Parallel Recursive Bayesian Estimation on Multicore Computational Platforms Using Orthogonal Basis Functions

Olov Rosén and Alexander Medvedev

***Abstract*— A method to solve the recursive Bayesian estimation problem by making use of orthogonal series expansions of the involved probability density functions is presented. The coefficients of the expansion for the posterior density are then calculated recursively via prediction and update equations. The method has two main benefits: it provides high estimation accuracy at a relatively low computational cost and it is highly amenable to parallel implementation. An application to a bearings-only tracking problem shows that the proposed method performs with the same accuracy as the particle filter but at a 24 times lower computational cost. A parallel implementation on a shared-memory multicore machine demonstrates that linear speedup in the number of cores is achievable.**

I. Introduction

Nonlinear non-Gaussian filtering problems arise in numerous signal processing and control applications such as communication, radar and sonar target tracking, and satellite navigation, to mention a few.

Consider a nonlinear discrete-time system

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + \mathbf{w}_k, \quad (1)$$

$$\mathbf{y}_k = g(\mathbf{x}_k) + \mathbf{v}_k, \quad (2)$$

with the state vector $\mathbf{x}_k \in \mathbb{R}^n$, the output vector $\mathbf{y}_k \in \mathbb{R}^p$ and the process and measurement noise $\mathbf{w}_k \in \mathbb{R}^n$, $\mathbf{v}_k \in \mathbb{R}^p$, respectively. The probability density functions (PDFs) $p(\mathbf{w}_k)$, $p(\mathbf{v}_k)$ are assumed to be *known* but are allowed to have arbitrary form. A more general formulation would as well include an deterministic input signal $\mathbf{u}_k$ but it is omitted here for brevity. A corresponding generalization of the formulae is straightforward.

The Recursive Bayesian Estimation (RBE) problem, see e.g. [1], [2], is the general underlying problem for nonlinear state estimation in stochastic dynamical systems. In RBE, the aim is to recursively obtain the PDF $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ subject to system model (1)-(2) and given the measurements

$$\mathbf{y}_{1:k} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_k\}. \quad (3)$$

For linear systems with white Gaussian process and measurement noise, an exact solution of RBE is given by the Kalman filter [3]. For more general system structures, no analytical closed-form solution is generally available and approximative approaches have to be resorted to. The most commonly used methods fall into three main categories. Grid-based methods evaluate the involved functions over a set of discrete grid points [4]. Numerical integration methods, such as e.g. Gaussian quadrature, exploit numerical integration to calculate the involved integrals [5]. Monte-Carlo methods, exemplified by the extensively studied particle filter techniques [6], use Monte-Carlo simulation to obtain a sample from the distribution of which the desired information can be extracted. For a survey of these methods see e.g. [7],[8].

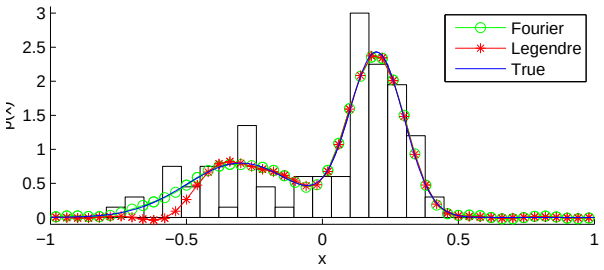

Fig. 1. Fourier, Legendre and histogram approximations of the PDF $p(x)$. The series expansions use the 10 first basis functions, and the histogram is based on 100 observations.

This paper investigates a solution to the RBE problem by expanding the PDFs in question in series of orthogonal functions. Orthogonal bases have been for a long time used in statistics to estimate PDFs with general distributions [9],[10],[11],[12]. Using orthogonal expansions, one expects that it should be possible to estimate the PDFs with a substantially lower number of variables than e.g. in the particle filter or grid-based methods since the latter two methods extract their estimates from, loosely speaking, a histogram. Fig. 1 shows a PDF approximated by a histogram based on a sample of size 100, compared to approximations by means of the first ten Legendre and Fourier basis functions. Even though the histogram uses 10 *times* more parameters, the estimation accuracy of the series approximations is still superior. Since much fewer variables are required to approximate the PDF, a smaller computational load for the same estimation accuracy can be expected when using orthogonal basis expansions.

Known approaches to Bayesian estimation with orthogonal basis functions employ e.g. Fourier series [13] and wavelets [14]. Fourier basis functions definitely present a suitable alternative. However, the compact support of the wavelets, as further discussed in Sec. VI-C, poses problems in some applications by forcing the number of terms required in the

The authors are partially supported by the project "Computationally Demanding Real-Time Applications on Multicore Platforms" funded by Swedish Foundation for Strategic Research.

O. Rosén and A. Medvedev are with the Department of Information Technology, Uppsala University, Uppsala SE-751 05, Sweden {olov.rosen, alexander.medvedev}@it.uu.se

expansion to be relatively high. This effectively undermines the low computational price of the method that is one of its main benefits.

The development in the present paper is with respect to a *general* orthogonal basis and is particularly focused on the favorable parallelization properties of the method exemplified by an implementation on a shared memory multicore processor. No particular orthogonal basis is thus assumed and the choice of a suitable basis clearly depends on the PDFs in hand. However, the method yields an estimation algorithm with good parallelization properties, no matter what particular basis is used.

