A Polynomial Approximation Algorithm for Real-Time Maximum-Likelihood Estimation

Christophe Villien and Eric P. Ostertag, *Member, IEEE*

***Abstract*—Maximum-likelihood estimation subject to nonlinear measurement functions is generally performed through optimization algorithms when accuracy is required and enough processing time is available, or with recursive filters for real-time applications but at the expense of a loss of accuracy. In this paper, we propose a new estimator for parameter estimation based on a polynomial approximation of the measurement signal. The raw dataset is replaced by $n+1$ independent polynomial samples (PS) for a smoothing polynomial of order n, resulting in a reduction of the computational burden. It is shown that the PSs must be sampled at some deterministic instants and an approximate formula for the variance of the PSs is also provided. Moreover, it is also proved and illustrated on three examples that the new estimator which processes the PSs is equivalent to the standard maximum-likelihood estimator based on the raw dataset, provided that the measurement function and its first derivatives can be approximated with a polynomial of order n. Since this algorithm proceeds from a compact representation of a measurement signal, it can find applications in real-time processing, power saving processing, or estimation based on compressed data, even if this latter field has not been investigated from a theoretical perspective. Its structure which is made up of several separate tasks is also adapted to distributed processing problems. Because the performance of the method is related to the polynomial approximation quality, the algorithm is well suited for smooth measurement functions like in trajectory estimation applications.**

***Index Terms*—Maximum-likelihood estimation, parameter estimation, polynomial approximation, real-time estimation.**

I. Introduction

THIS paper presents a method to reduce a dataset of measurements for nonlinear parameter estimation. Here, the focus is on real-time processing but the method can also find some applications in statistics or machine learning for speeding up nonlinear regressions based on large datasets for instance, or also as a means to compress data in distributed estimation. We consider the problem of estimating the unknown but deterministic $\boldsymbol{\theta} \in \Theta$ parameter of size m subject to the scalar measurement equation

$$y(t_k) = h(\boldsymbol{\theta}, t_k) + v(t_k) \tag{1}$$

Manuscript received October 25, 2007; accepted January 09, 2009. First published March 10, 2009; current version published May 15, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Brian M. Sadler.

C. Villien is with the Commissariat à l'Energie Atomique CEA/LETI, 38000 Grenoble, France (e-mail: christophe.villien@cea.fr).

E. P. Ostertag is with the Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection, LSIIT, UMR ULP-CNRS 7005, F-67412 ILLKIRCH, France (e-mail: Eric.Ostertag@ensps.u-strasbg.fr).

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.2009.2016875

where $h(\boldsymbol{\theta}, t_k)$ is an analytic measurement function and $v(t_k)$ is an independent Gaussian noise with mean $\mathrm{E}v(t_k) = 0$ and variance $\mathrm{E}v^2(t_k) = \sigma^2(t_k)$ supposed to be known. The method can be readily extended to vector measurements, provided that noises are mutually independent. We will focus on maximum-likelihood (ML) estimation since it presents nice properties like asymptotic efficiency, consistency, and normality. It is well known [1] that under these assumptions, the ML estimator is equivalent to a least-squares estimator $\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\Theta} C(\boldsymbol{\theta})$, where the quadratic cost function $C(\boldsymbol{\theta})$ is defined by

$$C(\boldsymbol{\theta}) = \frac{2}{N}\sum_{k=1}^{N} \frac{1}{\sigma^2(t_k)} \left(y(t_k) - h(\boldsymbol{\theta}, t_k)\right)^2 \tag{2}$$

where N is the number of samples and the $2/N$ factor is introduced here only for convenience but does not play any role in the estimation. A number of optimization algorithms, like the Gauss–Newton, the Levenberg–Marquardt (LM) [2] or the expectation-maximization algorithm [3], for instance, can be used to compute the estimate with an arbitrary degree of accuracy, provided that the algorithms converge. Unfortunately, because of the huge number of samples to process simultaneously, of the complexity and of the iterative nature of these algorithms, they are not well suited for real-time estimation. Therefore, many alternative strategies for ML estimation have been developed in order to increase the computational efficiency of the estimation algorithm, like reducing multidimensional search involved in ML estimation to multiple 1-D searches [5], [6] or using a hierarchical searching algorithm [7], for instance. However, although these algorithms reduce the complexity due to high dimensional parameter space, they do not address the problem of medium sized parameter estimation from large measurements sets.

Although they are designed for random state estimation instead of deterministic parameter estimation, recursive algorithms, such as the extended Kalman filter (EKF) [4] in its various forms, the unscented Kalman filter (UKF) [8]–[13], or particle filters (PF) [14] are sometimes preferred for practical purposes when processing time is critical. Indeed, implementation of these filters in a non-Bayesian setting is possible from a theoretical viewpoint by using a "diffuse" prior distribution for the parameter, i.e., a prior distribution that contains no information about the initial state, which often means in practice a large initial covariance matrix of the parameter. The benefit is that these filters offer a nice recursive structure and low computational complexity. Although a PF has a recursive structure, it presents a heavy computational load and will not be considered here. The EKF has been a standard technique in

1053-587X/$25.00 © 2009 IEEE

a number of nonlinear estimation problems but relies on a first order approximation of the nonlinear model, which can introduce large errors and may lead to suboptimal performance and sometimes even to divergence of the filter as pointed out in [8]. The UKF overcomes some of the EKF flaws as it is able to capture the first and second order moments of a Gaussian random variable propagated through the true nonlinear functions of the model. This improvement leads to better performance in many applications [8]–[13]. However, recursive filters are based on local approximations which remain suboptimal and are thus inferior to batch algorithms, as illustrated in the examples of Section VI or pointed out in [16]. Moreover, the complexity of the recurrence loop can still be problematic for very high sampling rates, especially for the UKF, or even for the square root UKF (SRUKF), which is a faster implementation of the UKF [11].

The originality of our approach is to use a global or piecewise approximation of the measurement function by a smoothing polynomial instead of a local approximation of the state distribution. The underlying idea is that, when a noise-free measurement signal can be approximated with a polynomial of order n, it seems that $n+1$ samples should be sufficient to characterize the whole signal since they describe a unique polynomial. Hence, when the polynomial order n is small compared to the number of raw measurements N, the gain in terms of number of samples to process $G = N/(n+1)$, also called processing gain hereafter, can be very high, and the ML estimation computation by an iterative method under real-time constraints could become tractable. However, for realistic signals, just picking up $n+1$ samples among N raw measurements will lead to poor performance because they are corrupted with noise, and some information can be lost. For this reason the measurement signal is smoothed with a polynomial, i.e., projected in a polynomial basis, to reduce the effects of noise. This is related to the principle used in [19] that the noise is "hard to compress" (think of projection as a means of compression) which also establishes general theorems on function learning using a compact representation (Occam approximation) and finds application to nonlinear filtering [20]. However, the accuracy of parameter estimation based on compressed data is not addressed in these papers and the proposed compression algorithm [21] is limited to first order polynomials and uses a different metric.

