

Nonlinear Filtering Using a New Proposal Distribution and the Improved Fast Gauss Transform With Tighter Performance Bounds

Roni Mittelman, *Student Member, IEEE*, and Eric L. Miller, *Senior Member, IEEE*

Abstract—We propose a new approach to approximating the Chapman–Kolmogorov equation (CKE) for particle-based nonlinear filtering algorithms, using a new proposal distribution and the improved fast Gauss transform (IFGT) with tighter accuracy bounds. The new proposal distribution, used to obtain a Monte Carlo (MC) approximation of the CKE, is based on the proposal distribution found in the auxiliary marginal particle filter (AMPF). By using MC integration to approximate the integrals of the AMPF proposal distribution as well as the CKE, we demonstrate significant improvement in terms of both error and computation time. We consider the additive state noise case where the evaluation of the CKE is equivalent to performing kernel density estimation (KDE), thus fast methods such as the IFGT can be used. In practice, the IFGT demonstrates performance far better than that which is predicted by current error analysis, therefore the existing bounds are not useful for determining the IFGT parameters which in practice have to be chosen experimentally in order to obtain satisfactory compromise between accuracy and speed of the filtering algorithm. We provide in this paper much improved performance bounds for the IFGT, and which unlike the previous bound, are consistent with the expectation that the error decreases as the truncation order of the IFGT increases. The new bounds lead to a new definition of the IFGT parameters and give important insight into the effect that these parameters have on the error, therefore facilitating the choice of parameters in practice. The experimental results show that we can obtain similar error to the sequential importance sampling (SIS) particle filter, while using fewer particles. Furthermore the choice of the new IFGT parameters remains roughly the same for all the examples that we give.

Index Terms—Fast Gauss transform, improved fast Gauss transform, kernel density estimation, nonlinear filtering, particle filters.

Manuscript received December 27, 2007; revised July 03, 2008. First published August 29, 2008; current version published November 19, 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Manuel Davy. This work was supported by the Center for Subsurface Sensing and Imaging Systems under the Engineering Research Centers Program of the National Science Foundation (Award Number EEC-9986821) and Textron Systems Division. Part of this work was presented at the IEEE Statistical Signal Processing Workshop 2007.

R. Mittelman is with the Department of Electrical and Computer Engineering, Northeastern University, 409 Dana Research Center, Boston MA 02115 USA (e-mail: rmittelm@ece.neu.edu).

E. L. Miller is with the Department of Electrical and Computer Engineering, Tufts University, Medford MA 02155 USA (e-mail: elmiller@ece.tufts.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSP.2008.2005095

I. INTRODUCTION

NONLINEAR filtering is concerned with dynamic systems of the form

$$x_k = f(x_{k-1}, w_k) \quad (1)$$

$$z_k = g(x_k, v_k) \quad (2)$$

where x_k and z_k denote the state and measurement vectors at time k respectively, w_k , and v_k denote two independent noise sequences, and $f()$, $g()$ can be any two functions of the state vector. The aim of filtering is to find the posterior state distribution $p(x_k|z_{1:k})$, where $z_{1:k} \triangleq \{z_1, \dots, z_k\}$. The posterior state distribution can be computed recursively using [11]

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \quad (3)$$

$$p(x_k|z_{1:k}) = \frac{1}{c_k} p(z_k|x_k)p(x_k|z_{1:k-1}) \quad (4)$$

$$c_k \triangleq \int p(z_k|x_k)p(x_k|z_{1:k-1})dx_k. \quad (5)$$

When the state and measurement equations are linear and the noises are Gaussian, (3)–(5) can be evaluated analytically using the Kalman filter [1]; otherwise, the recursive equations can only be approximated. One common approach is the extended Kalman filter (EKF) [1] where the nonlinear functions in (1) and (2) are replaced with a first-order Taylor series approximation and used in the Kalman filter. Another common approach is particle filtering or sequential Monte Carlo (SMC) methods [8], [21]. Most particle filters approximate the joint density $p(x_{1:k}|z_{1:k})$ although only the marginal $p(x_k|z_{1:k})$ is of interest, thereby necessitating the use of a resampling stage. The other paradigm to particle filtering [2], [11] approximates the marginal $p(x_k|z_{1:k})$ directly and was termed the marginal particle filter (MPF) in [2]. The MPF uses Monte Carlo (MC) integration and importance sampling (IS) to approximate the integrals in the Bayes recursion equations (3)–(5). When using the MPF, there are two challenges that have to be addressed, choosing a proposal distribution and evaluating the Chapman–Kolmogorov equation (CKE) (3) in a computationally efficient manner especially for high dimensional problems.

Using IS requires the definition of a proposal distribution from which one draws the samples that are used to obtain the

MC approximation of the integral. Although in theory any probability density function (PDF) with the same support as the posterior can be used, in practice due to the limited computational resources it is advisable that the proposal be as similar as possible to the posterior in order to obtain the best performance. In this work we develop a new proposal distribution for MC integration using IS.

The computational complexity of evaluating (3) N times is $O(N^2)$, where N is the number of samples used to represent the PDF in each time step. Most previous work employing the MPF [11] considered the one and two dimensional cases where the number of particles is relatively small, thus (3) was evaluated directly. In [2] the fast Gauss transform (FGT) [5], which is an efficient kernel density estimation (KDE) method, was used to speed up the computation when the state noise is assumed Gaussian and additive. KDE methods have also been used in other nonlinear filtering algorithms besides the MPF [19], [20]; however, whereas the later methods use the KDE as a part of the resampling stage, in [2] the KDE was regarded as another interpretation of the CKE. In this work we consider the use of the improved fast Gauss transform (IFGT) [7] to reduce the computational complexity of the CKE. Similarly to the FGT, the IFGT can be used to perform fast KDE; however, its computational constant grows more moderately with the dimension compared to the FGT for which the computational constant grows exponentially with the dimension. In practice, the IFGT demonstrates performance which is far better than the existing performance bounds; thus, they are not useful to determining the truncation order and the number of clusters that should be used to clusters the source points. In this work, we develop new performance bounds which are significantly tighter and which, unlike the old bounds, agree with our expectation that the error decreases as the truncation order increases. The new bounds lead to a new definition of the IFGT parameters and give new insight into the effect that these parameters have on the error, thus facilitating the experimental choice of the IFGT parameters.

Our experimental results show that using the new framework we can obtain similar error and timing results to the sequential importance sampling (SIS) particle filter; however, the number of particles can be reduced significantly. This may prove to be significant in practical real time implementation as the storage complexity is reduced significantly. Furthermore, when considering filtering in sensor networks, power conservation requirements necessitate using as few particles as possible [22] since the particles need to be transmitted between the different nodes in the network.

The remainder of this paper is organized as follows. In Section II, we give the background on the FGT and the IFGT and provide background on MC integration and IS. In Section III, we describe the nonlinear filtering algorithm and the new proposal distribution. In Section IV, we derive new upper error bounds for the IFGT, and in Section V we analyze the new bounds and explain how to choose the IFGT parameters based on the new bounds. Section VI presents experimental results for the nonlinear filtering algorithm. Finally, Section VII concludes this paper.

II. BACKGROUND

In this section, we provide the background on KDE and the FGT and IFGT algorithms, as well as the background on Monte Carlo integration and importance sampling which are used in the nonlinear filtering algorithm.

A. KDE

KDE is most often used to determine a nonparametric PDF from a set of samples $\{s_j\}_{j=1}^N$ [3]. The KDE evaluated at a target point t then takes the form

$$G(t) \triangleq \sum_{j=1}^N q_j K\left(\frac{t - s_j}{h}\right) \quad (6)$$

where $K(\cdot)$ is a kernel function with scale parameter h (the “bandwidth”), $s_j \in \mathbb{R}^D$ are referred to as source points, $q_j \in \mathbb{R}$ are the source strengths, and D denotes the dimension. For the purpose of obtaining a good nonparametric estimate of the PDF, the optimal bandwidth h^* has to be estimated [9]. Evaluating $G(\cdot)$ in (6) at N different target points has quadratic computational complexity, therefore there have been several approaches suggested in literature to reduce this computational complexity while compromising on the accuracy.