Since all high-performance and much of embedded hardware is nowadays based on parallel processing, parallelizability of algorithms is of outmost importance. The orthogonality of the basis functions helps to structure the computational workload into independent segments, a property that is sought for in parallel (or distributed) implementation of algorithms. This is in contrast to e.g. Gaussian sum filters [15], [7], [16], [5], where Gaussian density functions are used as basis functions. The Gaussian density functions do not enjoy the orthogonality property and hence yield estimation algorithms that cannot be parallelized efficiently because of the data dependencies in the calculations. On the contrary, efficient parallelizations of the Kalman filter and the particle filter on multicore platforms can be found in [17], [18].

To recapitulate the highlights of the proposed solution to the RBE problem: the use of orthogonal expansions yields a method that is relatively computationally cheap and can be very efficiently implemented on a parallel architecture.

The paper structure is as follows. In Sec. II, the notation and assumptions made are summarized. The RBE problem is briefly reviewed in Sec. III. In Sec. IV, the proposed method is presented followed in Sec. VI by the practical issues that have to be dealt with in an implementation. An application to a bearings-only tracking problem as well as a speedup evaluation for a parallel implementation are given in Sec. VII. A discussion of the results is presented in Sec. VIII, followed by the conclusions.

II. Notation and Assumptions

Let $\mathbb{R}$ and $\mathbb{N}$ be the set of real and natural numbers, respectively. Vectors are written in bold lower case and matrices are in bold upper case. The inner product over a domain $\mathbf{D}$ is

$$< f(\mathbf{x}), g(\mathbf{x}) >= \int_{\mathbf{D}} f(\mathbf{x})\overline{g(\mathbf{x})}d\mathbf{x}, \tag{4}$$

while $\times$ denotes the Cartesian product. Concatenation of two vectors $\mathbf{n} = (n_1, n_2, ..., n_d)$ and $\mathbf{m} = (m_1, m_2, ..., m_d)$ is written as

$$\mathbf{nm} \quad = \quad (n_1, n_2, ..., n_d, m_1, m_2, ..., m_d).$$

All series are assumed to be absolute convergent.

A. Orthogonal functions

Let $\mathbf{T}_i = [a_i, b_i] \subseteq \mathbb{R}$, $a_i < b_i$, $i = 1, 2, ..., d$ and

$$\mathbf{T}^d = \mathbf{T}_1 \times \mathbf{T}_2 \times \cdots \times \mathbf{T}_d.$$

Introduce an orthogonal basis $\mathbf{\Phi} = \{\phi_n^{(i)}\}_{n=0}^{\infty}$ on $\mathbf{T}_i$ with respect to the inner product in (4). A multivariate orthogonal basis on $\mathbf{T}^d$ is given by $\{\phi_{\mathbf{n}}(\mathbf{x})\}_{\mathbf{n}\in\mathbb{N}^d}$, where $\phi_{\mathbf{n}}(\mathbf{x})$ is interpreted as

$$\phi_{\mathbf{n}}(\mathbf{x}) = \prod_{i=1}^{d} \phi_{n_i}(x_i)$$

for the multi-index $\mathbf{n} = (n_1, n_2, ..., n_d) \in \mathbb{N}^d$. From L_2 theory, any square integrable function $f(\cdot)$ on $\mathbf{T}^d$ can be expressed as

$$f(\mathbf{x}) = \sum_{\mathbf{n}\in\mathbb{N}^d} c_{\mathbf{n}}\phi_{\mathbf{n}}(\mathbf{x}),$$

where $c_{\mathbf{n}} =< f(\mathbf{x}), \phi_{\mathbf{n}}(\mathbf{x}) >$. The product of the $\mathbf{n}$-th and $\mathbf{m}$-th basis function will be assumed to have the expansion

$$\phi_{\mathbf{n}}(\mathbf{x})\phi_{\mathbf{m}}(\mathbf{x}) = \sum_{\mathbf{k}\in\mathbb{N}^d} g_{\mathbf{nmk}}\phi_{\mathbf{k}}(\mathbf{x}). \tag{5}$$

III. Recursive Bayesian estimation

Only a brief summary of RBE for the optimal filtering problem is provided in this section , see e.g. [4], [19],[16],[7] for a more extensive exposition. RBE seeks to recursively obtain the PDF $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ that specifies the conditional probability density for the state $\mathbf{x}_k$ of the system (1), (2) at time step k, given the measurements $\mathbf{y}_{1:k}$.

An iteration of the algorithm is comprized of two steps: prediction and update. Assume that $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ is known. The predicted PDF $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ is obtained by the Kolmogorov-Chapman equation

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int_{\mathbb{R}^d} p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1}. \tag{6}$$

When the measurement $\mathbf{y}_k$ is acquired, the predicted PDF $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ is updated from Bayes' rule to give

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}. \tag{7}$$

In (6) and (7), $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ and $p(\mathbf{y}_n|\mathbf{x}_n)$ are implicitly given by the system model. With $\delta(\cdot)$ denoting the Dirac delta, they can be obtained, via the generalized convolution formula as

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_{k-1}) &= \int_{\mathbb{R}^d} \delta(\mathbf{x}_k - f(\mathbf{x}_{k-1}) - \mathbf{w}_k)p_w(\mathbf{w}_k)d\mathbf{w}_k, \\ &= p_w(\mathbf{x}_k - f(\mathbf{x}_{k-1})) \\ p(\mathbf{y}_k|\mathbf{x}_k) &= \int_{\mathbb{R}^d} p(\mathbf{y}_k - g(\mathbf{x}_k) - \mathbf{v}_k)p_v(\mathbf{v}_k)d\mathbf{v}_k \\ &= p_v(\mathbf{y}_k - g(\mathbf{x}_k)). \end{aligned}$$

In (7), $p(\mathbf{y}_k|\mathbf{y}_{1:k-1})$ is a normalizing constant and does not need to be explicitly evaluated in a practical implementation. Given an initial state $\mathbf{x}_0$, an a priori PDF $p(\mathbf{x}_0)$ and the measurements $\mathbf{y}_{1:k}$, $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ can be obtained by applying (6) and (7) recursively.