Our work is also closely related to Budin [22] and Musso [23]. Musso approximates the measurement function with the Lagrange interpolation polynomial and chooses the nodes of the Gauss quadrature formula for the interpolating point—which is the same as in our method—because they lead to a small bias. This approach, however, is severely limited in its application since the parameter to be estimated must be the values of the measurement function itself at the nodes of the Gauss quadrature formula and only the case of stationary variance is addressed. Contrary to our approach, the polynomial values are not treated as new samples for estimating the parameter and nothing is said about an estimation based on these values. Budin addresses several aspects of parameter estimation using least-squares polynomial smoothing of noisy observations, especially the choice of estimation time and the use of improper degree of polynomial to fit the observations. However, this approach is based on Taylor series of the measurement function around an estimation time to derive a relation between the parameter and the coefficients of the smoothing polynomial, but this could be inaccurate since the smoothing polynomial, i.e., the polynomial that solves the least-squares criterion, is generally different from the Taylor series. Moreover, the choice of the estimation time is based on a minimum variance of the Taylor series terms, but this does not guarantee a minimal variance for the parameter estimate and makes the estimation time depend on the application through the measurement function.

This paper considers a different method to compute an estimate from the smoothing polynomial, based on the evaluation of the smoothing polynomial at some deterministic instants, called the sampling instants (SI), resulting in a virtual sampling, which produces new measurements that are of the same nature as the raw measurements and alleviates issues due to the Taylor series calculation. The choice of the SIs is based on the independency of the polynomial samples (PS) instead of a minimum variance of the Taylor series coefficients, resulting in SIs which are independent of the measurement function. Our main result is that the SIs and the variance of the PSs match exactly the nodes and the inverse of the weights of the Gauss quadrature formulas [28]. The study of the Cramér–Rao lower bound (CRLB) shows that this result translates directly into the parameter estimation accuracy. Indeed, the Fisher information matrix (FIM) computed with the raw dataset is described by an integral which is replaced by a finite sum when the PSs are used instead, but since the terms in the finite sum match those of the quadrature formulae this latter is equal to the full integral with a very good accuracy, provided that the measurement function and its first derivatives can be approximated by a polynomial of order n. Because of the polynomial approximation, our approach is well suited for medium sized parameters ($m < 20$), smooth measurement functions, which means here time functions that can be fitted by polynomials of orders $n < 30$ in a given time interval, and large sample collections with respect to the polynomial order ($N > 10\ n$) in the same time interval. Typically, it can find applications in navigation [12], ballistic missile tracking [15], [16], or target motion analysis [17], [18], for instance.

Another feature of what we will call hereafter a polynomial approximation maximum likelihood (PAML) estimator is that it consists of three separate tasks, namely polynomial approximation, PS evaluation (virtual sampling) and parameter estimation based on the PSs, which makes the algorithm structure suitable for distributed processing or for estimation via compressed information [24]–[26] as illustrated by Example 3 in Section VI, even if this latter has not been investigated from a theoretical viewpoint.

The paper is organized as follow. Sections II and III present the algorithm and focus on the statistics of the PSs for an optimal virtual sampling. Section IV deals with the equivalence between the ML estimator and the PAML estimator. Section V provides some useful guidelines for the implementation of the compression algorithm. Section VI gives two examples of application for the real-time estimation problem and one example concerning the distributed processing estimation, and Section VII ends this paper with a few words of conclusion.

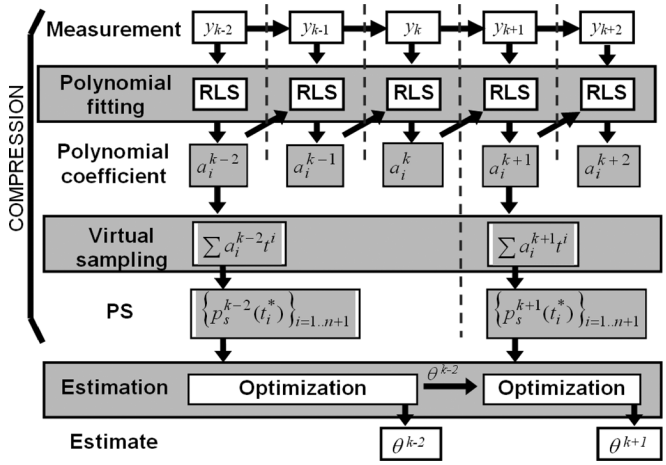

Fig. 1. Illustration of the algorithmic structure for PAML estimation.

II. Algorithm Description

The algorithm consists of three tasks: first, the raw data are approximated by a polynomial $p_s \in \mathrm{P}_n$ of order n, called the smoothing polynomial; second, a virtual sampling is performed by evaluation of the smoothing polynomial at some deterministic instants $\left\{ t_1^*, \ldots, t_{n+1}^* \right\}$ to generate PSs; and third, the PSs feed the estimation task to perform the parameter estimation as illustrated in Fig. 1. The virtual sampling and the estimation tasks occur only if a new estimate is needed and could run over several sampling periods.

The smoothing polynomial is chosen to minimize the same least-squares criterion as in (2). The coefficients of this polynomial can be computed in real time very easily with a recursive least-squares (RLS) algorithm, since it is a linear problem (see Section V). The SIs are chosen in such a way that the PSs are independent. It will be shown that, for a smoothing polynomial of order n, there are $n+1$ independent PSs whose SIs depend only upon the smoothing order n and some weighting function ω which is related to the evolution of the noise variance. Assuming the independency of the PSs, the PAML estimator is given by

$$\hat{\boldsymbol{\theta}}^* = \arg\min_{\boldsymbol{\theta}\in\Theta} C^*(\boldsymbol{\theta}) \tag{3}$$

with

$$C^*(\boldsymbol{\theta}) = \sum_{k=1}^{n+1} \frac{1}{\sigma_{PS}^2(t_k^*)} \left(p_s(t_k^*) - h(\boldsymbol{\theta}, t_k^*)\right)^2 \tag{4}$$

where $\sigma_{PS}^2(t_k^*)$ is the variance of the PS $p_s(t_k^*)$. This new estimator has the same form as the one in (2) except that n, the order of the smoothing polynomial, has replaced N, the length of the raw data segment.

The determination of the order of the smoothing polynomial remains an unsolved issue from a theoretical point of view. Although several conditions will be expressed concerning the approximation error, and thus the polynomial order, it seems to us that these disparate conditions can be merged together. Whatever that may be, it should be noted that, according to the Weierstrass theorem [27], the approximation error of regular functions by polynomials can be made arbitrarily small by increasing the polynomial order, especially for discrete sampling where the interpolating polynomial of the data leads to a perfect approximation. As a consequence, the method can always be applied, whatever the measurement function, by increasing the smoothing polynomial order, at the price of a bigger number of PSs and thus a smaller processing gain.

Practically, increasing the order is not a good solution since it worsens numerical stability and increases computational load. An alternative is then to use piecewise approximations of the measurement signal. In this case, the polynomial order is fixed at some arbitrary value and the length of the segment of approximation must be determined.

