownian bridge for the FE approach and SMC method respectively. The SMC results are calculated as the average of 50 runs with the standard deviation in parenthesis. Computing times T_{filt} and T_{smoo} have the unit of seconds.

$t = 0.5$	$e_{\mu}^{(x)} \times 10^{-4}$	$e_{\mu}^{(y)} \times 10^{-4}$	$e_{\sigma^2}^{(x)} \times 10^{-4}$	$e_{\sigma^2}^{(y)} \times 10^{-4}$	$D(\Phi \parallel \phi)$	$e_{L^1} \times 10^{3}$	T_{filt} (s)	T_{smoo} (s)
FE	3.0	−1.47	−1.45	0.40	103	4.2	3.3	2.3
SMC	−13.2 (24.1)	1.72 (19.3)	0.70 (1.81)	−0.30 (1.50)	3413 (792)	28.4 (4.3)	0.1 (0.0)	5.3 (0.9)

a single outcome. However, in contrast to the posterior mean case it always holds that $\widehat{\boldsymbol{\xi}}_k \in \{\boldsymbol{x}_k(1), \ldots, \boldsymbol{x}_k(M)\}$ for all k. The suitable choice of track depends on the specific application. For the tracking applications considered here we use the posterior mode to avoid invalid state estimates. We note that is also possible to draw random tracks from the posterior distribution, using a recursive algorithm similar to that of Thygesen et al. (2009).

6. Examples

In this section we illustrate the properties of the FE approach in comparison to an SMC method (Section 6.1), with respect to numerics (Section 6.2) and a real-data application (Section 6.3).

6.1. Comparison of FE with SMC

We set up a simple smoothing problem for tracking a particle in two state dimensions $\boldsymbol{X} = (X, Y)^T$. We assume the initial state at $t_1 = 0$ is known, $\boldsymbol{X}_1 = (0.6, 0.5)^T$, and that we have observed the state $\boldsymbol{z}_2 = (0.4, 0.5)^T$ of the particle without uncertainty at $t_2 = 1$. We also assume that $u = \mathbf{0}$, i.e. that the motion of the particle is Brownian. The aim is now to estimate the state probability density of the particle at $t = 0.5$ conditional on $\boldsymbol{z}_2$. The smoothing problem, known as a Brownian bridge, can be solved analytically and has a Gaussian posterior density with mean $\mu_x = 0.6 - 0.2t$, $\mu_y = 0.5$ and variance $\sigma^2 = 2Dt(1-t)$, where $D = 0.01$ is the diffusion coefficient of the Brownian motion which is assumed known and isotropic. The example resembles the situation where one wants to estimate the smoothed density between two observations which is often the case, in particular when analyzing sparse tracking data.

We compute the smoothed density with the FE approach as described in this text and compare it to the density estimated with an SMC method. Since resampling is not required we let the SMC implementation follow a sequential importance sampling scheme as in Cappé et al. (2007), their algorithm 2, using the prior kernel as the importance distribution and with marginal smoothing as proposed by Doucet et al. (2000). We set $M_* = M = 2665$, i.e. the number of SMC particles is equal to the number of nodes in the FE mesh. We construct a uniformly sized triangular FE mesh over the domain $\boldsymbol{x} \in \{0, \ldots, 1\}^2$. We define the FE time step $\delta = 0.014$ to obtain close to equal computation times for the two schemes. Observations without uncertainty are difficult to handle for both methods so the analyses are started at $t = 0.01$ and ended at $t = 0.99$ while adjusting the initial condition and the data likelihood to match the analytically known mean and variance at these times.

We compute a number of metrics that detail the comparison of the two methods. The deviation from the true mean and variance in the x direction are respectively

$$e_{\mu}^{(x)} = \mu_x - E(X), \qquad e_{\sigma^2}^{(x)} = \sigma^2 - V(X),$$

and similarly for the y direction we have

$$e_{\mu}^{(y)} = \mu_y - E(Y), \qquad e_{\sigma^2}^{(y)} = \sigma^2 - V(Y).$$

The Kullback–Leibler divergence, which can be regarded as an asymmetric distance between two probability densities, is

$$D(\Phi \parallel \phi) = \sum_{\boldsymbol{x}} \Phi(t, \boldsymbol{x}|z_2) \log \frac{\Phi(t, \boldsymbol{x}|\boldsymbol{z}_2)}{\phi(t, \boldsymbol{x}|\boldsymbol{z}_2)}.$$

The L^1 distance is

$$e_{L^1} = \sum_{\boldsymbol{x}} |\Phi(t, \boldsymbol{x}|\boldsymbol{z}_2) - \phi(t, \boldsymbol{x}|\boldsymbol{z}_2)|,$$

where $|\cdot|$ means absolute value.