The different methods to fast evaluation of KDE rely on the divide and conquer approach where either the source space or the joint source and target space are first partitioned into different regions. Beyond the initial partitioning the various algorithms can be categorized into discrete approximation techniques such as the dual tree algorithm (DTA) [4], [9] and continuous approximation algorithms such as the FGT and IFGT. The DTA uses tight upper error bounds to approximate large clusters in the joint source target space using a centroid approximation. Furthermore, the DTA can be used under several choices of kernel functions and its speed is independent of the dimension. However, significant speed-ups are usually obtained only when the bandwidth is smaller than the optimal bandwidth [9]. Since for the nonlinear filtering case, the bandwidth is determined by the state’s noise standard deviation, this may significantly limit the applicability of the DTA for cases where the state’s noise standard deviation is larger than the optimal bandwidth for the particles. The FGT applies only in the case of a Gaussian kernel where a Hermite polynomial expansion is used to approximate the Gaussian kernel function beyond the initial space partitioning. The Hermite expansion enables (6) to be approximated to any degree of accuracy using an alternative expression whose computational complexity is linear in the number of samples, however the computational constant grows exponentially with the dimension. The IFGT solves this problem by using a different form of expansion for the Gaussian kernel such that the computational constant grows more moderately with the dimension. Both the FGT and IFGT can offer a significant speedup when the bandwidth is larger than the optimal bandwidth.

1) *The FGT*: The FGT considers the special case where the kernel function in (6) is a Gaussian:

$$G(t) = \kappa \sum_{j=1}^N q_j e^{-\|t-s_j\|^2/2\sigma^2} \quad (7)$$

where σ^2 denotes the variance of the Gaussian kernel, and $\kappa^{-1} = (2\pi\sigma^2)^{D/2}$. The basis for the FGT is the one dimensional Hermite expansion of the Gaussian function around point $s^* \in \mathbb{R}$:

$$e^{-(t-s)^2/2\sigma^2} = \sum_{n=1}^{\infty} \frac{1}{n!} \left(\frac{s-s^*}{\sqrt{2\sigma}} \right)^n h_n \left(\frac{t-s^*}{\sqrt{2\sigma}} \right) \quad (8)$$

where $t, s \in \mathbb{R}$ and where $h_\alpha(t) \triangleq (-1)^\alpha d^\alpha / dt^\alpha e^{-t^2}$, and satisfies the recursion: $h_{\alpha+1}(t) = 2th_\alpha(t) - 2\alpha h_{\alpha-1}(t)$.

The extension to the multidimensional case is then obtained using the products of the one dimensional expansions. We adopt the multi-index notation used in [5], where $\alpha = (\alpha_1, \dots, \alpha_D)$ is a D dimensional tuple of nonnegative integers. For any $t \in \mathbb{R}^D$ we define $t^\alpha = t_1^{\alpha_1} \dots t_D^{\alpha_D}$, and $h_\alpha(t) = h_{\alpha_1}(t_1) \dots h_{\alpha_D}(t_D)$. The factorial of the multi-index α is defined as $\alpha! = \alpha_1! \dots \alpha_D!$, and the length is defined as $|\alpha| = \alpha_1 + \dots + \alpha_D$. The multidimensional expansion of a Gaussian around $s^* \in \mathbb{R}^D$ therefore takes the form

$$e^{-\frac{\|t-s\|^2}{2\sigma^2}} = \sum_{\alpha \geq 0} \frac{1}{\alpha!} h_\alpha \left(\frac{\Delta t}{\sqrt{2\sigma}} \right) \left(\frac{\Delta s}{\sqrt{2\sigma}} \right)^\alpha \quad (9)$$

where $t, s \in \mathbb{R}^D$, $\Delta t = t - s^*$, and $\Delta s = s - s^*$.

Truncating (9) after the first p terms and substituting into (7) yields

$$G(t) = \kappa \cdot \sum_{|\alpha| < p} A_\alpha h_\alpha \left(\frac{\Delta t}{\sqrt{2\sigma}} \right) + E_T \quad (10)$$

$$A_\alpha = \frac{1}{\alpha!} \sum_{j=1}^N q_j \left(\frac{\Delta s_j}{\sqrt{2\sigma}} \right)^\alpha \quad (11)$$

where $\Delta s_j = s_j - s^*$, E_T is the truncation error, and A_α are the FGT coefficients.

Let N_B sources with weights $\{q_j\}_{j=1}^{N_B}$ lie in a D dimensional box with center s_B and side lengths $2r\sigma$, with $r < 1$. Then the error E_T due to truncating the series (10) after p^D terms satisfies the bound [6]

$$|E_T| \leq Q_B c_p^{\text{FGT}} \quad (12)$$

where

$$c_p^{\text{FGT}} = \frac{\kappa}{(1-r)^D} \sum_{d=0}^{D-1} \binom{D}{d} (1-r^p)^d \left(\frac{r^p}{\sqrt{p!}} \right)^{D-d} \quad (13)$$

and where $Q_B = \sum_{j=1}^{N_B} |q_j|$

The FGT partitions the source space into D dimensional boxes of side $2r\sigma$ with $r < 1$, which are parallel to the axes.

Each source point s_j is then assigned to the box in which it lies, and the FGT coefficients (11) for each box are computed using the sources assigned to it where the s^* for each box is the center of the box. The FGT evaluates each of the target points at each of the $(2n+1)^D n \in \mathbb{N}$ nearest boxes, which adds an error which is bounded by $Q e^{-2r^2 n^2}$ due to ignoring all the source points in the other boxes. The truncation error incurred by evaluating the target points at the $(2n+1)^D$ nearest boxes using (10) is bounded by $Q c_p^{\text{FGT}}$, where $Q = \sum_{j=1}^N |q_j|$. Therefore, the FGT can approximate the KDE to any degree of accuracy.

As can be seen from (10) and (11), the computational complexity of the FGT is linear with the number of samples; however, since the number of FGT coefficients grows exponentially with the dimension so does the computational constant. Since (10) has to be evaluated at each cluster, the computational complexity of the FGT also grows linearly with the number of clusters. The FGT can also make use of translation operators which can reduce the computational cost; however, the computational constant still grows exponentially with the dimension.

2) *IFGT*: The IFGT [7] also deals with the Gaussian kernel case, however it uses another form of expansion around a point $s^* \in \mathbb{R}^D$ instead of the Hermite expansion (9)

$$\begin{aligned} e^{-\|t-s\|^2/2\sigma^2} &= e^{(-\|\Delta t\|^2 - \|\Delta s\|^2 + 2\Delta t^T \Delta s)/2\sigma^2} \\ &= e^{-\frac{\|\Delta t\|^2 + \|\Delta s\|^2}{2\sigma^2}} \sum_{n=0}^{\infty} \frac{2^n}{n!} \left(\frac{\Delta t \Delta s}{2\sigma^2} \right)^n \end{aligned} \quad (14)$$

where $t, s \in \mathbb{R}^D$, $\Delta t = t - s^*$, $\Delta s = s - s^*$, and where (14) follows from expanding the exponent of the cross term using a Taylor expansion. Truncating (14) after the first p terms and substituting into (7), we have:

$$G(t) = \kappa e^{-\|\Delta t\|^2/2\sigma^2} \cdot \sum_{|\alpha| < p} \tilde{A}_\alpha \left(\frac{\Delta t}{\sigma} \right)^\alpha + E_T \quad (15)$$

$$\tilde{A}_\alpha = \frac{1}{\alpha!} \sum_{j=1}^N \tilde{q}_j \left(\frac{\Delta s_j}{\sigma} \right)^\alpha \quad (16)$$

where $\tilde{q}_j = q_j e^{-\|\Delta s_j\|^2/2\sigma^2}$. Since the summation in (15) is over all the IFGT coefficients with multi-index α such that $|\alpha| < p$, then the computational constant grows more moderately with the dimension compared to the computational constant of the FGT that grows exponentially with the dimension.

Let r_t and r_s denote upper bounds on $\|\Delta t\|$ and $\|\Delta s\|$ respectively, and let N_B sources with weights $\{q_j\}_{j=1}^{N_B}$ lie in a ball with center s_B and radius r_s . Then the error E_T due to truncating the series (15) after the p th order, when evaluating a target point within distance r_t from the ball's center, satisfies the bound [7]

$$|E_T| \leq Q_B c_p^{\text{IFGT}} \quad (17)$$

where

$$c_p^{\text{IFGT}} = \kappa \frac{2^p}{p!} \left(\frac{r_t}{\sigma} \right)^p \left(\frac{r_s}{\sigma} \right)^p \quad (18)$$

and where $Q_B = \sum_{j=1}^{N_B} |q_j|$.