III. Polynomial Samples

Without any loss of generality we will assume that all the $\{t_k\}_{k=1\ldots N}$ belong to $[-1,\ 1]$.

By introducing the inner product

$$\langle f, g \rangle_\omega = \int_{-1}^{1} \omega(t) f(t) g(t) dt \tag{5}$$

and assuming that the sampling rate is high enough for the following integral approximation to hold

$$\int_{-1}^{1} \omega(t) f(t) g(t) dt \approx \frac{2}{N} \sum_{k=1}^{N} \omega(t_k) f(t_k) g(t_k), \tag{6}$$

the cost function can also be expressed in terms of the norm of the difference

$$C(\boldsymbol{\theta}) = \left(\sigma_0^2\right)^{-1} \|y - h(\theta)\|_\omega^2 = \left(\sigma_0^2\right)^{-1} \langle y - h(\theta), y - h(\theta) \rangle_\omega \tag{7}$$

where the weighting function is defined as $\omega(t) = \alpha(t)^{-1}$, with $\sigma^2(t) = \sigma_0^2 \alpha(t)$ and $\max[\alpha(t)] = 1,\ \ \alpha(t) > 0$.

The smoothing polynomial is chosen so as to satisfy the minimum norm criterion

$$p_s = \arg\min_{p\in\mathrm{P}^n} \frac{1}{\sigma_0^2} \|y - p\|_\omega^2. \tag{8}$$

Considering the space of the continuous functions $L_\omega^2([-1,\ 1])$ equipped with the inner product (5), the determination of the smoothing polynomial turns out to be a problem of approximation in Hilbert space. From the orthogonality principle [4], we know that the smoothing polynomial is given by the orthogonal projection

$$p_s = \sum_{i=0}^{n} \langle y, b_i \rangle_\omega b_i, \tag{9}$$

where the polynomials $\{b_i\}_{i=0\ldots n}$ form the orthonormal basis of $\mathrm{P}_n([-1, 1])$, i.e., $\langle b_i, b_j \rangle_\omega = \delta_{ij}$ with $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise, with respect to the inner product (5). Since the orthogonal projection is a linear operation which preserves the normality, the PSs have a Gaussian distribution.

From (9) and the linearity of the inner product, the smoothing polynomial can be written as the sum $p_s = p_h + p_v$, with

$$p_h = \sum_{i=0}^{n} \langle h, b_i \rangle_\omega b_i \qquad p_v = \sum_{i=0}^{n} \langle v, b_i \rangle_\omega b_i. \tag{10}$$

According to this decomposition, p_h stands for the smoothing polynomial of the measurement function only, and p_v for the smoothing polynomial of the noise only. Invoking the linearity of the mean operator, the mean of a polynomial sample evaluated at time t_k is given by

$$\mathrm{E}\, p_s(t_k) = \mathrm{E}\, p_h(t_k) + \mathrm{E}\, p_v(t_k). \tag{11}$$

Since the noise is supposed to be zero mean, we have

$$\mathrm{E}\, p_v(t_k) = \sum_{i=0}^{n} \langle \mathrm{E}v, b_i \rangle_\omega b_i(t_k) = 0 \tag{12}$$

and thus

$$\mathrm{E}\, p_s(t_k) = \mathrm{E}\, p_h(t_k). \tag{13}$$

Assuming that the approximation error is small compared with the residual noise

$$|h(\boldsymbol{\theta}, t_k) - p_h(t_k)| \ll \sqrt{\mathrm{E}\, p_v(t_k)^2} \tag{14}$$

we have $\mathrm{E}\, p_s(t_k) \approx h(\boldsymbol{\theta}, t_k)$.

The covariance between two PSs evaluated at time t_k and t_l is given by

$$\begin{aligned} C(t_k, t_l) &= \mathrm{E}\, p_v(t_k) p_v(t_l) \\ &= \sum_{i,j}^{n} \mathrm{E}\left[\langle v, b_i \rangle_\omega \langle v, b_j \rangle_\omega\right] b_i(t_k) b_j(t_l). \end{aligned} \tag{15}$$

From (6), we have

$$\begin{aligned} &\mathrm{E}\left[\langle v, b_i \rangle_\omega \langle v, b_j \rangle_\omega\right] \\ &\quad = \left(\frac{2}{N}\right)^2 \sum_{k,l}^{N} \omega(t_k)\omega(t_l) b_i(t_k) b_j(t_l) \mathrm{E}\left[v(t_k) v(t_l)\right]. \end{aligned} \tag{16}$$

Since the noise is assumed to be independent,

$$\mathrm{E}\left[v(t_k) v(t_l)\right] = \sigma_0^2 \alpha(t_k) \delta_{kl}, \tag{17}$$

it follows from the orthonormality of the polynomials $\{ b_i, b_j \}$ that

$$\begin{aligned} &\mathrm{E}\left[\langle v, b_i \rangle_\omega \langle v, b_j \rangle_\omega\right] \\ &\quad = \sigma_0^2 \left(\frac{2}{N}\right)^2 \sum_{k=1}^{N} \omega(t_k) b_i(t_k) b_j(t_k) = \frac{2\sigma_0^2}{N} \delta_{ij}. \end{aligned} \tag{18}$$

The substitution of (18) in (15) yields

$$C(t_k, t_l) = \sigma_0^2 \frac{2}{N} \sum_{i=0}^{n} b_i(t_k) b_i(t_l)\ . \tag{19}$$

The right-hand side is called the Christoffel–Darboux kernel [27] and is a polynomial of order $2n$ of two variables. The $2/N$ factor indicates the averaging effect of the smoothing polynomial, which reduces the noise variance. Invoking the Gaussian distribution of the PSs, we know that, if two PSs evaluated at times t_k and t_l are uncorrelated, they are also independent. Thus, if the SIs t_k and t_l verify $C(t_k, t_l) = 0$, then the associated PSs are independent.

Lemma 1: For a smoothing polynomial of order n, there are always $n+1$ independent PSs.

Proof: By the Christoffel–Darboux formula [27], we obtain

$$\begin{aligned} C(t_k, t_l) = \sigma_0^2 \frac{2}{N} \frac{a_n}{a_{n+1}} \\ \cdot \frac{b_{n+1}(t_k) b_n(t_l) - b_n(t_k) b_{n+1}(t_l)}{t_k - t_l},\ \forall\, t_k \neq t_l \end{aligned} \tag{20}$$

where a_n and a_{n+1} are the leading coefficients of the orthonormal polynomials b_n and b_{n+1}, respectively. Since b_{n+1} is orthonormal, it has exactly $n+1$ distinct zeros $\{ t_k^* \}_{k=1\ldots n+1}$ in $]-1,\ 1[$ [27]. Moreover, zeros of b_{n+1} verify $C(t_k^*, t_l^*) = 0,\ \forall k \neq l$.

Corollary 1: The PSs for a smoothing polynomial of order n should be evaluated at the $n+1$ zeros of b_{n+1}.