IV. Solving the RBE via series expansions

As there is no closed-form analytical solution for the general case of RBE numerical methods have to be employed to find an estimate of the state in (1), (2). The idea advocated here is to approximate the involved PDFs by orthogonal series expansions whose coefficients can be recursively determined from the prediction and update equations in (6)-(7).

Assume that $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, $p(\mathbf{y}_k|\mathbf{x}_k)$ and $p(\mathbf{x}_{k-1}|\mathbf{y}_{k-1})$ are given by the expansions

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \sum_{\mathbf{n}\in\mathbb{N}^d}\sum_{\mathbf{m}\in\mathbb{N}^d} a_{\mathbf{nm}}\phi_{\mathbf{n}}(\mathbf{x}_k)\phi_{\mathbf{m}}(\mathbf{x}_{k-1}), \quad (8)$$

$$p(\mathbf{y}_k|\mathbf{x}_k) = \sum_{\mathbf{n}\in\mathbb{N}^d}\sum_{\mathbf{m}\in\mathbb{N}^d} b_{\mathbf{nm}}\phi_{\mathbf{n}}(\mathbf{y}_k)\phi_{\mathbf{m}}(\mathbf{x}_k), \quad (9)$$

$$p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) = \sum_{\mathbf{n}\in\mathbb{N}^d} c_{\mathbf{n}}\phi_{\mathbf{n}}(\mathbf{x}_{k-1}). \quad (10)$$

Inserting (8)-(10) into the prediction and update equations (6) and (7) yields the following relationships:

Prediction step:

$$\begin{aligned}
p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) &= \int_{\mathbb{R}^d} p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1} \\
&= \int_{\mathbb{R}^d} [\sum_{\mathbf{n}\in\mathbb{N}^d}\sum_{\mathbf{m}\in\mathbb{N}^d} a_{\mathbf{nm}}\phi_{\mathbf{n}}(\mathbf{x}_k)\phi_{\mathbf{m}}(\mathbf{x}_{k-1}) \times \\
&\qquad \sum_{\mathbf{k}\in\mathbb{N}^d} c_{\mathbf{k}}\phi_{\mathbf{k}}(\mathbf{x}_{k-1})]d\mathbf{x}_{k-1} \\
&= \sum_{\mathbf{n}\in\mathbb{N}^d}\sum_{\mathbf{m}\in\mathbb{N}^d}\sum_{\mathbf{k}\in\mathbb{N}^d} a_{\mathbf{nm}}c_{\mathbf{k}}\phi_{\mathbf{n}}(\mathbf{x}_k) \times \\
&\qquad \int_{\mathbb{R}^d} \phi_{\mathbf{m}}(\mathbf{x}_{k-1})\phi_{\mathbf{k}}(\mathbf{x}_{k-1})d\mathbf{x}_{k-1} \\
&= \sum_{\mathbf{n}\in\mathbb{N}^d} e_{\mathbf{n}}\phi_{\mathbf{n}}(\mathbf{x}_k),
\end{aligned}$$

with

$$e_{\mathbf{n}} = \sum_{\mathbf{m}\in\mathbb{N}^d}\sum_{\mathbf{k}\in\mathbb{N}^d} a_{\mathbf{n},\mathbf{m}}c_{\mathbf{k}}\cdot I_{\mathbf{mk}}, \quad (11)$$

where

$$I_{\mathbf{mk}} = \int_{\mathbb{R}^d} \phi_{\mathbf{m}}(\mathbf{x}_{k-1})\phi_{\mathbf{k}}(\mathbf{x}_{k-1})d\mathbf{x}_{k-1}.$$

For real-valued basis functions, the integral above is equivalent to the inner product defined in (4) and hence

$$I_{\mathbf{mk}} = \begin{cases} 1 & if\ \mathbf{m}=\mathbf{k}, \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

For complex-valued basis functions, the integral is not equal to inner product (4). However, for the latter case, it holds that $\phi_{\mathbf{k}}(\mathbf{x}) = \overline{\phi_{-\mathbf{k}}}(\mathbf{x})$, and $I_{\mathbf{mk}}$ can be replaced by $I_{\mathbf{m}(-\mathbf{k})}$.