The IFGT clusters the source points into balls with radius r_s and computes the IFGT coefficients \hat{A}_α for each ball using (16). For each target point the IFGT finds the clusters whose centers lie within the range r_t , and evaluates the target point using (15) at each of these clusters. The error due to ignoring all the source points outside of range r_t from the clusters' centers is bounded by $Qe^{-(r_t-r_s)^2/2\sigma}$ [14]. The truncation error incurred by evaluating the target point at each of the clusters is bounded by $Q\epsilon_p^{\text{IFGT}}$, where $Q = \sum_{j=1}^N |q_j|$. Therefore, for every choice of r_s , r_t and p , we can bound the approximation error of the KDE. However since the truncation error and the error due to ignoring the source points outside of range r_t are coupled, it is unclear from this formulation how to choose r_s and r_t such that the approximation is smaller than a specified value $Q\epsilon$. Furthermore as was noted in [14], [17] the bound (17) is very pessimistic. An approach to determining the truncation order and the parameters r_t, r_s which is based on a bound which is tighter than (17) was presented in [17]. First in order to decouple the truncation error and the error due to ignoring source points outside of range r_t , it is proposed to use r_t of the form

$$r_t = r_s + R \quad (19)$$

where $R \in \mathbb{R}$, in which case the error due to ignoring source points outside of range r_t is $Qe^{-R^2/2\sigma}$ and we can choose R such that this error is smaller than $Q\epsilon$. Second, the source points are clustered into K balls where K is chosen according to some heuristic criteria which is related to the computational constant of the IFGT rather than its accuracy. The parameter r_s for each ball can then be obtained as the radius of the ball. The truncation order p for each cluster is then determined such that $\epsilon_p^{\text{IFGT}_2}(\|\Delta s_j\|/\sigma) < \epsilon \forall j = 1, \dots, N_B$, where the truncation error for the Gaussian kernel is given by

$$\epsilon_p^{\text{IFGT}_2}(r_0) = \kappa \frac{2^p}{p!} (r_0 \|\tau\|_*) e^{-(r_0 - \|\tau\|_*)^2} \quad (20)$$

where

$$\|\tau\|_* = \min \left(\frac{r_0 + \sqrt{r_0^2 + 2p}}{2}, \frac{r_t}{\sigma} \right). \quad (21)$$

The truncation error E_T in (15) is therefore bounded by

$$|E_T| \leq \sum_{j=1}^N |q_j| \epsilon_p^{\text{IFGT}_2}(\|\Delta s_j\|/\sigma) \quad (22)$$

$$\leq Q\epsilon. \quad (23)$$

Since the number of clusters is chosen based on a heuristic criteria rather than the bound and subsequently the truncation order is chosen such that a specified upper bound is satisfied, there is no guarantee that this choice is optimal under any criteria. Furthermore, since in practice for the nonlinear filtering case the IFGT parameters p, r_s, r_t are chosen experimentally, the current bounds provide us with very little insight into how they should be chosen. In Section IV, we present new bounds

and show that they are significantly tighter than (20) and (22). In Section V, we analyse the new bounds and demonstrate that the new bounds also provide us with important insight into the choice of the IFGT parameters. The new bounds also lead to a new definition of the IFGT parameters using which we show that one can easily approximate the KDE to any specified order of accuracy. In Section VI, we demonstrate experimentally that the new parameters remain roughly the same for different experimental settings.

B. MC Integration and IS

Consider the generic problem of evaluating integrals of the form

$$I = \mathbb{E}_f[h(x)] = \int h(x)f(x)dx. \quad (24)$$

and suppose that $f(\cdot)$ is a valid PDF. The MC framework approximates (24) by drawing the samples (x_1, \dots, x_N) from $f(x)$, and approximating (24) using the empirical average

$$I_N = \frac{1}{N} \sum_{i=1}^N h(x_i) \quad (25)$$

where I_N converges almost surely to I [10].

When it is difficult to sample from the distribution $f(x)$, then IS can be used to approximate the integral. Assume that we can obtain samples from a proposal distribution $\pi(x)$ that satisfies the condition $\text{support}(\pi) \supseteq \text{support}(f)$, then by rewriting (24) as

$$I = \mathbb{E}_\pi \left[h(x) \frac{f(x)}{\pi(x)} \right] = \int h(x) \frac{f(x)}{\pi(x)} \pi(x) dx \quad (26)$$

we can use the MC integration framework to approximate (26). By drawing the samples (x_1, \dots, x_N) from $\pi(x)$ then the approximation takes the form

$$I_N = \frac{1}{N} \sum_{i=1}^N h(x_i) \frac{f(x_i)}{\pi(x_i)} \quad (27)$$

and since (27) is a Monte Carlo estimator it converges almost surely to I . Although the IS framework applies to a broad range of proposal distributions, the convergence rate depends on how close the proposal distribution $\pi(x)$ is to $f(x)$. For poor choices of $\pi(x)$ the convergence rate may be quite slow.

Another important factor influencing the quality of the MC approximation is the quality of the samples obtained from the proposal distribution. Using entirely random sampling does not explore the sample space in the most uniform way, whereas quasi-MC (QMC) methods [11] (also known as low discrepancy point sets) which are deterministic in nature, lead to a more uniform exploration of the sample space. QMC methods have been shown empirically as well as theoretically to lead to a faster convergence compared to entirely random sampling. There have been several approaches presented in literature to generating QMC points [12]. In this work we use the Halton sequence which pertains to the digital nets low discrepancy point

sets family. The use of QMC methods in this work is limited to sampling from a Gaussian mixture of the form

$$\sum_{i=1}^N w_k^{(i)} p(x_k | x_{k-1}^{(i)}) \quad (28)$$

where $\{w_k^{(i)}\}_{i=1}^N$ are weights and $p(x_k | x_{k-1}^{(i)})$ is a Gaussian with mean $\mu_k^{(i)}$ and covariance matrix $\Sigma_k^{(i)}$. The exact details regarding QMC sampling of this form are given in algorithm 1. A comprehensive study on the use of QMC methods for non-linear filtering was presented in [11].

Algorithm 1: QMC sampling from a Gaussian mixture

- Draw $\{j^{(\ell)}\}_{\ell=1}^N$ from the distribution $\{w_{k-1}^{(i)}\}_{i=1}^N$.
 - **for** $i = 1, \dots, N$
 - Find the set $E_i \triangleq \{\ell | j^{(\ell)} = i\}$, and let $|E_i|$ denote the number of elements in E_i .
 - Generate $|E_i|$ QMC points $\{\mathbf{u}^{(\ell)}\}_{\ell=1}^{|E_i|}$ in $[0, 1)^D$.
 - Transform $\{\mathbf{u}^{(\ell)}\}_{\ell=1}^{|E_i|}$ to $\{\mathbf{x}^{(j)}\}_{j=|E_{i-1}|+1}^{|E_i|}$:
 - perform Cholesky decomposition on $\Sigma_k^{(i)}$, i.e., $\Sigma_k^{(i)} = R^T R$
 - transform $\{\mathbf{u}^{(\ell)}\}_{\ell=1}^{|E_i|}$ to $\{\mathbf{y}^{(\ell)}\}_{\ell=1}^{|E_i|}$, via $y_d^{(\ell)} = \phi^{-1}(u_d^{(\ell)})$, $d = 1, \dots, D$, where $\phi(x) = (1/\sqrt{2\pi}) \int_{-\infty}^x \exp(-w^2/2) dw$.
 - set $\mathbf{x}^{(|E_{i-1}|+\ell)} = \mu_k^{(i)} + R\mathbf{y}^{(\ell)}$, $\ell = 1, \dots, |E_i|$.
- end for**
-

III. NONLINEAR FILTERING

A. The Marginal Particle Filter

The nonlinear filtering approach that we use in this work follows that in [11] where IS and QMC methods are used to approximate the integrals in (3)–(5). Assuming we have a proposal density $\pi(x_k)$ from which we can sample easily, then using (27) to approximate the integrals we can replace the recursion (3)–(5) with

$$\hat{p}(x_k | z_{1:k-1}) = \sum_{i=1}^N w_{k-1}^{(i)} p(x_k | x_{k-1}^{(i)}) \quad (29)$$

$$\begin{aligned} w_k^{(i)} &\triangleq \frac{\hat{p}(x_k^{(i)} | z_{1:k})}{\pi(x_k^{(i)})} \\ &= \frac{1}{\hat{c}_k} p(z_k | x_k^{(i)}) \frac{\hat{p}(x_k^{(i)} | z_{1:k-1})}{\pi(x_k^{(i)})} \end{aligned} \quad (30)$$

$$\text{where } \hat{c}_k = \sum_{i=1}^N p(z_k | x_k^{(i)}) \frac{\hat{p}(x_k^{(i)} | z_{1:k-1})}{\pi(x_k^{(i)})} \quad (31)$$

and the state estimator can also be approximated using the IS framework

$$\hat{x}_k \triangleq \frac{1}{\hat{c}_k} \int x_k \cdot p(z_k | x_k) \hat{p}(x_k | z_{1:k-1}) dx_k \approx \sum_{i=1}^N x_k^{(i)} w_k^{(i)} \quad (32)$$

where $\hat{c}_k = \int p(z_k | x_k) \hat{p}(x_k | z_{1:k-1}) dx_k$, and the result is obtained when using IS to approximate both the integrand and \hat{c}_k .