Indeed, this heuristic appears to be the best choice for the following reasons. Regarding the number of PSs, since a polynomial of order n is described uniquely by $n+1$ values, choosing less than $n+1$ PSs will result in a loss of information, whereas choosing more than $n+1$ PSs will result in an information redundancy, the extra PSs being linear combinations of the other PSs. Regarding the choice of the SIs, the zeros of b_{n+1} lead to a diagonal covariance matrix, which simplifies the computation of the estimate and also results in a good accuracy as will be detailed in Section IV.

The variance of the PSs is given by

$$\sigma_{PS}^2(t_k^*) = C(t_k^*, t_k^*) = \sigma_0^2 \frac{2}{N} (\lambda_n(t_k^*))^{-1} \tag{21}$$

where $(\lambda_n(t_k^*))^{-1} = \sum_{i=0}^{n} {b_i}^2(t_k^*)$ are the Christoffel numbers [27].

Lemma 2: The variance of the PSs can be approximated by

$$\sigma_{PS}^2(t_k^*) \approx \sigma^2(t_k^*) \frac{n+1}{N} \omega_T \tag{22}$$

where ω_T is the Chebychev weight defined by

$$\omega_T(t) = \frac{1}{\pi\sqrt{1-t^2}}, \quad t \in]-1,\ 1[. \tag{23}$$

Proof: It is shown in [29] that if ω belongs to the Szegö class, i.e., $\log(\omega) \in L^1$, the following convergence holds:

$$\forall t \in [-1,\ 1], \quad \lim_{n\to\infty} (n+1)\lambda_n(t) = \frac{\omega(t)}{\omega_T(t)}. \tag{24}$$

Assuming that the following approximation holds even for small n:

$$\lambda_n(t) \approx \frac{\omega(t)}{(n+1)\omega_T(t)} \tag{25}$$

the substitution of (25) into (21) and the use of the definition of the weight $\omega(t)$ yield (22).

IV. Performance of PAML

The new estimator (4) is elaborated by sampling the polynomial at the zeros of b_{n+1} associated with the weights $1/\sigma_{PS}^2$, which can be calculated through the exact formula (21) or the approximate formula (22). It should be emphasized here that both the SIs and the variance of the PSs depend only on the polynomial order and the noise variance evolution through $\alpha(t)$, and not on the measurement function $h(\boldsymbol{\theta}, t_k)$.

At this point, let us focus on the dataset reduction effects on the final parameter estimation. We claim that, for regular enough measurement functions, it is possible to find an order n such that the PAML is equivalent to the standard ML estimator (2). This relies on the following important property of the method.

Lemma 3: Let $\left\{ t_1^*, \ldots, t_{n+1}^* \right\}$ be the zeros of b_{n+1}. For any $f, g \in \mathrm{P}_N$ the following holds:

$$\langle f, g \rangle_\omega = \sum_{k=1}^{n+1} \lambda_n(t_k^*) f(t_k^*) g(t_k^*). \tag{26}$$

Proof: Observing that $\left\{ t_1^*, \ldots, t_{n+1}^* \right\}$ and $\lambda_n(t_k^*)$ coincide exactly with the nodes and the weights of the $(n+1)$-points Gauss quadrature formula (see the Appendix), respectively, and letting $\varphi(t) = f(t)g(t)$, we have

$$\int_{-1}^{1} \omega(t)\varphi(t)dt = \sum_{k=1}^{n+1} \lambda_n(t_k^*)\varphi(t_k^*) + \varepsilon_{n+1}(\varphi) \tag{27}$$

which is exact ($\varepsilon_{n+1} = 0$) because $\varphi \in \mathrm{P}^{2n}$ and the $(n+1)$-points quadrature formula has a degree of exactness of $2n+1$. This important property leads to the following lemma.

Lemma 4: Given a set of measurements $\{ y_k \}_{k=1\ldots N}$, and assuming that the measurement function and its first derivatives can be approximated with polynomials of order n, the PAML estimator leads to the same estimate as the standard ML estimator.

Proof: The ML estimate (2) based on the whole data is solution of the equation $\nabla C(\hat{\boldsymbol{\theta}}) = 0$, or equivalently $\left\langle \nabla h(\hat{\boldsymbol{\theta}}), y - h(\hat{\boldsymbol{\theta}}) \right\rangle_\omega = 0$, where ∇ denotes the gradient with respect to the parameter $\boldsymbol{\theta}$. The estimate $\hat{\boldsymbol{\theta}}^*$ of the PAML estimator is solution of the equation

$$\sum_{k=1}^{n+1} \lambda_n(t_k^*) \nabla h(\hat{\boldsymbol{\theta}}^*, t_k^*) \left[p_s(t_k^*) - h(\hat{\boldsymbol{\theta}}^*, t_k^*) \right] = 0. \tag{28}$$

By assuming that the order n is high enough to neglect $\varepsilon_{n+1}(\nabla h(\boldsymbol{\theta}) h(\boldsymbol{\theta}))$, and applying Lemma 3, we obtain

$$\left\langle \nabla h(\hat{\boldsymbol{\theta}}^*), p_s - h(\hat{\boldsymbol{\theta}}^*) \right\rangle_\omega = 0. \tag{29}$$

Substituting p_s with (9) yields

$$\begin{aligned} &\left\langle \nabla h(\hat{\boldsymbol{\theta}}^*), \sum_{i=0}^{n} \langle y, b_i \rangle_\omega b_i - h(\hat{\boldsymbol{\theta}}^*) \right\rangle_\omega = 0 \\ \Leftrightarrow &\left\langle \nabla h(\hat{\boldsymbol{\theta}}^*), \sum_{i=0}^{n} \langle y, b_i \rangle_\omega b_i \right\rangle_\omega - \left\langle \nabla h(\hat{\boldsymbol{\theta}}^*), h(\hat{\boldsymbol{\theta}}^*) \right\rangle_\omega = 0. \end{aligned} \tag{30}$$

Observe that

$$\begin{aligned} &\left\langle \nabla h(\hat{\boldsymbol{\theta}}^*), \sum_{i=0}^{n} \langle y, b_i \rangle_\omega b_i \right\rangle_\omega \\ &\quad = \left\langle y, \sum_{i=0}^{n} \left\langle \nabla h(\hat{\boldsymbol{\theta}}^*), b_i \right\rangle b_i \right\rangle_\omega = \left\langle y, p_{\nabla h}(\hat{\boldsymbol{\theta}}^*) \right\rangle \end{aligned} \tag{31}$$

where $p_{\nabla h}(\hat{\boldsymbol{\theta}}^*)$ stands for the polynomials of the orthogonal projections of the functions $\nabla h(\hat{\boldsymbol{\theta}}^*)$ onto P^n, and that the hypothesis $\nabla h(\boldsymbol{\theta}) \in \mathrm{P}_n$ yields $p_{\nabla h}(\hat{\boldsymbol{\theta}}^*) \approx \nabla h(\hat{\boldsymbol{\theta}}^*)$. Finally, the estimate of the PAML estimator is solution of $\left\langle \nabla h(\hat{\boldsymbol{\theta}}^*), y - h(\hat{\boldsymbol{\theta}}^*) \right\rangle_\omega = 0$, which is the same equation as the standard ML estimator, leading to $\hat{\boldsymbol{\theta}} \approx \hat{\boldsymbol{\theta}}^*$. It should be emphasized here that the correspondence is not of a statistical nature but really deterministic. As a result, the PAML estimator is equivalent to the ML estimator and inherits from all its properties, in particular it is asymptotically unbiased and efficient [4].