For the FE approach we obtain $\phi(t, \boldsymbol{x}|\boldsymbol{z}_2)$ by interpolation of the nodal values via the linear basis functions to a rectangular grid $\boldsymbol{x} \in \{0, \ldots, 1\}^2$ with $n_x = n_y = 501$ grid points. For the SMC method we used kernel density estimation on the same grid with a two-dimensional Gaussian kernel with isotropic variance 0.0005. The kernel bandwidth was chosen to obtain the best approximation of the kernel density estimation to the true pdf (i.e. smallest L^1 and Kullback–Leibler divergence). The accuracy of the kernel density estimate is rather sensitive to the choice of bandwidth. It is therefore important in practice to employ some objective bandwidth selection protocol.

The results with computing times for filtering (T_{filt}) and smoothing (T_{smoo}) are summarized in Table 1. The SMC results are values averaged over 50 runs with corresponding empirical standard deviations. The FE solution gave slightly better

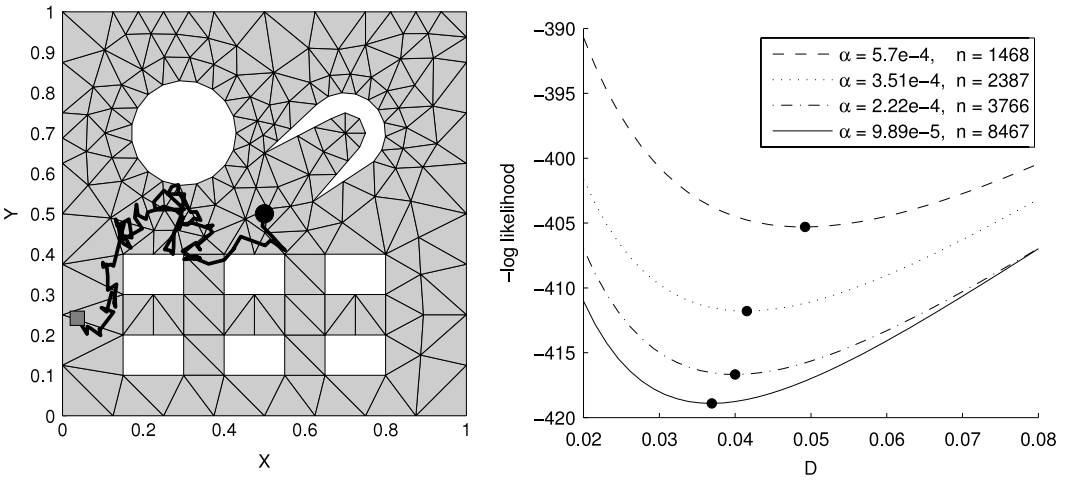

Fig. 1. Left: Coarsely meshed geometry of the artificial cityscape with the "true" simulated track of the test person. The solid circle represents the starting location; the shaded square represents the end location. White areas represent buildings, i.e. areas that are inaccessible to the test person. Right: Negative log-likelihood functions for D for meshes of different fineness indicated by α, the average element area, and n, the number of elements. It is noted that the ML estimates show convergence towards some value that is close to the true value of D.

estimates of the means while the SMC method was marginally better at estimating the variances. The density function itself was significantly better represented by the FE solution (by a factor of 33 for the Kullback–Leibler divergence and a factor of 7 for the L^1 distance). The SMC computing time was largely dominated by the smoothing step whereas the FE approach spent the time more evenly on the two steps. Owing to the $O(M_*^2)$ scaling, the computing effort of the SMC smoothing step may become prohibitively large for the SMC solution to reach a desired density accuracy. In such cases the FE approach is expected to deliver superior results. In fact, increasing M_* by a factor of 10 only led to an improvement in $D(\Phi \parallel \phi)$ by a factor of 2.5, and an improvement in e_{L^1} by a factor of 1.5 while increasing the total computing time of the SMC method by a factor of 100.