The MPF filtering algorithm is summarized as follows.

Algorithm 2: The MPF

- **Initialization:** Sample N points $\{x_0^{(i)}\}_{i=1}^N$ from the prior distribution $p(x_0)$, and set $w_0^{(i)} = p(x_0^{(i)})$, $i = 1, \dots, N$.
 - **for** $k = 1, 2, \dots$
 - Sample N points $\{x_k^{(i)}\}_{i=1}^N$ from the proposal distribution $\pi(x_k)$.
 - Compute the predictive density $\hat{p}(x_k^{(i)} | z_{1:k-1})$, $i = 1, \dots, N$, using (29).
 - Compute the posterior weights $w_k^{(i)}$, $i = 1, \dots, N$, using (30) and (31).
 - Compute the state estimator using (32).
- end for**
-

B. Proposal Distributions

The choice of the proposal distribution has a significant affect on the performance of the filtering algorithm. Next we describe several choices for proposal distributions that we consider in this work.

1) **SIS:** The sequential IS (SIS) is the most computationally beneficial choice for a proposal distribution. If we take

$$\pi(x_k) = p(x_k | z_{1:k-1}) \quad (33)$$

it simplifies (30)

$$w_k^{(i)} \propto p(z_k | x_k^{(i)}) \quad (34)$$

thereby avoiding the computationally expensive stage of evaluating the predictive density values (29). Sampling from (33) is then performed using Algorithm 1. However this choice does not take into consideration the information given by the latest measurement z_k , and therefore may be inefficient and require the number of particles N to be very large, especially in high dimensions. Having a large number of particles N is inefficient since it leads to a large memory complexity.

2) **AMPF:** The optimal choice of a proposal distribution is the posterior distribution $\hat{p}(x_k | z_{1:k})$, however it is often hard to obtain samples from the posterior. Using the AMPF [2] it is possible to obtain samples from a proposal distribution that

is close to the posterior distribution. The posterior distribution can equivalently be written as

$$\begin{aligned}\hat{p}(x_k|z_{1:k}) &\propto p(z_k|x_k) \sum_{i=1}^N w_{k-1}^{(i)} p\left(x_k \middle| x_{k-1}^{(i)}\right) \\ &= \sum_{i=1}^N w_{k-1}^{(i)} p\left(z_k \middle| x_{k-1}^{(i)}\right) p\left(x_k \middle| x_{k-1}^{(i)}, z_k\right) \\ &= \sum_{i=1}^N p(i|z_{1:k}) p\left(x_k \middle| x_{k-1}^{(i)}, z_k\right) \quad (35)\end{aligned}$$

$$\begin{aligned}p(i|z_{1:k}) &\propto w_{k-1}^{(i)} p\left(z_k \middle| x_{k-1}^{(i)}\right) \\ &= w_{k-1}^{(i)} \int p(z_k|x_k) p\left(x_k \middle| x_{k-1}^{(i)}\right) dx_k. \quad (36)\end{aligned}$$

Using (35) we can sample from the proposal distribution $\hat{p}(x_k|z_{1:k})$ by first sampling i , which is known as the auxiliary variable from $\hat{p}(i|z_{1:k})$, and then sampling x_k from $p(x_k|x_{k-1}^{(i)}, z_k)$. Since (36) cannot usually be evaluated analytically, it was suggested in [2] to use the approximation $\hat{p}(i|z_{1:k}) \propto w_{k-1}^{(i)} p(z_k|\mu_k^{(i)})$ where $\mu_k^{(i)} = \mathbb{E}[x_k|x_{k-1}^{(i)}]$. Furthermore, since sampling from $p(x_k|x_{k-1}^{(i)}, z_k)$ is not always possible, we follow [2] and sample from $p(x_k|x_{k-1}^{(i)})$ instead, where in all the cases which we consider $p(x_k|x_{k-1}^{(i)})$ is a Gaussian distribution from which samples are easily drawn. Therefore, the AMPF proposal distribution is [2]

$$\pi(x_k|z_{1:k}) \propto \sum_{i=1}^N \hat{p}(i|z_{1:k}) p\left(x_k \middle| x_{k-1}^{(i)}\right) \quad (37)$$

and the sampling is performed using Algorithm 1.

3) *AMPF-IS*: We propose a new proposal distribution which we call AMPF-IS, where (36) is approximated using IS. For each i we draw m samples $\{\hat{x}_k^{(n)}\}_{n=1}^m$ from $p(x_k|x_{k-1}^{(i)})$ and approximate (36) using

$$\hat{p}(i|z_{1:k}) \propto w_{k-1}^{(i)} \sum_{n=1}^m p\left(z_k \middle| \hat{x}_k^{(n)}\right). \quad (38)$$

Sampling from the proposal $\pi(x_k|z_{1:k})$ proceeds similarly to the AMPF case using (37), using the approximation in (38) and using Algorithm 1.

C. Reducing the Computational Complexity of the AMPF Using the IFGT

The AMPF and AMPF-IS improve on the SIS proposal distribution since the information provided by the latest measurement z_k is used to perform sampling in areas of high probability of the posterior $\hat{p}(x_k|z_{1:k})$, however there is an added computational cost incurred by having to evaluate (29) and (37), both of which have $O(N^2)$ computational complexity. Next we show that this computational complexity can be reduced to $O(N)$ using the IFGT.

In this work, we consider the case where the state noise in (1) is a zero mean Gaussian with a covariance matrix Σ . Factorizing Σ into $\Sigma = V^T \Lambda V$ we can rewrite (29) as

$$\hat{p}(x_k|z_{1:k-1}) = \kappa \sum_{i=1}^N w_{k-1}^{(i)} e^{-\|\nu_k - s_k^{(i)}\|^2/2} \quad (39)$$

where $\nu_k = \Lambda^{1/2} V x_k$, and $s_k^{(i)} = \Lambda^{1/2} V f(x_{k-1}^{(i)})$. Therefore, (39) is equivalent to (7) where $\sigma = 1$, $q_j = w_{k-1}^{(j)}$, and ν_k and $s_k^{(i)}$ are the target and source points respectively. Thus, (39) can be approximated using the IFGT. Similarly, we can approximate (37) using the IFGT.

IV. NEW ERROR BOUNDS FOR THE IFGT

In this section, we derive new upper error bounds for the IFGT. The first bound assumes that the source points and their weights are given, whereas the second bound applies to the case where only the radius of a ball containing the source points is known.

Theorem 1: The error E_T due to truncating the series (15) after p th order satisfies the bound

$$|E_T| \leq \kappa \max_{\|\tau\|>0} e^{-\|\tau\|^2/2} \Theta_1(\|\tau\|) \quad (40)$$

where

$$\Theta_1(\|\tau\|) = \sum_{j=1}^N |q_j| e^{-\|\xi_j\|^2/2 + \|\xi_j\| \|\tau\|} - \sum_{n=0}^{p-1} d_n \|\tau\|^n, \quad (41)$$

$$d_n = \sum_{j=1}^N \frac{1}{n!} |q_j| e^{-\|\xi_j\|^2/2} \|\xi_j\|^n \quad (42)$$

and where $\xi_j = \Delta s_j / \sigma$, $\tau = \Delta t / \sigma$. Furthermore the global maximum in (40) can be found by performing a line search on $\|\tau\|$, and the upper bound decreases as the truncation order p increases.