It should be kept in mind that the equivalence of the two estimators relies upon a polynomial approximation of both the measurement function and its first derivative with respect to the parameter. The quality of approximation must be such that the rest of the $(n+1)$-points Gauss quadrature could be neglected. Fortunately, the convergence of the quadrature formula is very good in practice and the PAML estimator has shown to be equivalent to the ML estimator for the applications that have been tested (see Section VI). The benefit of the compression in terms of number of samples to process is given by the processing gain $G = N/(n+1)$. In this ratio, the denominator is determined by the minimal order of the smoothing polynomial which enables it to carry all the information contained in the measurement function, whereas the numerator depends on the sampling frequency.

V. Implementation

In this section, the focus is only on the polynomial smoothing and the virtual sampling tasks, the estimation task being dependent on the user application. It is assumed that the application is such that the maximal duration of observation is finite, $t \in [0, T_{\max}]$, $T_{\max} < \infty$, and that the whole dataset of measurements can be described by the parametric model (1). The method can also be extended to state space models under non-finite measurement duration, as will be illustrated in Example 3.

The first step is to determine the minimal order of the smoothing polynomial n that satisfies the approximation constraints

$$\forall \boldsymbol{\theta} \in \Theta \quad \int_0^{T_{\max}} \omega(t) h^2(\boldsymbol{\theta}, t) dt \gg \varepsilon_{n+1}\left(h^2(\boldsymbol{\theta}) \right) \tag{32}$$

and

$$\begin{aligned} &\forall \boldsymbol{\theta} \in \Theta \quad \forall i = 1, \ldots, m, \\ &\int_0^{T_{\max}} \omega(t) \frac{\partial h^2(\boldsymbol{\theta}, t)}{\partial [\boldsymbol{\theta}]_i} dt \gg \varepsilon_{n+1}\left(\frac{\partial h^2(\boldsymbol{\theta})}{\partial [\boldsymbol{\theta}]_i} \right) \end{aligned} \tag{33}$$

where $[\boldsymbol{\theta}]_i$ refers to the i^{th} element of $\boldsymbol{\theta}$ and ε_{n+1} to the rest of the $(n+1)$-points quadrature. Because of the complexity of

the two criteria, especially the validity upon every $\boldsymbol{\theta}$, empirical approaches are preferred. One can for instance choose an order n and test the approximations for either random values of $\boldsymbol{\theta}$ or specific values like worst case values. If the test ever fails, one should increase the value of n and try again until it succeeds.

The second step consists in determining the SIs and the variance of the PSs using the corollary 1 and the exact formula (21) or the approximate formula (22), respectively. For non stationary variance application, the SIs and the PSs variance must be computed for each time a new estimation is required. Efficient algorithms [30], [31] that stem from Gaussian quadrature rules computation can be used for that purpose, but since this involves complex calculations, values should be computed offline and stored into memory. The total memory occupation is $N_{\text{est}} \times 2(n+1)$ floating point values, where N_{est} is the number of estimations.

In case of stationary variance, the SIs and the variance of the PSs do not depend on time and only a single set of $2(n+1)$ values is necessary. The polynomial fitting task is performed with an RLS algorithm [4] described by the following equations:

$$\mathbf{K}_k = \mathbf{P}_{k-1}\mathbf{V}_k\left(\omega_k^{-1} + \mathbf{V}_k^T\mathbf{P}_{k-1}\mathbf{V}_k\right)^{-1} \tag{34}$$

$$\mathbf{A}_k = \mathbf{A}_{k-1} + \mathbf{K}_k\left(y_k - \mathbf{V}_k^T\mathbf{A}_{k-1}\right) \tag{35}$$

$$\mathbf{P}_k = \mathbf{P}_{k-1} - \mathbf{K}_k\mathbf{V}_k^T\mathbf{P}_{k-1} \tag{36}$$

where $\mathbf{V}_k^T = [v_0(t_k) \ \ldots \ v_n(t_k)]$ is a vector of length $n+1$ whose elements $\{v_i(t)\}$ form a polynomial basis (not necessarily orthogonal) of P_n, and $\mathbf{A}_k^T = [a_0 \ \ldots \ a_n]$ are the coefficients of the smoothing polynomial at time t_k with respect to this basis. For numerical purpose, the basis $\{v_i(t)\}$ is preferably composed with polynomials that are numerically stable, such as Legendre, Chebyshev, or Jabobi polynomials, rather than the canonical basis $\{1, t, t^2, \ldots, t^n\}$. The matrix $\mathbf{P}_k$ is of size $(n+1)\times(n+1)$ and is initialized at $\infty \times Id$, where Id is the identity matrix and ∞ an arbitrary huge scalar value. It should be noted here that only (35) must be computed online, whereas (34) and (36) can be computed offline which implies that the K_k vector must be stored into memory and optionally the V_k vector to achieve the minimal real-time processing. Hence, the only task which is constrained by the sampling frequency can be reduced to only $2(n+1)$ multiplications and $2(n+1)$ additions per sample and to a memory occupation of $N \times 2(n+1)$ floating point values, where N is the number of measurements approximated by the polynomial.

The virtual sampling task consists in the following equation:

$$i = 1, \ldots, n+1: \qquad p_s(t_i^*) = \mathbf{V}^T(t_i^*)\mathbf{A}_k \tag{37}$$

where the polynomial basis is evaluated at the SIs $\{t_1^*, \ldots, t_{n+1}^*\}$. This results in $2(n+1)^2$ floating point operations (flops) but should be performed only if an estimate is required. Table I gives a summary of the processing capacity required for the polynomial smoothing and virtual sampling tasks assuming the fastest configuration, i.e., as much computation as possible has been done offline. The estimation task is not considered here since it is very dependent on the method chosen by the user. A numerical application is also provided for a time varying noise variance, a smoothing polynomial of order $n = 5$, a number of $N = 1000$ raw measurements, an estimation every 10 samples, that is $N_{\text{est}} = 100$, and floating point numbers stored on 32 bits.

TABLE I
SUMMARY OF THE PROCESSING CAPACITY REQUIRED. (EXAMPLE $n = 5$, $N = 1000$, $N_{\text{est}} = 100$, float size = 32 bits).

Task	flops		*memory*	
	theory	example	*theory*	example
Polynomial smoothing	4 (n+1) flops/sample	24 flops/sample	2N (n+1) floats	48 Kbytes
Virtual sampling	2 (n+1)² flops/eval.	72 flops/sample	2N_{est}(n+1) floats	4.8 Kbytes

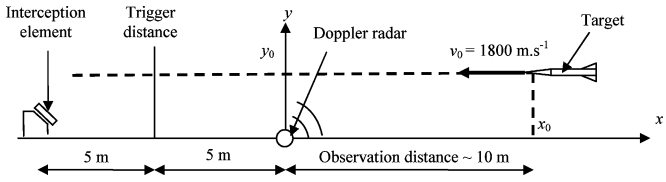

Fig. 2. Description of the test facility configuration for 2-D localization based on Doppler measurement.