6.2. Synthetic data example

Now we address the behavior of the estimator in relation to the mesh fineness. For clarity we disregard the advection term, i.e. $u = \mathbf{0}$. We estimate the horizontal location $\boldsymbol{X}_k$ of a moving object, e.g. a human, for $k \in \{1, \ldots, N\}$, $N = 100$. Data consist of readings from an attached GPS device with constant time step. We thus have the observation equation

$$\boldsymbol{z}_k = \boldsymbol{X}_k + \boldsymbol{w}_k,$$

where $\boldsymbol{w}_k \sim N(\mathbf{0}, 0.01^2\boldsymbol{I})$. The movements of the object are Brownian with $D = 0.03\boldsymbol{I}$ and boundary behavior is implemented by disallowing steps outside of the model domain. Fig. 1 shows the artificial cityscape and the true simulated trajectory of the object.

We investigate how the posterior mode estimate and the likelihood function of D behave as functions of the fineness of the mesh quantified by the average element area α. We set $\boldsymbol{\theta} = D$, thus assuming that $\sigma_{\boldsymbol{w}}^2$ is known, and investigate the behavior of $l(\boldsymbol{\theta}|\mathcal{Z}_N)$ as a function of α; see Fig. 1. An indication of convergence towards a value close to the true value of D is observed as the mesh is refined. Furthermore, the likelihood of the parameter increases, an effect arising owing to the fact that the data likelihood is resolved more accurately. A coarse mesh results in overdispersion of the data likelihood which leads to a higher parameter estimate and a smaller likelihood value.

For each of the increasingly fine meshes we fix the value of D to the true value and compute the posterior mode. Define the track error metric

$$\varepsilon = 1/N \sum_{k=1}^{N} (\widehat{X}_k - X_k)^2,$$

where $\widehat{X}_k$ is the x coordinate of the posterior mode at t_k. The track error metric is computed analogously for the y coordinate. Fig. 2 shows how ε behaves as a function of α. Linear regression lines in the log–log domain fitted to the sloped part of the curves show an approximately linear order of convergence with visually estimated cut-offs that lie at $\alpha \approx 0.2 \cdot 10^{-4}$. The analytical variance of the posterior mode at t_k for Gaussian infinite domain problems (Thygesen and Nielsen, 2009) is given by

$$\sigma_{MPT}^2 = \sigma_{\boldsymbol{w}}^2 \frac{U(U + \sqrt{U^2 + 4})}{4 + U(U + \sqrt{U^2 + 4})} = 0.997 \cdot 10^{-4},$$

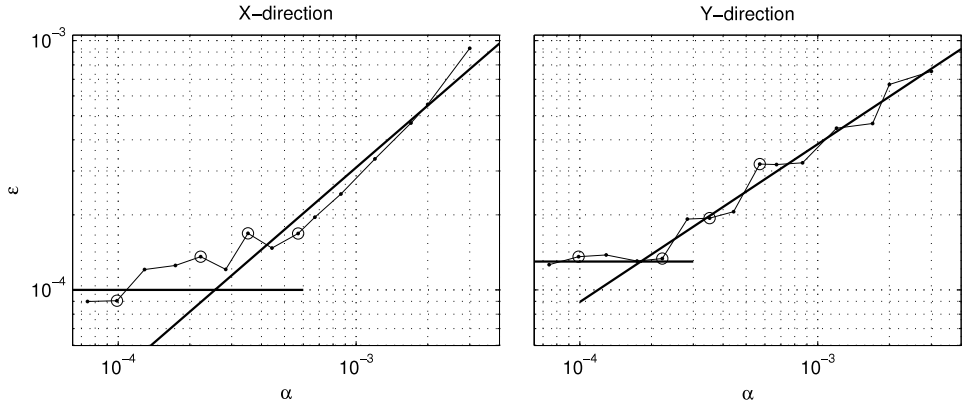

Fig. 2. Convergence plots of the track error metric of the posterior mode as a function of the resolution of the mesh (average element area, α) shown for the x and y coordinates. The four points highlighted with circles correspond to the likelihood curves in Fig. 1. With coarse grids ($\alpha > 2 \cdot 10^{-4}$), the track error is determined by the grid resolution. With fine grids, the error is largely independent of the grid and determined by the random measurement errors.