Proof: The truncation error in (15) can be bounded by,

$$\begin{aligned}|E_T| &\leq \kappa \sum_{j=1}^N |q_j| \left| e^{-\|\xi_j\|^2/2} e^{-\|\tau\|^2/2} \sum_{n=p}^{\infty} \frac{1}{n!} (\xi_j \cdot \tau)^n \right| \\ &\leq \kappa e^{-\|\tau\|^2/2} \sum_{j=1}^N |q_j| e^{-\|\xi_j\|^2/2} \sum_{n=p}^{\infty} \frac{1}{n!} (\|\xi_j\| \|\tau\|)^n \quad (43)\end{aligned}$$

$$\begin{aligned}&= \kappa e^{-\|\tau\|^2/2} \sum_{j=1}^N |q_j| e^{-\|\xi_j\|^2/2} \\ &\quad \times \left(e^{\|\xi_j\| \|\tau\|} - \sum_{n=0}^{p-1} \frac{1}{n!} (\|\xi_j\| \|\tau\|)^n \right) \quad (44)\end{aligned}$$

$$\leq \kappa \max_{\|\tau\|>0} e^{-\|\tau\|^2/2} \Theta_1(\|\tau\|) \quad (45)$$

where (43) follows from the Cauchy–Schwartz inequality, (44) follows since the infinite sum in (43) is the tail of the Taylor series expansion of the exponent function, and (45) is obtained by maximizing over $\|\tau\|$.

Let $\Gamma_1(\|\tau\|) = e^{-\|\tau\|^2/2}\Theta_1(\|\tau\|)$. Since $e^{-\|\tau\|^2/2}$ is a decreasing function of $\|\tau\|$, and $\Theta_1(\|\tau\|)$ is an increasing function of $\|\tau\|$ (this can be easily seen by the infinite sum form (43)), then their product $\Gamma_1(\|\tau\|)$ can either be a monotonically increasing function, or monotonically decreasing function, or have a single global maximum/minimum. We note that $\Gamma_1(0) = 0$, and since

$$\begin{aligned} 0 \leq \Gamma_1(\|\tau\|) &\leq \sum_{j=1}^N |q_j| e^{-\|\tau\|^2/2} e^{\|\xi_j\|\|\tau\|} \\ &= \sum_{j=1}^N |q_j| e^{-\|\tau\|(\|\tau\|/2 - \|\xi_j\|)} \longrightarrow 0 \end{aligned} \quad (46)$$

as $\|\tau\| \rightarrow \infty$ for any bounded $\{\|\xi_j\|\}_{j=1}^N$, we have $\Gamma_1(\infty) = 0$. Therefore, $\Gamma_1(\|\tau\|)$ cannot be a monotonically increasing or a monotonically decreasing function, and since it is a positive function it must have a single global maximum. Therefore, (45) has a single global maximum which can be found by a line search maximization algorithm. Additionally since the infinite sum in (43) decreases as p increases, so does $\Theta_1(\|\tau\|)$. Therefore, the upper bound decreases as the truncation order increases. \square

In order to derive the second bound we first have to use two propositions which are proven in the Appendix.

Proposition 1: Let

$$\Gamma_2(\|\tau\|, r_0) = e^{-\|\tau\|^2/2} \Theta_2(\|\tau\|, r_0) \quad (47)$$

where

$$\Theta_2(\|\tau\|, r_0) = e^{-r_0^2/2} \left(e^{r_0 \cdot \|\tau\|} - \sum_{n=0}^{p-1} \frac{1}{n!} (r_0 \cdot \|\tau\|)^n \right) \quad (48)$$

then for a given $r_0 > 0$ the solution to

$$a = \arg \max_{\|\tau\| > 0} \Gamma_2(\|\tau\|, r_0) \quad (49)$$

satisfies $a > r_0$.

Proof: See Appendix I.

Proposition 2: If $\|\tau\| > r_0 > 0$, it follows that

$$\nabla_{r_0} \Theta_2(\|\tau\|, r_0) > 0. \quad (50)$$

Proof: See Appendix II.

Theorem 2: Let N_B sources with weights $\{q_j\}_{j=1}^{N_B}$ lie in a ball with radius $r_0\sigma$, then the error E_T due to truncating the series (15) after p th order satisfies the bound

$$|E_T| \leq Q_B \epsilon_p^{\text{IFGT}_3}(r_0) \quad (51)$$

where

$$\epsilon_p^{\text{IFGT}_3}(r_0) = \kappa \max_{\|\tau\| > 0} e^{-\|\tau\|^2/2} \Theta_2(\|\tau\|, r_0). \quad (52)$$

Furthermore the global maximum in (52) can be found by performing a line search on $\|\tau\|$, and the upper bound decreases as the truncation order p increases.

Proof: Taking $N = 1$, $\|\xi_1\| = r_0$ in Theorem 1 it can be seen that (51) is satisfied. All that is left to be shown is that $\epsilon_p^{\text{IFGT}_3}(r_0)$ in (52) is a monotonically increasing function of r_0 . Let a be defined as in (49) then using proposition 1 we have $a > r_0 > 0$. Using proposition 2, and the definition of the derivative we have

$$\lim_{\delta \rightarrow 0} \frac{1}{\delta} (\Theta_2(a, r_0 + \delta) - \Theta_2(a, r_0)) > 0. \quad (53)$$

Multiplying both sides of (53) by $e^{-a^2/2}$, we get

$$\lim_{\delta \rightarrow 0} \frac{1}{\delta} (\Gamma_2(a, r_0 + \delta) - \Gamma_2(a, r_0)) > 0. \quad (54)$$

By the definition of a and $\epsilon_p^{\text{IFGT}_3}(r_0)$ in (49) and (52), respectively, we have

$$\epsilon_p^{\text{IFGT}_3}(r_0) = \kappa \Gamma_2(a, r_0) \quad (55)$$

and

$$\epsilon_p^{\text{IFGT}_3}(r_0 + \delta) \geq \kappa \Gamma_2(a, r_0 + \delta). \quad (56)$$

Using (55) and (56) in (54), we obtain

$$\lim_{\delta \rightarrow 0} \frac{1}{\delta} (\epsilon_p^{\text{IFGT}_3}(r_0 + \delta) - \epsilon_p^{\text{IFGT}_3}(r_0)) > 0 \quad (57)$$

which is equivalent to $\nabla_{r_0} \epsilon_p^{\text{IFGT}_3}(r_0) > 0$. Therefore, $\epsilon_p^{\text{IFGT}_3}(r_0)$ is an increasing function of r_0 . \square

V. ANALYSIS OF THE NEW ERROR BOUNDS AND CHOOSING THE IFGT PARAMETERS USING THE NEW BOUNDS

In this section, we first analyse the new error bounds and demonstrate that the new error bounds a) are significantly tighter than the old bound and b) are consistent with our expectation that the error decreases as the truncation order increases. Since in practice the truncation order and the number of clusters that are used to partition the source points have to be chosen experimentally such that a compromise between the speed and accuracy of the filtering algorithm is obtained, these new bounds provide important insight into this choice. We then show how the IFGT parameters r_s , r_t , and p can be chosen using the new bounds such that the KDE can be approximated to any order of accuracy.

A. Analysis of the Error Bounds

1) Analysis of the Error Bound in Theorem 1: We evaluate the new upper error bound in Theorem 1 and the old upper error bound (22), assuming that the source points are drawn from a multivariate Gaussian PDF, and the weights are obtained by evaluating the multivariate Gaussian PDF at the source points. This is an appropriate model for the filtering case where the

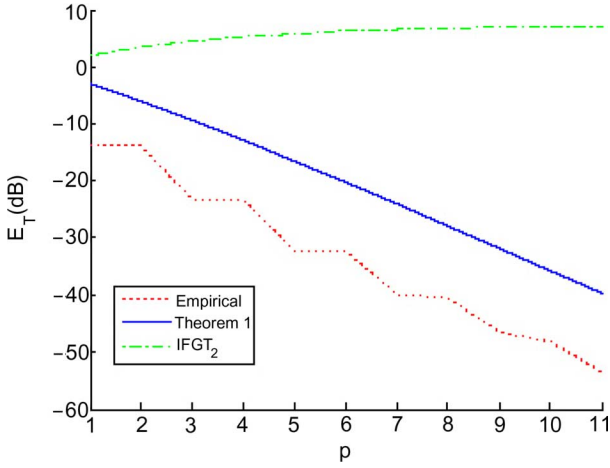


Fig. 1. Empirical upper error bound and upper error bounds (40) and (22) versus truncation order for source points generated from a zero mean Gaussian with covariance matrix $0.4 I$.

source points and weights represent a PDF that could be modeled as a Gaussian mixture where each cluster of source points originates from a different mixture. We compare the upper error bounds to the maximum error obtained experimentally when using the IFGT, where a single cluster is used for all the source points. We generated 5000 source points from a 4-D multivariate Gaussian with covariance matrix $0.4 I$, and set the weights of all the source points to the same value. For simplicity, we used $\sigma = 1$ in (7), and we assume $\kappa = 1$. We generated 5000 target points from a uniform distribution over $[-5, 5]^4$. Fig. 1 compares the upper error bounds (40) and (22), and the empirical maximum error obtained for the generated data. It can be seen that the new bound is significantly tighter compared to the old bound. Since as discussed above the experimental scenario that we used to evaluate the bound given in Theorem 1 is very similar to what is expected in a nonlinear filtering problem, Fig. 1 gives strong evidence that using small truncation orders should be sufficient to obtain satisfactory performance in nonlinear filtering problems.