VI. APPLICATION EXAMPLES

In this section, three application examples are provided. The first application, which is at the origin of the method, concerns the problem of accurate estimation under a very high sampling rate. The second application is a kind of extension of the first one to more complex measurement functions. In these cases, the PAML algorithm makes it possible to perform an optimization by means of an LM algorithm within very severe time constraints. The third one deals with the problem of distributed processing with a limited capacity of the radio link between the different sites of processing. It also provides a way of implementing the PAML algorithm for continuous observations and state space models.

A. Example 1

This first application stems from a problem which arises in the Active Protection context [32]. The objective is to estimate the trajectory of a very high speed (1000 to 1800 m $\cdot$ s^{-1}) ballistic target from Doppler radar measurements [33]. Since the target is observed at very close ranges, the trajectory is assumed to be a straight line and the velocity a constant. For this example, we consider the simplest case in two dimensions, with a single measurement of the radial velocity as depicted in Fig. 2, which corresponds to our test facility context. Full estimation of the trajectory in three dimensions with several radars is described in [34] and is very similar to the second example. In this scenario, the interception element is placed behind the radar and thus the trajectory can be observed until the target crosses the radar. Accounting for the constraints of the interception process, the trajectory must be estimated at about 5 m from the interception device, that is 5 m behind the radar. This implies that the estimation delay between the last measurement when the target crosses the radar and the moment when it reaches the trigger distance is less than 3 ms. On the other side, the Doppler radar generates a velocity measurement extracted from the Doppler signal

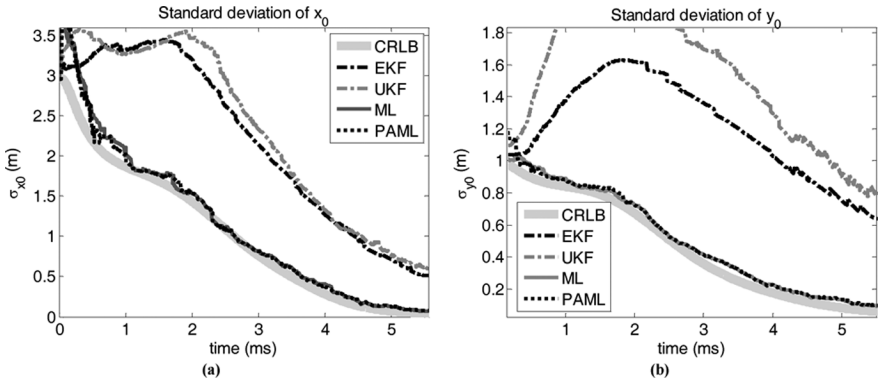

Fig. 3. Results of Monte Carlo simulations in the 2-D case: standard deviation of x0 (a) and y0 (b) estimated by recursive filters (EKF and UKF) and batch algorithms (LM and PAML).

by zero crossing detection every half-period of that signal, i.e., every 4 μs approximately, and a total amount of 1200 samples over the observation distance.

The parameter to be estimated is $\boldsymbol{\theta}^T = [x_0, y_0, v_0]$ and the measurement function is given by

$$h(\boldsymbol{\theta}, t) = kv_0 \frac{x_0 - v_0 t}{\sqrt{(x_0 - v_0 t)^2 + y_0{}^2}} \tag{38}$$

where $k = 66$ is the Doppler radar constant.

The variance evolution of the noise which results from the jitter of the zero crossing detection has been modeled in [34] according to

$$\sigma^2(t) = K(x_0 - v_0 t)^4 + \sigma_0^2 \tag{39}$$

with $K = 1.6 \times 10^4\ \mathrm{s}^{-2} \cdot \mathrm{m}^{-4}$, $x_0 = 10$ m, $v_0 = 1800\ \mathrm{m} \cdot \mathrm{s}^{-1}$, and $\sigma_0 = 1.19$ Hz.

Monte Carlo simulations using 300 runs and randomly generated parameters have been conducted to compare the performance of EKF, UKF, standard ML estimator using a batch LM method and the PAML approach with the same batch method but with PSs instead of raw measurements. The PAML uses a smoothing polynomial of order 5 resulting in only 6 PSs instead of the 1200 raw samples, and the same LM algorithm for PAML and ML has been used for estimation.

Fig. 3 shows the standard deviation of the estimates with respect to the duration of the observation. As expected, the ML and the PAML achieve the same performance and reach the Cramér–Rao lower bound in accordance with the property of asymptotic efficiency of the ML estimator. Here, the advantage of both the ML estimator and the PAML estimator over the recursive filters EKF and UKF is clear. The poor accuracy of the recursive filters results from a poor observability of the parameters at the beginning of the measurement process, i.e., when the target is far away from the radar, despite a good prior knowledge. Indeed, the poor observability of the parameters drives the recursive filters towards wrong estimates, which penalize further estimations even with better observability. At the opposite, batch methods account for the whole available information and the lack of observability of the first measurements does not affect the estimation.

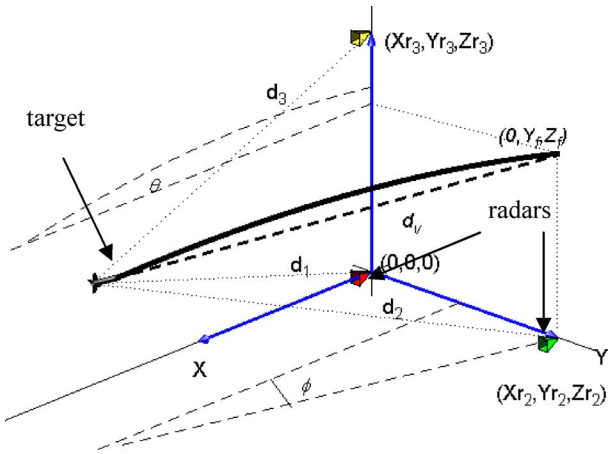

Fig. 4. Description of the scenario for a 3-D ballistic trajectory reconstruction from 3 MFCW radar measurements.

The polynomial fitting process has been implemented in real time on a dual processor board [33]. The processors are two TigerSHARC ADSP-TS101S, from Analog Devices, running at 250 MHz. An efficient programming at assembly level has made it possible to reduce the execution time of the polynomial fitting process down to 250 ns per sample.

B. Example 2

This second example is closely related to the first one, although this application of sniper detection is very different from active protection. Here, the objective is to estimate the trajectory of a bullet from distance measurements obtained from three multiple-frequencies continuous-wave (MFCW) radars, as illustrated in Fig. 4.