where $U = \sqrt{2Dh}/\sigma_{\boldsymbol{w}}$, with constant time step h and $\sigma^2_{\boldsymbol{w}}$ being the variance of the Gaussian noise, $\boldsymbol{w}_k$. The posterior distribution is not resolved accurately on the coarse mesh indicated by the values of ε becoming larger than their analytically expected value. As the mesh is refined this effect diminishes, as seen in Fig. 2, and ε converges to σ^2_{MPT}.

6.3. Real-data example

Finally we give a real-data application of the FE approach. We consider a geolocation problem which consists of estimating the large-scale horizontal movements of a fish based on observations from an electronic data storage tag (DST) attached to the fish while at liberty (Righton et al., 2006). The observations are readings of depth and salinity from which the horizontal movements of the fish can be inferred by comparing with databases that contain bathymetry and hydrographically modeled information for the salinity field. Here, a $N = 294$ data set from a cod in the Baltic Sea is analyzed. Fig. 3 depicts the FE mesh; note the highly complex boundary geometry which makes the governing partial equations difficult to solve for other methods. The mesh consists of 12 706 elements with a total of 7703 nodes and was refined recursively through two solutions of the problem. Again, we disregard the drift term, i.e. $u = \mathbf{0}$.

The data likelihood on a given day is computed by assuming that the observation $\boldsymbol{z}_k = (s_k, d_k)^T$, of salinity and depth respectively, is given by

$$\boldsymbol{z}_k = h(\boldsymbol{X}_k) + \boldsymbol{w}_k,$$

where $\boldsymbol{w}_k$ is Gaussian distributed with zero mean and covariance matrix

$$\boldsymbol{\Sigma}_{\boldsymbol{w}} = \begin{bmatrix} \sigma_s^2 & 0 \\ 0 & \sigma_d^2 \end{bmatrix}.$$

The values of σ_s^2 and σ_d^2 are considered as known and defined on the basis of the measurement uncertainty in the DST and the uncertainty within the databases. Including these in the estimation procedure is straightforward, although this leads to an increase in computation time.

The data likelihood for $j \in \{1, \ldots, M\}$ is

$$L(\boldsymbol{z}_k|\boldsymbol{x}_k^{(j)}) = \frac{1}{2\pi\sqrt{\det \boldsymbol{\Sigma}_{\boldsymbol{w}}}} \exp\left\{-\frac{1}{2}[\boldsymbol{z}_k - \widehat{\boldsymbol{z}}_k(\boldsymbol{x}_k^{(j)})]^T \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}[\boldsymbol{z}_k - \widehat{\boldsymbol{z}}_k(\boldsymbol{x}_k^{(j)})]\right\},$$

where $\widehat{\boldsymbol{z}}_k(\boldsymbol{x}_k^{(j)})$ is the expected observation of salinity and depth at location $\boldsymbol{x}_k^{(j)}$ given by the databases. In the data likelihood at t_1 and t_N we include information about the known release and recapture locations respectively.

MAP estimation of D using a uninformative prior converged at

$$\widehat{D} = \begin{bmatrix} 21.1 & -2.1 \\ -2.1 & 48.3 \end{bmatrix} \text{ km}^2 \text{ day}^{-1},$$

with $\text{sd}(\widehat{D}_{xx}) = 8.33 \text{ km}^2 \text{ day}^{-1}$, $\text{sd}(\widehat{D}_{yy}) = 19.2 \text{ km}^2 \text{ day}^{-1}$ and $\text{sd}(\widehat{D}_{xy}) = 10.6 \text{ km}^2 \text{ day}^{-1}$ estimated by the inverse Hessian of the likelihood function, i.e. the observed Fisher information.