2) *Analysis of the Error Bound in Theorem 2:* Fig. 2 plots $\epsilon_p^{\text{IFGT}_3}(r_0)$ in (52), and $\epsilon_p^{\text{IFGT}_2}(r_0)$ in (20) for $0 \leq r_0 \leq 3$, and for truncation orders $p = 5, 8$. It can be seen that the new bound $\epsilon_p^{\text{IFGT}_3}(r_0)$ is significantly tighter than the existing bound $\epsilon_p^{\text{IFGT}_2}(r_0)$. Furthermore it can be seen that the old bound is not consistent with our expectation that the error decreases as the truncation order increases. Theorem 2, on the other hand, stipulates that the error decreases as the truncation order increases. In the next subsection we describe our implementation of the IFGT which clusters the source points into balls with radius σr_0 . Therefore, based on Theorem 2 we can expect the error performance of the filtering algorithm to improve as the truncation order increases for any value of r_0 .

B. Choosing the IFGT Parameters Using the New Bounds

1) *Choosing Parameters That Satisfy a Specified Error Bound:* Similarly to (19), we take $r_t = r_s + n\sigma$, thus the error due to ignoring all the other clusters is bounded by $Qe^{-n^2/2}$.

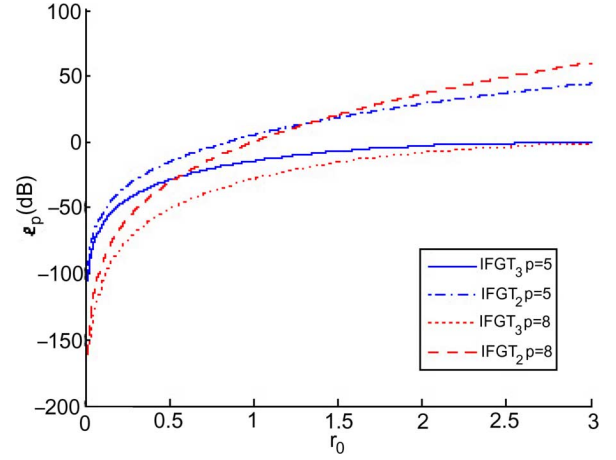


Fig. 2. ϵ_p for the IFGT₂ and IFGT₃ versus r_0 , for $\kappa = 1$, for truncation orders 5, and 8.

Taking $r_s = r_0\sigma$, the error due to truncating the expansion after the p th order is bounded by $Q\epsilon_p^{\text{IFGT}_3}(r_0)$. Therefore, by fixing r_0 and p according to plots such as Fig. 2, we can bound the maximum truncation error. Similarly we can choose n such that $Qe^{-n^2/2}$ is less than $Q\epsilon_p^{\text{IFGT}_3}(r_0)$. Therefore, we define the new IFGT parameters to be r_0 , n , and p . The specific details of the IFGT using this choice of parameters is summarized in Algorithm 3. We may obtain an even tighter bound than $Q\epsilon_p^{\text{IFGT}_3}(r_0)$ for the truncation error by applying the bound given in Theorem 1 for each cluster and summing the error bounds over all the clusters.

2) *Choosing the Parameters Experimentally:* As is evident from the previous discussion about choosing the parameters that satisfy a specified error bound, the parameters that determine the performance of the IFGT are n , r_0 , and p . Since n is independent of r_0 and p it can be specified separately, whereas it can be seen from Fig. 2 that r_0 and p are dependent. Furthermore the larger p is, and the smaller r_0 is the more accurate the IFGT becomes. The process of choosing r_0 and p experimentally therefore requires incrementing p and decreasing r_0 until a satisfactory performance of the filtering algorithm is obtained. It should be noted that as p increases and r_0 decreases the computational constant of the IFGT also increases therefore the filtering algorithm becomes slower.

Algorithm 3: IFGT

- Cluster all the source points into balls with radius $r_0\sigma$.
 - Compute the IFGT coefficients for each cluster using (16).
 - For each target point, sum over all evaluations of (15) at all the clusters with the centers that lie within $(n + r_0)\sigma$ from the target point.
-

VI. EXPERIMENTAL RESULTS

In this section, we show experimental results for the new framework to nonlinear filtering. We also compare our results to using the SIS filter discussed previously.

TABLE I
TIME [sec], AND RMSE WHEN USING THE AMPF AND AMPF-IS PROPOSAL DISTRIBUTIONS
WITH DIRECT EVALUATION AND IFGT, AND WHEN USING THE SIS PARTICLE FILTER

Algorithm	Direct		IFGT		N	m
	RMSE	time (sec)	RMSE	time (sec)		
AMPF	6.3	46.2	6.4	3.7	1000	1
AMPF-IS	5.9	12.3	5.9	2	500	10
AMPF-IS	5.5	12.6	5.5	2.8	500	50
AMPF-IS	5.6	14.5	5.5	3.8	500	100
AMPF-IS	5.6	46.6	5.7	4.3	1000	10
AMPF-IS	5.2	49.2	5.3	6	1000	50
AMPF-IS	5.1	51.4	5.2	8.2	1000	100
SIS	6	1.9			5000	
SIS	5.6	3.8			10000	
SIS	5.5	5			15000	
SIS	5.4	8.2			20000	

TABLE II
TIME [sec], AND RMSE FOR DIFFERENT VALUES OF r_0 AND p WHEN
USING THE AMPF-IS PROPOSAL DISTRIBUTION WITH THE IFGT

$r_0 \setminus p$	1		3		5		7	
	RMSE	time (sec)	RMSE	time (sec)	RMSE	time (sec)	RMSE	time (sec)
1	5.5	4.8	5.4	5.7	5.3	9.4	5.3	19.2
2	5.6	4.5	5.4	5.2	5.4	8	5.3	15.5
3	5.6	3.9	5.5	4.4	5.4	6.4	5.4	11.8
4	5.8	3.4	5.6	3.9	5.5	5.5	5.5	9.2
5	5.9	3.2	5.7	3.3	5.7	4.3	5.6	6.4

A. Example 1

For our simulations we use the a four dimensional state space model which is an extension of the one dimensional state space model used in [8]

$$x_{t+1}^{(d)} = \frac{x_t^{(u_1(d))}}{2} + 25 \frac{x_t^{(u_2(d))}}{1 + \left(x_t^{(u_2(d))}\right)^2} + 8 \cos(1.2t) + w_t^{(d)} \quad (58)$$

$$y_t^{(d)} = \frac{\left(x_t^{(d)}\right)^2}{20} + v_t^{(d)} \quad (59)$$

where $d = 1 \dots 4$, $u_1 = [2, 4, 1, 3]$, $u_2 = [3, 4, 1, 2]$, $w_t^{(d)} \sim N(0, 10)$, $v_t^{(d)} \sim N(0, 1)$, and $x_0^{(d)} \sim N(0, 5)$. Each experiment had 200 time steps, and the results were averaged over 100 different experiments. We used $n = 4$, $r_0 = 3$, and $p = 3$ in the IFGT. In Table I, we compare the root mean square error (RMSE) and timing results for the different algorithms when using the IFGT and when using direct evaluation of (37), (39). In the upper group of algorithms, we compare the results when using the MPF with the AMPF proposal distribution and with the AMPF-IS proposal distribution that we developed in this work. In the bottom of the table, we show the results obtained when using the SIS particle filter. It can be seen that the new AMPF-IS proposal distribution significantly improves the RMSE over the AMPF proposal distribution, and that increasing the number of samples m used in the AMPF-IS proposal distribution, decreases the RMSE. Furthermore the use of the IFGT

instead of the direct evaluation speeds up the execution time significantly while increasing the error only slightly. The timing and RMSE results using the AMPF-IS with the IFGT are comparable to the results obtained using the SIS particle filter, however the number of samples in the SIS particle filter is significantly larger.

In Table II, we show the RMSE and timing results obtained for different values of r_0 and truncation orders p for the same nonlinear filtering example using $n = 4$, $m_0 = 50$ and $N = 1000$. Each experiment had 200 time steps, and the results were averaged over 100 different experiments. It can be seen that the results verify what is predicted by the new bounds: a) the error increases as the value of r_0 increases; b) the error decreases as the value of p increases; and c) satisfactory RMSE performance can be obtained for low truncation orders. Tables such as Table II can be used to find the optimal compromise between accuracy and speed of the filtering algorithm.