Update step: When the measurement $\mathbf{y}_k$ becomes available, the PDF $p(\mathbf{y}_k|\mathbf{x}_k)$ is conditionalized to yield

$$\begin{aligned}
p(\mathbf{y}_k|\mathbf{x}_k) &= \sum_{\mathbf{n}\in\mathbb{N}^d}\sum_{\mathbf{m}\in\mathbb{N}^d} b_{\mathbf{nm}}\phi_{\mathbf{n}}(\mathbf{y}_k)\phi_{\mathbf{m}}(\mathbf{x}_k) \\
&= \sum_{\mathbf{m}\in\mathbb{N}^d} f_{\mathbf{m}}\phi_{\mathbf{m}}(\mathbf{x}_k),
\end{aligned}$$

where

$$f_{\mathbf{m}} = \sum_{\mathbf{n}\in\mathbb{N}^d} b_{\mathbf{nm}}\phi_{\mathbf{n}}(\mathbf{y}_k). \quad (13)$$

The multiplication in the update step is then carried out as:

$$\begin{aligned}
p(\mathbf{x}_k|\mathbf{y}_{1:k}) &= cp(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \\
&= c\sum_{\mathbf{n}\in\mathbb{N}^d} f_{\mathbf{n}}\phi_{\mathbf{n}}(\mathbf{x}_k)\sum_{\mathbf{m}\in\mathbb{N}^d} e_{\mathbf{m}}\phi_{\mathbf{m}}(\mathbf{x}_k) \\
&= c\sum_{\mathbf{n}\in\mathbb{N}^d}\sum_{\mathbf{m}\in\mathbb{N}^d} f_{\mathbf{n}}e_{\mathbf{m}}\phi_{\mathbf{n}}(\mathbf{x}_k)\phi_{\mathbf{m}}(\mathbf{x}_k) \\
&= c\sum_{\mathbf{n}\in\mathbb{N}^d}\sum_{\mathbf{m}\in\mathbb{N}^d} f_{\mathbf{n}}e_{\mathbf{m}}\sum_{\mathbf{k}\in\mathbb{N}^d} g_{\mathbf{nmk}}\phi_{\mathbf{k}}(\mathbf{x}_k) \\
&= c\sum_{\mathbf{k}\in\mathbb{N}^d} h_{\mathbf{k}}\phi_{\mathbf{k}}(\mathbf{x}_k),
\end{aligned}$$

where

$$h_{\mathbf{k}} = \sum_{\mathbf{n}\in\mathbb{N}^d}\sum_{\mathbf{m}\in\mathbb{N}^d} e_{\mathbf{m}}f_{\mathbf{n}}g_{\mathbf{nmk}}, \quad (14)$$

and c is a normalizing constant given by

$$c = \left[\sum_{\mathbf{n}\in\mathbb{N}^d} h_{\mathbf{n}}\int_{\mathbf{T}^d}\phi_{\mathbf{n}}(\mathbf{x}_k)d\mathbf{x}_k\right]^{-1}. \quad (15)$$

The filtering can thus be performed by recursively calculating Eq. (11) and Eq. (14).

To recapitulate: Given the prior distribution

$$p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) = \sum_{\mathbf{n}\in\mathbb{N}^d} a_{\mathbf{n}}\phi_{\mathbf{n}}(\mathbf{x}_{k-1}),$$

the steps required at one iteration of the algorithm to obtain the posterior distribution are:

- Perform the prediction step by evaluating (11).
- Calculate $f_{\mathbf{n}}$, $\mathbf{n}\in\mathbb{N}^d$ from (13).
- Perform the update step by evaluating $h_{\mathbf{k}}$, $\mathbf{k}\in\mathbb{N}^d$ from (14), and c from (15).
- The posterior distribution is now given from $p(\mathbf{x}_k|\mathbf{y}_{1:k}) = c\sum_{\mathbf{n}\in\mathbb{N}^d} h_{\mathbf{n}}\phi_{\mathbf{n}}(\mathbf{x}_k)$.

V. Mean and Covariance

The mean and covariance for the PDF $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ are typically of interest in estimation problems. The expected value in dimension i can be calculated by marginalizing the

expansion for the i-th dimension and taking the expected value of the marginalized distribution, i.e.

$$\begin{aligned} E[x_i|\mathbf{y}_k] &= \int_{\mathbf{T}^d} x_i p(\mathbf{x}_k|\mathbf{y}_{1:k}) d\mathbf{x}_k \\ &= \sum_{\mathbf{n}\in\mathbb{N}^d} a_{\mathbf{n}} \int_{\mathbf{T}^d} x_i \phi(\mathbf{x}_k) d\mathbf{x}_k. \end{aligned} \quad (16)$$

Let x_i denote the i-th element of $\mathbf{x}_k$. The covariance between x_i and x_j is given by

$$\begin{aligned} \mathrm{cov}(x_i, x_j) &= E[(x_i - E[x_i])(x_j - E[x_j])] \\ &= E[x_i x_j] - E[x_i]E[x_j], \end{aligned}$$

where the second term is evaluated using (16), while the first term can be calculated as

$$\begin{aligned} E[x_i x_j] &= \int_{\mathbf{T}^d} x_i x_j p(\mathbf{x}_k|\mathbf{y}_{1:k}) d\mathbf{x}_k \\ &= \sum_{\mathbf{n}\in\mathbb{N}^d} a_{\mathbf{n}} \int_{\mathbf{T}^d} x_i x_j \phi_{\mathbf{n}}(\mathbf{x}_k) d\mathbf{x}_k. \end{aligned} \quad (17)$$

VI. Practical issues

The practical aspects that have to be dealt with in an implementation of RBE using orthogonal basis functions are discussed in this section.

A. Truncation

For an implementation, the infinite series must be truncated to some order $N < \infty$, in each dimension. For simplicity of notation, it will be assumed that, in the multivariate case, the same approximation order is used for each dimension. The alternations in the formulae necessary for different approximation orders in different dimensions are straightforward.