The posterior mode is depicted in Fig. 4. Note that the track, most prominently in the final part, at times crosses tounges of dry land. These steps seem unlikely and result from coarse temporal resolution which allows the fish to move around narrow tongues of land within the time span of two observations. A simple remedy is to increase the temporal resolution of the track estimation procedure. Still, the posterior mode gives an overall sensible impression of the movements of the fish.

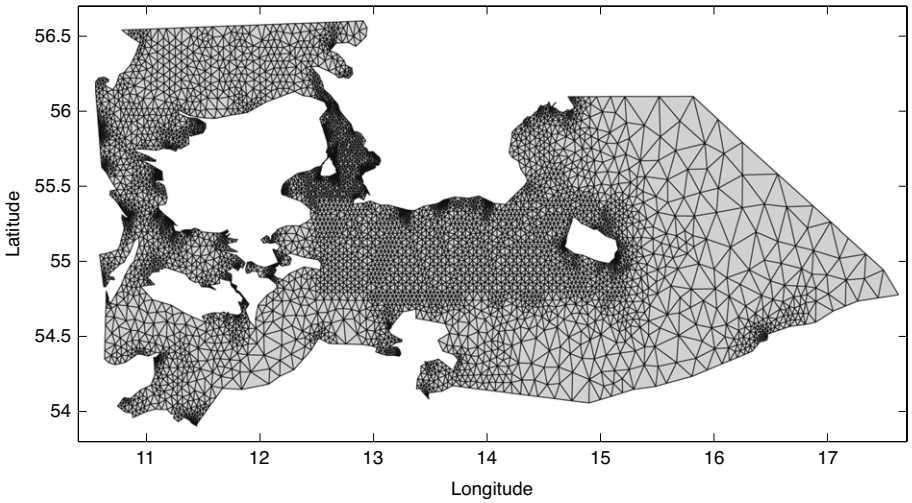

Fig. 3. The FE-triangulated mesh. Refinements to the mesh have been made in the central part of the Baltic Sea and around Zealand, particularly in Øresund, the sound between Zealand and Sweden. The mesh consists of 12 706 elements with a total of 7703 nodes.

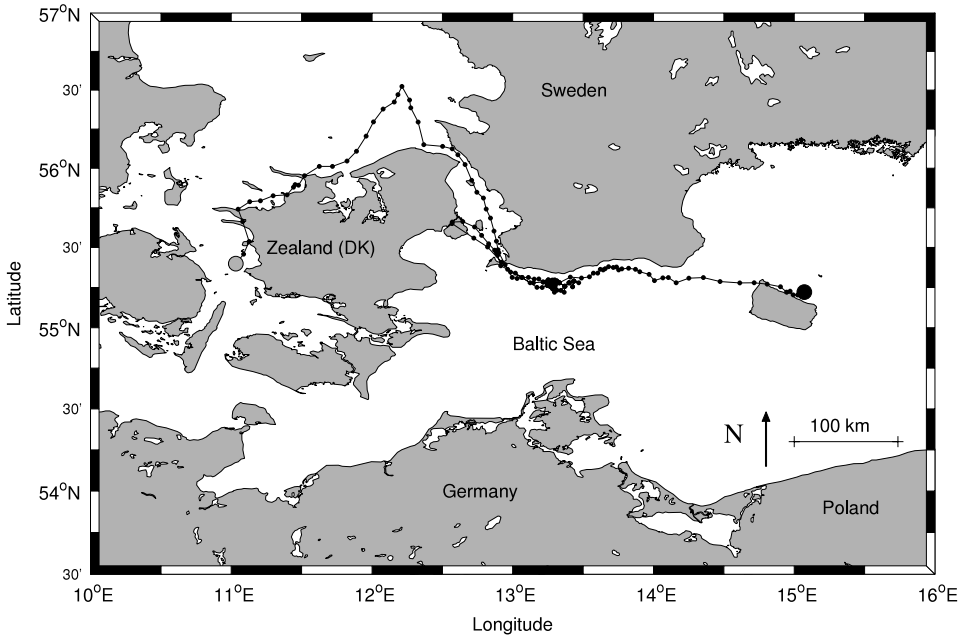

Fig. 4. Fish movements illustrated by the posterior mode estimated with the Viterbi algorithm suggesting that the fish, released at the solid circle at 55°15′N, 15°E and recaptured at the shaded circle at 55°30′N, 11°E, resided in the Baltic Sea at 13°E before migrating around Zealand via a northerly route.