Contrary to the first example, the velocity is about $900\ \mathrm{m} \cdot \mathrm{s}^{-1}$ and the trajectory is observed over 30 m. This implies that the assumption of a straight line trajectory is no longer valid. The measurement function is then given by

$$h_i(\boldsymbol{\theta}, t) = \sqrt{(x(\boldsymbol{\theta}, t) - Rx_i)^2 + (y(\boldsymbol{\theta}, t) - Ry_i)^2 + (z(\boldsymbol{\theta}, t) - Rz_i)^2} \tag{40}$$

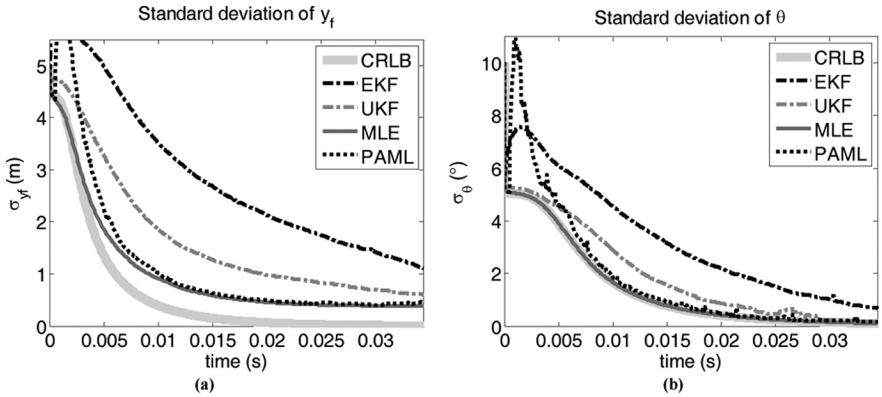

Fig. 5. Partial results of Monte Carlo simulation for a 3-D ballistic trajectory reconstruction: standard deviations of y_f (a) and $\theta(b)$ estimated by recursive filters (EKF and UKF) and batch algorithms (ML and PAML).

with $\{Rx_i, Ry_i, Rz_i\}$ the coordinates of the radar number i and $\{x(\boldsymbol{\theta},t), y(\boldsymbol{\theta},t), z(\boldsymbol{\theta},t)\}$ the coordinates of the bullet, according to the ballistic equation

$$\begin{aligned} x(\boldsymbol{\theta},t) &= \left[d_v - \frac{1}{\gamma}\left(1 - \frac{1}{1+\gamma v_0 t}\right)\right]\cos(\varphi) \\ y(\boldsymbol{\theta},t) &= \left[d_v - \frac{1}{\gamma}\left(1 - \frac{1}{1+\gamma v_0 t}\right)\right]\sin(\varphi) + y_f \\ z(\boldsymbol{\theta},t) &= \frac{t}{1+\gamma v_0 t}\left(v_0\theta - \frac{gt}{6}(3+\gamma v_0 t)\right) + z_f - d_v\theta. \end{aligned} \quad (41)$$

The parameter to be estimated is $\boldsymbol{\theta}^T = [y_f, z_f, d_v, v_0, \varphi, \theta, \gamma]$, where y_f and z_f denote the coordinates of the impact point, d_v and v_0 the distance and the velocity at the beginning of the measurement process, θ and φ the angles of incidence and γ a ballistic coefficient, which depends on the shape of the bullet.

The variance evolution of the measurements has been derived from the radar signal in [34] yielding

$$\sigma^2(t) = K(d_0 - v_0 t)^4 + \sigma_0^2 \quad (42)$$

with $K = 2.26 \times 10^{-8}$ m^{-2}, $d_0 = 30$ m, $v = 900$ m $\cdot$ s^{-1} and $\sigma_0 = 5$ cm.

Clearly, the measurement function is more complex than in the previous example. For this reason a smoothing polynomial of order 7 has been used here for a global approximation of the measurements. The EKF, UKF, ML, and PAML algorithms have been compared through Monte Carlo simulations using 20 000 runs with randomly generated parameters and 500 point trajectories. Some of the results are shown in Fig. 5. A discrepancy between MLE and PAML exists for a small amount of data, i.e., for $t < 5$ ms, because (6) is not valid and this is magnified by the poor observability of the parameters at the beginning of the measurement. Once again, ML and PAML performances nearly achieve the CRLB and are superior to EKF and UKF.

Execution times of PAML, EKF, and UKF for the two examples have been measured through Matlab simulations and are listed in Table II. Although results must be analyzed with caution since the low level routines of Matlab are out of reach, they give a good idea of the relative complexity of the algorithms. Values are normalized with respect to the polynomial fitting process of Example 1.

TABLE II
NORMALIZED MATLAB EXECUTION TIMES FOR THE PAML ALGORITHM (POLYNOMIAL FITTING AND LEVENBERG-MARQUARDT OPTIMIZATION) AND RECURSIVE FILTERS (EKF AND UKF)

	PAML		EKF	UKF
	Polynomial fitting	LM		
Example 1 $n=5$; $m=3$	1	109	12	138
Example 2 $n=7$; $m=7$	3.5	890	131	321

The execution time for the LM optimization algorithm, which is given for one iteration, is roughly ten times longer than for the EKF, and three times longer than for the UKF in the worst case. However, a significant difference between these algorithms is that, contrary to UKF and EKF, the estimation task of PAML is not constrained by the sampling frequency and can span over several sampling periods as illustrated in Fig. 1. On the other hand, the polynomial fitting stage which is the recursive kernel of the PAML method is much faster than the recursive filters. Hence, combination of the two processes results in optimal accuracy even with very high sampling frequencies.

C. Example 3

This last example is given to illustrate an implementation of the algorithm for state space models and piecewise polynomial approximation. Although these aspects have not been investigated from a theoretical perspective, this example also shows the potentials of the algorithm which are inherent to its structure, for addressing distributed processing and estimation based on compressed data problems. This application deals with ad hoc ultra wideband (UWB) networks for emergency scenarios [35]. Here, a network of UWB emitters is deployed on the scene of a disaster in order for the command post, called hereafter the central unit (CU), to localize firemen on intervention, the mobile units (MU), as depicted in Fig. 6. It is assumed that the UWB emitters can be self-localized in a first step and that their relative positions are known at the CU site. Then, pairs of UWB emitters broadcast in turns short radio pulses in a synchronized fashion

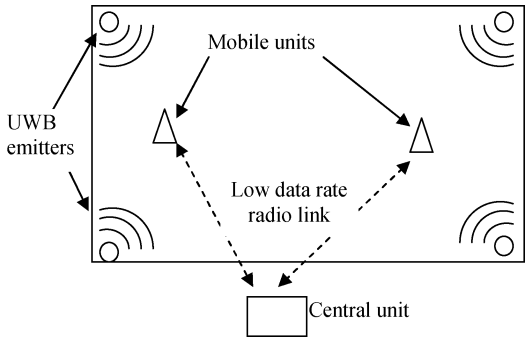

Fig. 6. Configuration of an UWB ranging scenario of several MU whose positions are computed by the central unit and where measurements are transmitted through a low data-rate radio link.

according to a particular protocol. The MU are equipped with an UWB receiver that can measure the time difference of arrival (TDOA) of these radio pulses [36]. These TDOA measurements are proportional to the difference of distance between the MU with coordinates (x, y) and emitters i and j, whose coordinates are (R_{xi}, R_{yi}) and (R_{xj}, R_{yj}), respectively:

$$h_{ij}(t,\boldsymbol{\theta}) = \frac{1}{c}\sqrt{(x(t)-R_{xi})^2+(y(t)-R_{yi})^2} - \sqrt{(x(t)-R_{xj})^2+(y(t)-R_{yj})^2} \quad (43)$$

with c the speed of light.