In the update step (7), the order of the series expansion is doubled, in each dimension, due to the multiplication of series. Thus, to keep the order from growing exponentially, the series have to be truncated at each iteration. For simplicity, the truncation is made by keeping the first N terms. It should be noted that the truncation can result in an approximation $\hat{p}(\mathbf{x})$ that takes on negative values, and is hence not a PDF. However the purpose of the approximation is to make inference about the state $\mathbf{x}$, in this sense it is not worse to have $e(\mathbf{x}) = \hat{p}(\mathbf{x}) - p(\mathbf{x})$ negative than having $e(\mathbf{x})$ positive but merely $|e(\mathbf{x})|$ is of importance, as argued in [9].

B. Domain limitations

The discussion below concerns the one-dimensional case. For the multivariate case, the results hold for each dimension separately.

In general, the support of a PDF $p(x)$ is the whole real axis $\mathbb{R}$. However, for practical purposes, the PDF can be considered to have a compact support of $[a, b]$ if

$$x \notin [a, b] \implies p(x) < \epsilon$$

for some user-defined value of ϵ. The truncated support is denoted $\mathrm{supp}_\epsilon(\mathrm{p(x)}) = [\mathrm{a}, \mathrm{b}]$. Assume that the state and measurements are bounded $\forall k \geq 0$ as

$$a_x \leq x_k \leq b_x, \quad (18)$$

$$a_y \leq y_k \leq b_y, \quad (19)$$

and that, for a given $x_{k-1} \in [a_x, b_x]$, it holds that $\mathrm{supp}_\epsilon(\mathrm{p(x_k|x_{k-1})}) \subseteq [\mathrm{a}_1, \mathrm{b}_1]$. The approximation of $p(x_k|x_{k-1})$ must then be accurate over the domain

$$\{(x_k, x_{k-1}) | a_x \leq x_k \leq b_x, a_x - a_1 \leq x_{k-1} \leq b_x + b_1\}.$$

Similarly, for a given $y_k \in [a_y, b_y]$ and $\mathrm{supp}_\epsilon(\mathrm{p(y_k|x_k)}) \subseteq [\mathrm{a}_2, \mathrm{b}_2]$, the approximation of $p(y_k|x_k)$ must be accurate over the domain

$$\{(y_k, x_k) | a_y \leq y_k \leq b_y, a_y - a_2 \leq x_k \leq b_y + b_2\}.$$

C. Choice of basis functions

Since the computational burden will grow with the order of the series expansions N, it is highly desirable to use basis functions that allow for close approximations of the PDFs for as low N as possible.

In general, $p(x_k|x_{k-1})$ and $p(y_k|x_k)$ have unlimited support. In the estimation algorithm they are approximated offline over some finite domain. Using basis functions that decay rapidly near the domain borders, such as e.g. wavelets or Hermite functions, it can be challenging to give an accurate approximation with a low number of terms. In Fig. 2, the PDF $p(x_k|x_{k-1})$ is plotted for the system $x_k = x_{k-1}^2 + w_{k-1}$, where w_{k-1} is normally distributed. Approximations of the PDF by Fourier basis functions and Hermite basis functions, with $N = 15$ terms in each dimension, are also shown. As each Hermite function has the factor e^{-x^2} and hence decays rapidly, it is difficult to capturer the shape of $p(x_k|x_{k-1})$ using few terms. [14] uses wavelets and indeed an approximation order of $N = 125$ is used in one dimension, whereas in the Fourier functions example in Sec. VII only $N = 15$ i s required to give a satisfactory tracking accuracy, for a similar problem.

D. Computational complexity and parallel implementation

The PDFs $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p(\mathbf{y}_k|\mathbf{x}_k)$ are assumed to be determined beforehand and offline.

At each iteration Eq. (11) and Eq. (14) have to be evaluated. The prediction step (11) requires N^{2d} multiplications and $(N-1)^{2d}$ additions. The computational cost of the update step in (14) is dependent on $g_{\mathbf{nmk}}$. In many cases $g_{\mathbf{nmk}}$ is zero, except for a few certain values of $\mathbf{n}$ and $\mathbf{m}$, which property will reduce the computational complexity of (14). However, if none of $g_{\mathbf{nmk}}$ is zero, the update step requires N^d multiplications and $N^d(N-1)^d$ additions. The total flop count is then given by

$$c(N, d) = 3N^{2d} + 2N^d(N-1)^d.$$

An implementation advantage with the series expansion approach is that all required computations consist only of

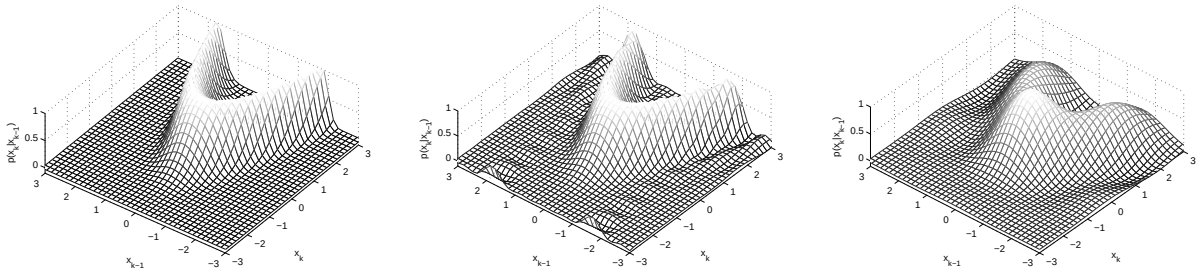

Fig. 2. Left figure: the true PDF $p(x_k|x_{k-1})$. The middle figure: approximation using Fourier basis functions. The right figure: approximation using Hermite basis functions. Both approximations make use of $N = 15$ first basis functions in each dimension.

summation of products. This algorithm therefore belongs to the class of so-called "embarrassingly parallel algorithms", and can straightforwardly be parallelized by assigning a sub-sum for evaluation to each processing unit.