B. Example 2

In this section, we apply the new nonlinear filtering framework to the task of bearings only tracking of a single target in a sensor network [11], [16]. The state vector \mathbf{x}_k includes the position of the target and its velocity on the $x - y$ plane, and takes the form

$$\mathbf{x}_{k+1} = \begin{pmatrix} I_2 & \Delta t I_2 \\ 0 & I_2 \end{pmatrix} \mathbf{x}_k + \begin{pmatrix} \frac{(\Delta t)^2}{2} I_2 \\ \Delta t I_2 \end{pmatrix} w_k \quad (60)$$

where Δt is the time difference between two consecutive measurements, I_2 denotes the 2×2 identity matrix, and $w_k \sim$

TABLE III
TIME [sec], AND RMSE WHEN USING THE AMPF AND AMPF-IS PROPOSAL DISTRIBUTIONS
WITH DIRECT EVALUATION AND IFGT, AND WHEN USING THE SIS PARTICLE FILTER

Algorithm	Direct					IFGT					N	m
	RMSE				time (sec)	RMSE				time (sec)		
	x_k	y_k	\dot{x}_k	\dot{y}_k		x_k	y_k	\dot{x}_k	\dot{y}_k			
AMPF	12.01	11.65	1.16	1.27	14.45	19.94	12.36	1.28	1.32	2.91	600	1
AMPF-IS	3.09	3.37	0.93	1	14.68	2.73	3.3	0.88	0.98	3.63	600	20
AMPF-IS	2.27	2.92	0.88	0.97	15.23	2.67	3.38	0.89	0.99	4.43	600	50
AMPF	3.33	4.4	0.95	1.06	39.34	8.74	8.47	1.09	1.17	5.26	1000	1
AMPF-IS	2.25	2.84	0.88	0.96	39.95	2.08	2.83	0.86	0.95	6.73	1000	20
AMPF-IS	2.14	2.82	0.87	0.95	40.81	2.1	2.81	0.87	0.95	8.35	1000	50
SIS	11.21	10.95	1.18	1.22	0.23						500	
SIS	3.56	3.35	0.93	0.99	0.35						1000	
SIS	2.15	2.78	0.86	0.96	0.55						2000	
SIS	2.07	2.62	0.85	0.93	1.08						4000	

$N(0, \Sigma)$. The measurement equation which we use follows the model used in [13]

$$z_k \sim WC\left(\arctan\left(\frac{y_k - \bar{y}}{x_k - \bar{x}}\right), \rho\right) \quad (61)$$

where (\bar{x}, \bar{y}) denotes the position of the sensor, and where $WC(\mu, \rho)$, denotes the two parameter wrapped Cauchy distribution [15], that takes the form

$$f(z_k|\mu) = \frac{1}{2\pi} \frac{1 - \rho^2}{1 + \rho^2 - 2\rho \cos(z_k - \mu)} \quad (62)$$

where $0 \leq \rho \leq 1$, $-\pi < z_k, \mu < \pi$.

Each sensor has to choose the next leader sensor to perform the filtering once it has performed the measurement update. We use the scheme that was used in [11] where the new leader ℓ^* is chosen using the following decision rule:

$$\ell^* = \arg \min_{i \in \Omega(\ell)} [(\bar{x}_i - x_{k+1|k})^2 + (\bar{y}_i - y_{k+1|k})^2] \quad (63)$$

where $\Omega(\ell)$ denotes the set of sensors that sensor i can communicate with, (\bar{x}_i, \bar{y}_i) is the coordinate of sensor i , and $(x_{k+1|k}, y_{k+1|k})$ is the one step ahead prediction made by the previous leader for the location of the target. Once the new leader sensor is chosen, the previous leader sensor transmits its state estimate $\hat{p}(x_k|z_{1:k})$ to the new leader sensor. For the experiments presented here, we used $\Delta t = 1$ and $\Sigma = 0.5I_2$, where the track with 200 time steps was generated once and used for all the experiments. The measurements were generated using $\rho = 1 - \sigma^2$, where we used $\sigma = 0.02$ to generate the measurements, and $\sigma = 0.1$ to evaluate $p(z_k|x_k^{(i)})$ in (30) since this was necessary in order to avoid divergence from the real track for all the tested algorithms. The initial state estimate that we used was distributed as $N([x_0; y_0]^T, 100I_2)$. For approximating the AMPF-IS integral using (38), we use $\sigma = 0.02$. We used $n = 4$, $r_0 = 3$, and $p = 5$ in the IFGT. The timing and RMSE results were averaged over 100 different runs. The sensor network included 200 sensors. Fig. 3 shows the sensors, the target's track, and its estimate for a single experiment using the AMPF-IS and the IFGT algorithm.

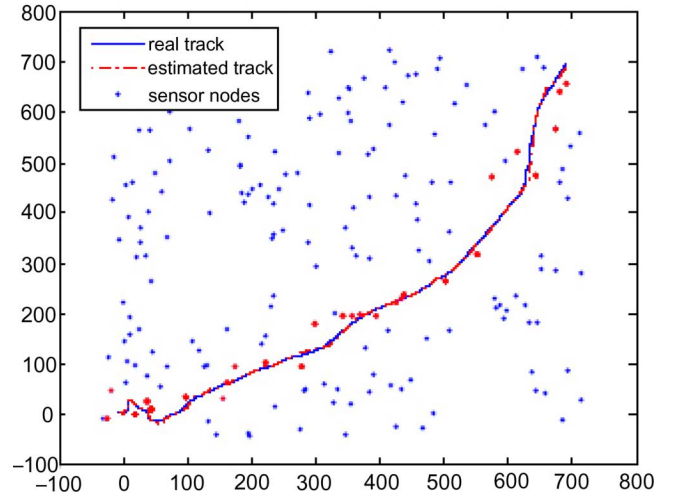


Fig. 3. Sensor network, the real and the estimated tracks.

TABLE IV
TIME [sec], AND RMSE WHEN USING THE AMPF-IS PROPOSAL DISTRIBUTION
AND IFGT FOR THE EXAMPLE IN SECTION VI-A FOR DIFFERENT
VALUES OF r_0 AND K

r_0	RMSE	Time (sec)	K	RMSE	Time (sec)
3	5.6	4.2	10	5.5	4.2
2	5.35	5	15	5.36	5
1	5.35	6	20	5.35	5.8

Table III shows the RMSE and timing results for the tracking experiments. It can be seen that using the AMPF-IS and the IFGT we can obtain similar RMSE results to the SIS particle filter using fewer particles, however the SIS filter is faster. It should be noted that since in such a sensor network application the particles have to be transmitted between the different sensors, energy conservation requirements mandate using as few samples as possible. Thus, using fewer samples may be more important than the computation time. It can also be seen that the IFGT improves the computation time over the direct evaluation, and the use of the AMPF-IS proposal distribution significantly improves the RMSE results over using the AMPF proposal distribution.

TABLE V
TIME [sec], AND RMSE WHEN USING THE AMPF-IS PROPOSAL DISTRIBUTION AND IFGT FOR THE
EXAMPLE IN SECTION VI-B FOR DIFFERENT VALUES OF r_0 AND K

r_0	RMSE				Time (sec)	K	RMSE				Time (sec)
	x_k	y_k	\dot{x}_k	\dot{y}_k			x_k	y_k	\dot{x}_k	\dot{y}_k	
3	2.26	2.84	0.87	0.96	8.5	140	2.37	2.9	0.88	0.96	11.6
2	2.28	2.81	0.86	0.95	11.4	170	3.2	4.5	0.9	1	13.5
1	2.42	2.94	0.88	0.97	14.94	200	2.35	2.86	0.88	0.96	15.38

C. Choosing r_0 versus Choosing the Number of Clusters

As was discussed in Section V-B, the new bounds suggest that one has to choose r_0 such that the radius of the balls that are used to cluster the data is $r_0\sigma$. This is as opposed to the approach taken in [8] where as was discussed in Section II-A, one had to specify the number of clusters instead. In order to compare these two approaches for the nonlinear filtering case, we show the results of the two nonlinear filtering examples in Sections VI-A and VI-B for different choices of r_0 , and K in Tables IV and V, respectively. The minimal value for K was chosen such that none of the experiments diverged.