A feature of this problem is that much information is contained in the recapture location at the end of the track, and thus the filtered locations differ markedly from the smoothed locations. As a result of this feature, an SMC method for this problem would require a large number of particles for smoothing.

The calculations were conducted on a laptop PC with a 1.4 GHz Centrino CPU and 2 GB RAM. Matlab 7.4 was used as computing environment leading to computation times on the scale of minutes for the smoothed posterior density, hours for parameter estimation and days for track estimation.

7. Conclusion

The FE approach to nonlinear state estimation problems presented is an alternative to existing SMC and point-mass filters. It is a deterministic method for solving the Kolmogorov equations with the ability to refine the numerical discretization at

complex boundary geometries and in areas where a more accurate solution is required. The discretization remains fixed throughout the computations, thus simplifying the estimation of smoothed state densities and MAP parameter values. The FE method differs markedly from SMC in terms of the effort required for implementation. While simple particle filters are straightforward to implement, greater effort is required for accurate smoothing and parameter estimation, in particular for problems such as the real-data example of Section 6.3. In contrast, with the FE method, the majority of the effort is spent at an early stage when implementing the solution scheme for the governing Kolmogorov equations. For some applications, for example oceanography which motivated this work, discretizing and solving partial differential equations is standard and optimized code may even be available. This would argue in favour of the FE method. On the other hand, for high state dimensions, the FE method becomes increasingly complex to implement and expensive to compute and is therefore best suited for problems on lower-dimensional state-spaces.

The two-dimensional Brownian bridge example showed that the SMC and FE smoothing methods performed equally well for moment estimation but for density estimation, the FE approach was superior by a factor of 33 with respect to the Kullback–Leibler divergence and a factor of 7 with respect to the L^1 distance for equal computation times. It is expected that the FE method will be preferable for smoothing problems, in particular when high density accuracy is required. The synthetic data example illustrated the convergence properties of the FE estimators where an approximately linear relation was observed for the mean square error of the posterior mode in relation to the average element area in the sloped part of the curve. Unlike SMC methods, the FE approach provides a deterministic likelihood function of the model parameters which, conditional on the fixed mesh, is simple to optimize numerically. On refining the mesh, the likelihood function converged towards the true value until plateauing when the optimally resolved density was reached. The real-data example showed how a simple implementation of the FE approach was used to solve a tracking problem involving complex state-space boundary geometry, parameter estimation and state density smoothing which would have been challenging to do with other available methods.

Acknowledgements

The authors thank the associate editor and two anonymous reviewers for their very helpful comments that significantly improved the quality of this text.

Appendix

This Appendix summarizes the use of the FE method for advection–diffusion partial differential equations (PDEs). To solve (3) with the FE method, we derive its weak formulation. For transparency we consider only the case where parameters are homogeneous in time and space. Comprehensive treatment of the FE method can be found in e.g. Cook et al. (2001) and Reddy and Gartling (2001).

For the PDEs (3) and (6) we assume no-flux boundary conditions, i.e. it holds for u that

$$(u\phi - D\nabla\phi) \cdot n = 0, \tag{10}$$

at any boundary point where n is a vector normal to the boundary $\partial\Omega$. The weak formulation of (3) is obtained by constructing the inner product with v_i:

$$\langle v_i, \dot{\phi}\rangle = \langle v_i, -\nabla \cdot (u\phi - D\nabla\phi)\rangle. \tag{11}$$

This equation is reduced by first considering the LHS of (11) and inserting (8), thereby obtaining after simplifications

$$\langle v_i, \dot{\phi}\rangle = \sum_{j=1}^{M} \dot{\beta}_j(t)\langle v_i, v_j\rangle.$$