More precisely, assume that M processing units are available. If $\mathbf{N}$ is a set of cardinality N, $\mathbf{N}_m$ is a subset of $\mathbf{N}$ of cardinality N/M (assumed to be integer) and $\mathbf{N}_m \cap \mathbf{N}_n = \varnothing$, $\bigcup_{m=1}^{M} \mathbf{N}_m = \mathbf{N}$ then processing unit m will calculate the following quantities:

$$\begin{aligned} e_{\mathbf{n}} &= \sum_{\mathbf{m}\in\mathbf{N}} a_{\mathbf{n},\mathbf{m}} c_{\mathbf{m}}, \quad \mathbf{n} \in \mathbf{N}_m, \\ h_{\mathbf{k}}^{(m)} &= \frac{1}{a} \sum_{\mathbf{m}\in\mathbf{N}_m} \sum_{\mathbf{n}\in\mathbf{N}} e_{\mathbf{m}} f_{\mathbf{n}} g_{\mathbf{n},\mathbf{m},\mathbf{k}}. \end{aligned}$$

All the sub-sums are summed up by a single processing unit as a last step to yield

$$h_{\mathbf{k}} = \sum_{m=1}^{M} h_{\mathbf{k}}^{(m)}.$$

VII. Example

For illustration of the suggested estimation concept, a scalar but realistic numerical example demonstrating is given below. A bearings-only tracking problem with a severe non-linearity in the measurement equation is considered that is known to require nonlinear filtering to avoid divergence of the estimate [20].

An object is traveling along a straight line and noisy bearing measurements y_k of its position x_k are taken by a sensor, Fig. 3. The system model is given by

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ y_k &= \tan^{-1} x_k + v_k, \end{aligned}$$

where w_k is normally distributed with the mean $\mu_w = 0$ and the variance $\sigma_w^2 = 0.01$. The measurement noise v_k obeys the multi-modal PDF

$$p_v(v) = \frac{p_1}{\sigma_{v1}\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{v-\mu_{v1}}{\sigma_{v1}})^2} + \frac{p_2}{\sigma_{v2}\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{v-\mu_{v2}}{\sigma_{v2}})^2}$$

shown in Fig. 4, with $p_1 = 0.2$, $p_2 = 0.8$, $\sigma_{v1} = 0.1$, $\sigma_{v2} = 0.05$, $\mu_{v1} = 0.1$, $\mu_{v2} = -0.05$.

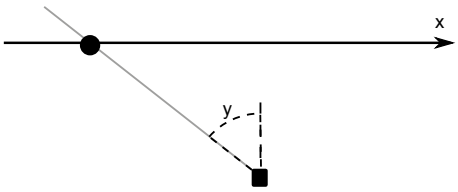

Fig. 3. An object (black dot) is traveling along a straight line. Noisy bearing measurements are taken by a sensor (black rectangle).

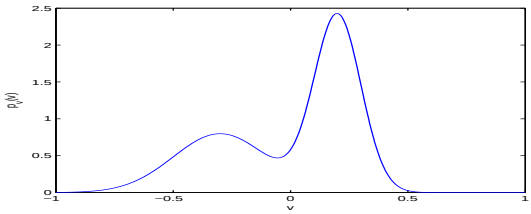

Fig. 4. Measurement noise PDF, $p_v(v)$.

A. Solution using Fourier basis functions

This section presents a solution of the bearing-only tracking problem obtained by applying the Fourier basis functions [21]

$$\phi_n(x) = \frac{1}{\sqrt{2\pi}} e^{inx}, \quad |n| \le \frac{N-1}{2}$$

that are orthogonal over the interval $[-\pi, \pi]$. To modify the basis functions to be orthogonal over a general interval, a linear transformation of x can be applied.

The sought estimate is the expected value of the approximated PDF $\hat{x}_k = E[x_k|y_k]$. From (16) and (17), the mean and covariance can be calculated as

$$E[x_k] = \sum_{n=-N/2}^{N/2} a_n \varphi_n,$$

$$E[(x_k - E[x_k])^2] = [\sum_{n=-N/2}^{N/2} \frac{-1}{in\pi} a_n \varphi_n] - E[x_k]^2,$$

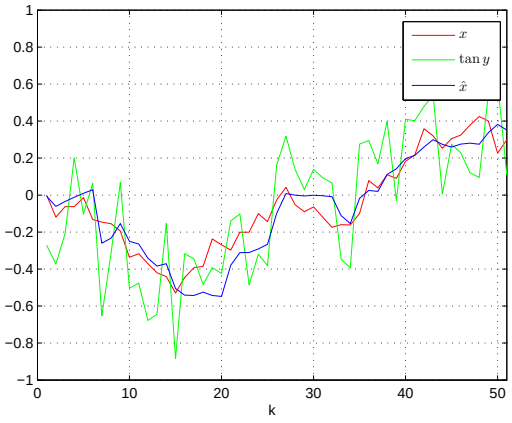

Fig. 5. True state x, tangens of the measurement $\tan y$ and estimated state $\hat{x}$, for time step k.