It can be seen that similar results can be obtained when fixing the right parameters for r_0 or K ; however, the values for r_0 remain the same for the two filtering examples, whereas the values for K change significantly between the two examples. This indicates that the formulation that is proposed here may have a significant advantage for the nonlinear filtering case as it requires less effort when choosing the IFGT parameters.

VII. CONCLUSION

We considered the use of the IFGT to perform fast KDE for evaluating the Chapman Kolmogorov equation in nonlinear filtering. We presented new upper error bounds for the IFGT which are significantly tighter than the existing bound. We related the truncation error to the distance of a source point from the center of the expansion and to the order of truncation for any target point analytically, and we showed that this error increases as the radius increases, and decreases as the truncation order increases. Since in practice the IFGT parameters have to be fixed experimentally, the new bounds facilitate the choice of parameters. We applied the IFGT to nonlinear filtering using the new AMPF-IS proposal distribution that uses IS to approximate not only the integrals in Bayes recursion equations, but also to approximate the integral in the AMPF. The experimental results using the new framework verify its effectiveness and show that it can be used to reduce the number of particles that have to be used. We also examined the effect that the IFGT parameters have on the nonlinear filtering performance, and confirmed the predictions of the new bounds.

APPENDIX I

PROOF OF PROPOSITION 1

Taking the derivative of $\Gamma_2(\|\tau\|, r_0)$ with respect to $\|\tau\|$ and evaluating it at r_0 we obtain

$$\left. \frac{\partial}{\partial \|\tau\|} (\Gamma_2(\|\tau\|, r_0)) \right|_{\|\tau\|=r_0} = \frac{r_0^{2p-1}}{(p-1)!} e^{-r_0^2} > 0 \quad (64)$$

for any $r_0 > 0$. Therefore, all that there is left to show is that $\Gamma_2(\|\tau\|, r_0)$ has a single maximum solution, since then $(\partial/\partial \|\tau\|)(\Gamma_2(\|\tau\|, r_0)) = 0$ only if $\|\tau\| > r_0$. Similarly to the proof of Theorem 1, $\Theta_2(\|\tau\|, r_0)$ is an increasing function of $\|\tau\|$ since it is the tail of a the Taylor series expansion of the exponent function, and $e^{-\|\tau\|^2/2}$ is a decreasing function of $\|\tau\|$. Since $\Gamma_2(0, r_0) = 0$,

$$0 < \Gamma_2(\|\tau\|, r_0) \leq e^{-\|\tau\|^2/2} e^{r_0\|\tau\|} = e^{-\|\tau\|(\|\tau\|/2 - r_0)} \rightarrow 0 \quad (65)$$

as $\|\tau\| \rightarrow \infty$ for any bounded r_0 , and since $\Gamma_2(\|\tau\|, r_0)$ is a positive function it must have a single maximum.

APPENDIX II

PROOF OF PROPOSITION 2

Taking the derivative of $\Theta_2(\|\tau\|, r_0)$ with respect to r_0 we obtain

$$\begin{aligned} & \nabla_{r_0} (\Theta_2(\|\tau\|, r_0)) \\ &= \left((\|\tau\| - r_0) \sum_{n=p}^{\infty} \frac{1}{n!} (r_0\|\tau\|)^n + \frac{\|\tau\|^{p-1}}{(p-1)!} \right) e^{-r_0^2/2} \quad (66) \end{aligned}$$

which is strictly positive for every $\|\tau\| > r_0 > 0$.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for valuable comments that improved the presentation of this paper.

REFERENCES

- [1] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.
- [2] M. Klaas, N. de Freitas, and A. Doucet, "Toward practical N^2 Monte Carlo: The marginal particle filter," presented at the 21st Conf. Uncertainty Artificial Intelligence, Edinburgh, Scotland, 2005.
- [3] A. Elgammal, R. Duraiswami, and L. S. Davis, "Efficient kernel density estimation using the fast gauss transform with applications to color modeling and tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 11, pp. 1499–1504, Nov. 2003.
- [4] K. Deng and A. W. Moore, "Multiresolution instance-based learning," in *Proc. 12th Int. Joint Conf. Artificial Intelligence*, 1995, pp. 1233–1239.
- [5] L. Greengard and J. Strain, "The fast gauss transform," *SIAM J. Scientific Comput.*, vol. 12, no. 1, pp. 79–94, 1991.
- [6] B. J. C. Baxter and G. Roussos, "A new error estimate of the fast gauss transform," *SIAM J. Scientific Comput.*, vol. 24, no. 1, pp. 257–259, 2002.
- [7] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis, "Improved fast gauss transform and efficient kernel density estimation," in *Proc. IEEE Int. Conf. Computer Vision*, 2003, pp. 464–471.

- [8] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [9] A. G. Gray and A. W. Moore, "Rapid evaluation of multiple density models," in *Proc. 9th Int. Workshop Artificial Intelligence Statistics*, C. M. Bishop and B. J. Frey, Eds., Key West, FL, Jan. 3–6, 2003.
- [10] C. P. Robert and G. Casella, "Monte Carlo integration," in *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 2004.
- [11] D. Guo and X. Wang, "Quasi-Monte Carlo filtering in nonlinear dynamic systems," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2087–2098, Jun. 2006.
- [12] "Random and quasi-random point sets," in *Lecture Notes in Statistics*, P. Hellekalek and G. Larcher, Eds. New York: Springer-Verlag, 1998.
- [13] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. Amer. Stat. Assoc.*, vol. 94, no. 446, pp. 590–599, 1999.
- [14] D. Lang, M. Klaas, and N. de Freitas, "Empirical testing of fast kernel density estimation algorithms," University of British Columbia, BC, Canada, TR-2005-03, 2005.
- [15] N. I. Fisher, *Statistical Analysis of Circular Data*. Cambridge, U.K.: Cambridge Univ. Press, 1993.
- [16] J. H. Kotecha and P. M. Djurić, "Gaussian particle filtering," *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.
- [17] V. C. Raykar, C. Yang, R. Duraiswami, and N. Gumerov, "Fast computation of sums of Gaussians in high dimensions," Computer Sci. Dept., Univ. of Maryland, College Park, MD, CS-TR-4767, 2006.
- [18] R. Mittelman and E. L. Miller, "Fast Gauss transforms based on a high order singular value decomposition for nonlinear filtering," in *Proc. IEEE Statistical Signal Processing Workshop*, 2007, pp. 94–98.
- [19] F. Campillo and V. Rossi, "Convolution particle filtering for parameter estimation in general state-space models," presented at the 45th IEEE Conf. Decision Control, San Diego, CA, Dec. 2006.
- [20] C. Chang and R. Ansari, "Kernel particle filter for visual tracking," *IEEE Signal Process. Lett.*, vol. 12, no. 3, pp. 242–245, Mar. 2005.
- [21] A. Doucet, N. de Freitas, and N. J. Gordon, *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.
- [22] A. T. Ihler, J. W. Fisher, and A. S. Willsky, "Particle filtering under communications constraints," in *Proc. IEEE Statistical Signal Processing Workshop*, 2005, pp. 89–94.



Roni Mittelman (S'08) received the B.S. and M.S. (*cum laude*) degrees from the Technion-Israel Institute of Technology, Haifa, Israel, both in electrical engineering. He is currently working towards the Ph.D. degree in electrical and computer engineering at Northeastern University, Boston, MA.

His research interests include statistical signal processing, machine learning, computer vision, and sensor networks.



Eric L. Miller (S'90–M'95–SM'03) received the S.B., S.M., and Ph.D. degrees, all in electrical engineering and computer science, from the Massachusetts Institute of Technology, Cambridge, in 1990, 1992, and 1994, respectively.

He is currently a Professor in the Department of Electrical and Computer Engineering at Tufts University, Medford, MA. His research interests include physics-based tomographic image formation and object characterization, inverse problems in general and inverse scattering in particular, regularization, statistical signal and imaging processing, and computational physical modeling. This work has been carried out in the context of applications, including medical imaging, nondestructive evaluation, environmental monitoring and remediation, landmine and unexploded ordnance remediation, and automatic target detection and classification.

Dr. Miller is a member of Tau Beta Pi, Phi Beta Kappa, and Eta Kappa Nu. He received the CAREER Award from the National Science Foundation in 1996 and the Outstanding Research Award from the College of Engineering at Northeastern University, Boston, MA, in 2002. He is currently serving as an Associate editor for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING and was in the same position at the IEEE TRANSACTIONS ON IMAGE PROCESSING from 1998 to 2002. He was the Co-General Chair of the 2008 IEEE International Geoscience and Remote Sensing Symposium held in Boston, MA.