The diffusion part of the RHS becomes

$$\begin{aligned}\langle v_i, \nabla \cdot (D\nabla\phi)\rangle &= \sum_{j=1}^{M} \beta_j(t)\langle v_i, \nabla \cdot (D\nabla v_j)\rangle \\ &= \sum_{j=1}^{M} \beta_j(t)\left[-\langle \nabla v_i, D\nabla v_j\rangle + \int_{\partial\Omega} v_i D\nabla v_j \mathrm{d}n\right].\end{aligned}$$

Due to the no-flux boundary condition, the term $\int_{\partial\Omega} v_i D\nabla v_j \mathrm{d}n$ vanishes. The advection part of the RHS becomes

$$\begin{aligned}\langle v_i, \nabla\cdot(u\phi)\rangle &= \sum_{j=1}^{M}\beta_j(t)\langle v_i, \nabla\cdot(uv_j)\rangle \\ &= \sum_{j=1}^{M}\beta_j(t)\left[-\langle\nabla v_i, uv_j\rangle + \int_{\partial\Omega} v_i u v_j \mathrm{d}n\right].\end{aligned}$$

Again, due to the no-flux boundary condition, the term $\int_{\partial\Omega} v_i u v_j \mathrm{d}n$ vanishes. Collecting the above terms we obtain

$$\sum_{j=1}^{M}\dot{\beta}_j(t)\langle v_i, v_j\rangle = \sum_{j=1}^{M}\beta_j(t)\left[\langle\nabla v_i, uv_j - D\nabla v_j\rangle\right], \tag{12}$$

which is a system of ordinary differential equations that can be solved for $\beta_j(t)$ with standard numerical methods. With

$$c_{ij} = \langle v_i, v_j\rangle$$

and

$$a_{ij} = \langle\nabla v_i, D\nabla v_j - uv_j\rangle,$$

we can express (12) in matrix notation as

$$\boldsymbol{C}\dot{\boldsymbol{\beta}}(t) = -\boldsymbol{A}\boldsymbol{\beta}(t), \tag{13}$$

with

$$\boldsymbol{\beta}(t) = \begin{bmatrix}\beta_1(t)\\ \vdots \\ \beta_j(t)\end{bmatrix}, \qquad \boldsymbol{C} = \begin{bmatrix}c_{11} & \cdots & c_{1j}\\ \vdots & \ddots & \vdots \\ c_{i1} & \cdots & c_{ij}\end{bmatrix}, \qquad \boldsymbol{A} = \begin{bmatrix}a_{11} & \cdots & a_{1j}\\ \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ij}\end{bmatrix}.$$

There are various ways of solving (13) numerically. To avoid unstable, oscillating, and possibly negative solutions for the density, we utilize a simple implicit Euler scheme

$$\frac{1}{\delta}(\boldsymbol{\beta}_{t_k+(m+1)\delta} - \boldsymbol{\beta}_{t_k+m\delta}) = -\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\beta}_{t_k+(m+1)\delta} \quad m\in\{0,\ldots,T-1\}, \tag{14}$$

where $\boldsymbol{\beta}_t = \boldsymbol{\beta}(t)$, δ is the FE time step and $T\delta = t_{k+1} - t_k$. The choice of δ and the fineness of the mesh determines the magnitude of the approximation error of the solution as discussed in Cook et al. (2001). The appropriate value of T depends on the required accuracy of the solution and on the computing speed. Reformulating and simplifying (14) gives

$$\boldsymbol{\beta}_{t_k+(m+1)\delta} = \boldsymbol{R}^{-1}\boldsymbol{C}\boldsymbol{\beta}_{t_k+m\delta}, \tag{15}$$

where $\boldsymbol{R} = \boldsymbol{C}+\delta\boldsymbol{A}$ is called the coefficient matrix. The matrices involved are sparse for large systems. It is therefore important to exploit this in the implementation to save memory and reduce the number of trivial computations.

In order to solve (6), close to identical derivations can be made.

References

Bucy, R.S., Senne, K.D., 1971. Digital synthesis of non-linear filters. Automatica 7, 287–298.

Cappé, O., Godsill, S.J., Moulines, E., 2007. An overview of existing methods and recent advances in sequential Monte Carlo. IEEE Proc. Signal Proc. 95 (5), 899–924.

Cappé, O., Moulines, E., Rydén, T., 2005. Inference in Hidden Markov Models. Springer, New York, NY.