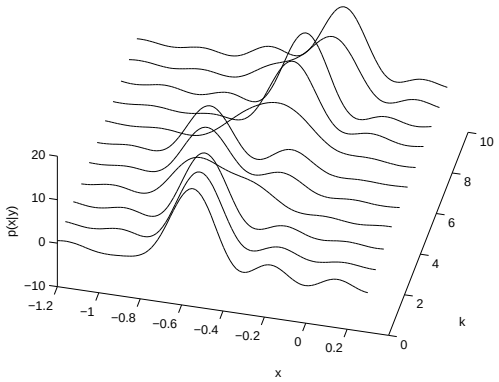

Fig. 6. $p(\mathbf{x}_k|\mathbf{y}_k)$ plotted for $k = 1, .., 10$

where φ_n is defined as

$$\varphi_n = \int_{-\pi}^{\pi} x\phi_n(x)dx = \frac{1}{\sqrt{2\pi}}\int_{-\pi}^{\pi} xe^{inx}dx = \begin{cases} 0 & if\ n = 0, \\ (-1)^{n+1}\frac{\sqrt{2\pi}}{n}i & \text{otherwise.} \end{cases}$$

Since $\phi_{\mathbf{n}}(\mathbf{x})\phi_{\mathbf{m}}(\mathbf{x}) = \phi_{\mathbf{n+m}}(\mathbf{x})$, it follows that

$$g_{\mathbf{nmk}} = \begin{cases} 1 & if\ \mathbf{n} + \mathbf{m} = \mathbf{k}, \\ 0 & \text{otherwise.} \end{cases}$$

The above result reduces the computational complexity of Eq. (14). The approximation order of $N = 15$ is selected. Fig. 5 depicts the true state, the measurement and the estimated state. In Fig. 6, the sequence of estimated PDF:s $p(x_k|y_k)$ is shown for $k = 1, 2, ..., 10$.

B. Particle filter comparison

For reference, the suggested nonlinear estimation method was compared to a SIR particle filter (PF) [19] in the root mean square error

$$E_{RMSE} = \sqrt{\frac{1}{K}\sum_{k=0}^{K}(x_k - \hat{x}_k)^2}$$

and the floating point operation (flop) demand.

Using the approximation of 10 flops to generate a random number, and 50 flops to evaluate the exponential function, the particle filter implementation requires about $410N_p$ flops per iteration, where N_p denotes the number of particles. Tab. I summarises the RMSE values and the flop count for different particle set sizes N_p. This can be compared to the RMSE=0.079 obtained by the series expansion filtering at a flop count of 1700 flops.

TABLE I

RMSE AND FLOP COUNT FOR A SIR PARTICLE FILTER IMPLEMENTATION.

N_p	50	100	200
E_{RMSE}	0.094	0.078	0.078
flops	$2.05 \cdot 10^4$	$4.1 \cdot 10^4$	$8.1 \cdot 10^4$

TABLE II

EXECUTION TIME FOR SINGLE CORE EXECUTION FOR DIFFERENT PROBLEM SIZES.

N	100	300	500	1000
Execution time	0.0021	0.0568	0.2625	2.1467

C. Execution time and speedup

The execution time and scalability for different problem sizes $N_T = N^d$ were studied. Tab. II shows the execution time for single core execution. Fig. 7 shows the speedup $s(M)$, i.e.

$$s(M) = \frac{t_1}{t_M},$$

where t_M is the execution time obtained employing M processors. The program was written in C++ using OpenMP for parallelization and execution was performed on a shared memory multicore processor (Quad-core Intel® Xeon 5520, Nehalem 2.26 GHz, 8MB cache).

VIII. DISCUSSION

A. Estimation accuracy

In the comparison to a SIR PF, it can be seen from Tab. I that about $N_p = 100$ particles are required to yield the same tracking performance as the suggested method. The computational cost of the PF for that number of particles is $4.1 \cdot 10^4/1700 \approx 24$ times greater than the cost for the orthogonal series expansion approach.

B. Speedup

One of the main advantages of the algorithm is its amenability to parallel implementation, as can be seen from the speedup plot in Fig. 7. The evolution of the multi- and many-core architectures is advancing at an increasing rate, while

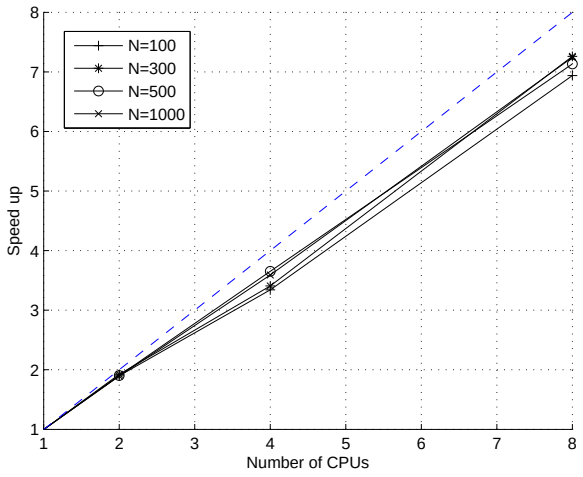

Fig. 7. Speedup plots for different values of N. For reference linear speedup is marked by the dashed line.

the single-core ones have basically stopped evolving. Therefore, it is highly desirable that an algorithm meant for online execution has good parallelization properties. As mentioned, Gaussian sum filters cannot be efficiently implemented in parallel due to the strong dependencies between the basis functions.