Clough, R.W., 1980. The finite element method after twenty-five years: a personal view. Comput. Struct. 12 (4), 361–370.

Comaniciu, D., Ramesh, V., Meer, P., 2003. Kernel-based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. 564–577.

Cook, R.D., Malkus, D.S., Plesha, M.E., Witt, R.J., 2001. Concepts and Applications of Finite Element Analysis. John Wiley & Sons Inc., Hoboken, NJ.

Creal, D., 2008. Analysis of filtering and smoothing algorithms for Lévy-driven stochastic volatility models. Comput. Statist. Data Anal. 52 (6), 2863–2876.

Doucet, A., Godsill, S., Andrieu, C., 2000. On sequential Monte Carlo sampling methods for Bayesian filtering. Stat. Comput. 10 (3), 197–208.

Doucet, A., Tadić, V., 2003. Parameter estimation in general state-space models using particle methods. Ann. Inst. Statist. Math. 55 (2), 409–422.

Golightly, A., Wilkinson, D., 2008. Bayesian inference for nonlinear multivariate diffusion models observed with error. Comput. Statist. Data Anal. 52 (3), 1674–1693.

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. Trans. ASME 82 (1), 35–45.

Kitagawa, G., 1987. Non-Gaussian state-space modeling of nonstationary time series. J. Amer. Statist. Assoc. 82 (400), 1032–1041.

Ljung, L., 1997. System Identification: Theory for the User, 2nd ed.. Prentice-Hall, Upper Saddle River, NJ.

Øksendal, B., 2007. Stochastic Differential Equations. Springer-Verlag, New York, NY.

Pearson, J., Stear, E., 1974. Kalman filter applications in airborne radar tracking. IEEE Trans. Aerosp. Electron. Syst. 319–329.

Reddy, J.N., Gartling, D.K., 2001. The Finite Element Method in Heat Transfer and Fluid Dynamics. CRC Press, New York, NY.

Righton, D., Kjesbu, O.S., Metcalfe, J.D., 2006. A field and experimental evaluation of the effect of data storage tags on the growth of cod. J. Fish Biol. 68, 385–400.

Ristic, B., Arulampalam, S., Gordon, N., 2004. Beyond the Kalman Filter. Particle Filters for Tracking Applications. Artech House Publishers, Boston, MA.

Shewchuk, J.R., 1996. Triangle: engineering a 2D quality mesh generator and Delaunay triangulator. In: Lin, M.C., Manocha, D. (Eds.), Applied Computational Geometry: Towards Geometric Engineering. In: Lecture Notes in Computer Science, vol. 1148. Springer-Verlag, pp. 203–222. from the First ACM Workshop on Applied Computational Geometry.

Sibert, J.R., Musyl, M.K., Brill, R.W., 2003. Horizontal movements of bigeye tuna (*Thunnus obesus*) near Hawaii determined by Kalman filter analysis of archival tagging data. Fish. Oceanogr. 12, 141–151.

Šimandl, M., Královec, J., Söderström, T., 2006. Advanced point-mass method for nonlinear state estimation. Automatica 42 (7), 1133–1145.

Thygesen, U., Nielsen, A., 2009. Lessons from a prototype geolocation problem. In: Nielsen, J., Arrizabalaga, H., Fragoso, N., Hobday, A., Lutcavage, M., Sibert, J. (Eds.), Tagging and Tracking of Marine Animals with Electronic Devices. In: Reviews: Methods and Technologies in Fish Biology and Fisheries, vol. 9. Springer, pp. 257–276.

Thygesen, U.H., Pedersen, M.W., Madsen, H., 2009. Geolocating fish using hidden Markov models and data storage tags. In: Nielsen, J., Arrizabalaga, H., Fragoso, N., Hobday, A., Lutcavage, M., Sibert, J. (Eds.), Tagging and Tracking of Marine Animals with Electronic Devices. In: Reviews: Methods and Technologies in Fish Biology and Fisheries, vol. 9. Springer, pp. 277–293.

Viterbi, A.J., 2006. A personal history of the Viterbi algorithm. IEEE Signal Process. Mag. 23 (4), 120–123.