C. Limitations

The main limitations of the algorithm are the assumptions formulated in (18), (19). If $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p(\mathbf{y}_k|\mathbf{x}_k)$ are to be determined offline, the intervals $[a_x, b_x]$ and $[a_y, b_y]$ must be small enough relative $\mathrm{supp}_\epsilon(\mathrm{p}(\mathbf{x}_\mathrm{k}|\mathbf{x}_{\mathrm{k}-1}))$ and $\mathrm{supp}_\epsilon(\mathrm{p}(\mathbf{y}_\mathrm{k}|\mathbf{x}_\mathrm{k}))$. The PDFs $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p(\mathbf{y}_k|\mathbf{x}_k)$ will otherwise appear as "spikes" and will demand an unreasonable high approximation order to produce a good fit. If the expansions for the PDFs $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p(\mathbf{y}_k|\mathbf{x}_k)$ are updated online, restrictions (18), (19) can be dropped. Doing so will, however, require a large amount of online computation and, therefore, reduce the real-time feasibility of the method. Similar to most of the estimation techniques, the exponential growth of the computational complexity with the dimension is a limitation that confines the possible applications to relatively low-dimensional ones.

IX. Conclusions

A novel method for solving the recursive Bayesian estimation problem is presented. The method expands the involved probability density functions in series of orthogonal basis functions. The coefficients of the expansion are recursively calculated at each iteration through a prediction and an update step. The approach has two main advantages. First, a high estimation accuracy can be obtained by a small computational effort. Secondly, the orthogonality properties allows the computations to be separated into independent segments, making the algorithm very amenable for parallel implementation. A possible drawback is mainly that the state should be confined to a restricted domain. It is also possible to not specify a domain restriction on the state, but the method will then require considerably greater computational power.

References

[1] B. Anderson and J. Moore, *Optimal Filtering*. Dover Publications, Incorporated, 2012.

[2] H. Van Trees, *Detection, Estimation, and Modulation Theory*, ser. Detection, Estimation, and Modulation Theory. Wiley, 2004, no. del 1.

[3] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[4] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, Apr. 1970.

[5] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *Automatic Control, IEEE Transactions on*, vol. 45, no. 5, pp. 910 –927, may 2000.

[6] A. Doucet, N. de Freitas, N. Gordon, and A. Smith, *Sequential Monte Carlo Methods in Practice*, ser. Statistics for Engineering and Information Science. Springer, 2001.

[7] Z. Chen, "Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond," McMaster University, Tech. Rep., 2003.

[8] A. Budhiraja, L. Chen, and C. Lee, "A survey of numerical methods for nonlinear filtering problems," *Physica D: Nonlinear Phenomena*, vol. 230, no. 1-2, pp. 27–36, June 2007.

[9] B. Silverman, *Density Estimation for Statistics and Data Analysis*, ser. Monographs on Statistics and Applied Probability. Taylor & Francis, 1986.

[10] S. C. Schwartz, "Estimation of probability density by an orthogonal series," *The Annals of Mathematical Statistics*, vol. 38, no. 4, pp. pp. 1261–1265, 1967.

[11] M. Tarter and R. Kronmal, "On multivariate density estimates based on orthogonal expansions," *The Annals of Mathematical Statistics*, vol. 41, no. 2, pp. pp. 718–722, 1970.

[12] J. Gentle, *Elements of Computational Statistics*, ser. Statistics and Computing. Springer, 2002.

[13] D. Brunn, F. Sawo, and U. Hanebeck, "Nonlinear multidimensional bayesian estimation with Fourier densities," in *Decision and Control, 2006 45th IEEE Conference on*, dec. 2006, pp. 1303 –1308.

[14] A. Hekler, M. Kiefel, and U. Hanebeck, "Nonlinear bayesian estimation with compactly supported wavelets," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, dec. 2010, pp. 5701 –5706.

[15] D. Alspach and H. Sorenson, "Nonlinear bayesian estimation using gaussian sum approximations," *Automatic Control, IEEE Transactions on*, vol. 17, no. 4, pp. 439–448, Aug.

[16] T. Söderström, *Discrete-time stochastic systems: estimation and control*, ser. Prentice Hall international series in systems and control engineering. Prentice Hall, 1994.

[17] O. Rosen and A. Medvedev, "Parallelization of the Kalman filter for banded systems on multicore computational platforms," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, dec. 2012, pp. 2022 –2027.

[18] ——, "Efficient parallel implementation of state estimation algorithms on multicore platforms," *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 1, pp. 107 –120, jan. 2013.

[19] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, ser. Artech House Radar Library. Artech House, 2004.

[20] V. Aidala, "Kalman filter behavior in bearings-only tracking applications," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. AES-15, no. 1, pp. 29 –39, jan. 1979.

[21] A. Vretblad, *Fourier Analysis and Its Applications (Graduate Texts in Mathematics)*. Springer, Nov. 2010.