This is very similar to the global positioning system (GPS) except that the MU cannot compute its position because the positions of emitters are not known, and that its position should be known at the CU site. Therefore, the TDOA measurements of the MU are sent through the firemen professional mobile radio (PMR) to the CU where the positions can be calculated. However the transmission channel has a limited capacity of 8 Kb/s and is shared with vocal communications. There is therefore a big interest in reducing the data transmissions over the channel.

Contrary to the previous examples, estimation relates here to a state space model, because no analytic model of the trajectory is available, under continuous observations, rather than to a parametric model with finite length observations. In order to apply the compression algorithm, it is necessary to parameterize the model. For this reason, we consider a piecewise trajectory of order 2 for a segment duration of T :

$$t \in [0,T] \quad x(t,\theta) = x_0 + v_x(t) + \frac{1}{2}a_x t^2$$
$$y(t,\theta) = y_0 + v_y t + \frac{1}{2}a_y t^2. \quad (44)$$

Hence, each segment of the trajectory depends on the parameter $\boldsymbol{\theta}^T = [x_0, y_0, v_x, v_y, a_x, a_y]$. The measurement function is given by (40) where the dependence on (x, y) is replaced by a dependence on (t, θ) thanks to the parameterization (44). Although the noise variance depends on the distance between MU and emitter, it is assumed to be constant over a segment since small T corresponds to small displacements of firemen. Obviously the variance can be different from segment to segment but this only changes the factor σ_0^2 in the definition of the PS variance (21) and does not play any role in the PAML estimator (3).

TABLE III
RMS POSITION ERROR FOR MLE, CMLE AND EKF

	ML	PAML	EKF
RMS error	8 cm	8.1 cm	8.2 cm

For this example, four UWB emitters positioned at the corners of a 19-m side square produce a vector valued measurement $\mathbf{h}(t,\theta) = [h_{12}(t,\theta), h_{13}(t,\theta), h_{14}(t,\theta)]^T$ every 2 ms. The receiver is such that TDOA can take only 2048 values, which can thus be coded on 11 bits. As a consequence the transmission of raw data requires 16.5 Kb/s. For this application, the PAML algorithm uses a smoothing polynomial of order 5 for each scalar TDOA measurement and segments durations of $T = 2$ s. The PSs being floating point numbers, they are coded on 32 bits, which results in a channel load for the PAML algorithm of 288 b/s. Benefits of the PAML algorithm are thus a reduction of the channel load by a factor 57 and a reduction of the computational complexity at the CU.

Simulations using random trajectories with 200 s duration have been performed. Position errors for the EKF based on the true state space model, and the ML and PAML estimators using the parametric model are given in Table III.

Once again, PAML achieves the same performance as a standard ML estimator and EKF, but with smaller radio capacity and in a distributed processing fashion with smaller complexity at each processing site.

VII. CONCLUSION

A new estimator based on a polynomial approximation of the measurement signal has been proposed. This new estimator is computed from few PSs instead of many raw measurements, resulting in a reduction of the computational burden. It has been proven that, for a smoothing polynomial of order n, there are $n + 1$ independent PSs. An exact and an approximate formula of their variance have also been given. Assuming that the measurement function and its first derivatives with respect to the parameter can be approximated with a polynomial of order n, it has also be proven that the new estimator is equivalent to the standard ML estimator, thanks to a match between the SIs and the inverse of the variance of the PSs with the nodes and the weights of the Gauss quadrature formula. Moreover, the polynomial smoothing task is made up of an RLS algorithm which can be reduced to the very small number of $4(n + 1)$ operations per sample, where n stands for the order of the smoothing polynomial. One benefit of this new estimator is that it can be used for real-time applications where processing time is critical, while maintaining the accuracy obtained with standard batch algorithms. Three examples of application have shown that the PAML approach, which can be seen as a global/piecewise approximation, leads to better performance than the standard approaches based on recursive filters, like the EKF or the UKF which proceed from a local approximation. In future works we will study the use of the dataset reduction method as a means to reduce the complexity of recursive filters, like the particle filter, in order to extend the method to state estimation. We will

also investigate the possibility to use this estimator for compression/estimation issues and distributed processing in sensor networks, since its structure is well suited to that kind of problems.

APPENDIX

The Gauss quadrature formula, sometimes also called Gauss-Jacobi mechanical quadrature [27] or generalized Gauss quadrature, is a generalization of the classical Gauss quadrature to any weighting function ω. Let $\{ b_0, \ldots, b_n \}$ be a set of orthonormal polynomials of order at most n for the following inner product:

$$\langle f, g \rangle_\omega = \int_a^b \omega(t) f(t) g(t) dt \tag{45}$$

with $\omega > 0$ being any weighting function. It is a well known [27] property of orthogonal polynomials that b_i has exactly i distinct zeros in $[a, b]$. Let the nodes $\{ t_1^*, \ldots t_n^* \}$ of the n-points quadrature be the n zeros of b_n and the weights of the quadrature $\{\lambda_1, \ldots, \lambda_n\}$ be the Christoffel numbers, given by

$$\lambda_k^{-1} = \lambda^{-1}(t_k^*) = \sum_{i=0}^{n-1} b_i^2(t_k^*). \tag{46}$$

Then, for any polynomial π_{2n-1} of order at most $2n-1$, the following quadrature formula holds [28]:

$$\int_a^b \omega(t) \pi_{2n-1}(t) dt = \sum_{i=1}^{n} \lambda_i \pi_{2n-1}(t_i^*). \tag{47}$$

Moreover, for any function $\varphi \in L^1([a, b])$ integrable over $[a, b]$, we have

$$\int_a^b \omega(t) \varphi(t) dt = \sum_{i=1}^{n} \lambda_i \varphi(t_i^*) + \varepsilon_n(\varphi). \tag{48}$$

Assuming that φ is $2n$ times derivable, the remainder can be expressed as

$$\begin{aligned} \varepsilon_n(\varphi) &= \frac{\varphi^{(2n)}(\xi)}{(2n)!} \int_a^b \omega(t) \left[\prod_{i=1}^{n} (t - n t_i^*) \right]^2 dt \\ &= c_n \varphi^{(2n)}(\xi), \quad a < \varepsilon < b \end{aligned} \tag{49}$$

where the constant c_n does not depend on φ.

REFERENCES

[1] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques and Software*. Norwood, MA: Artech House, 1998.

[2] J. F. Bonnans, J. C. Gilbert, C. Lemarechal, and C. A. Sagastizabal, *Numerical Optimization: Theoretical and Practical Aspects*. New York: Springer-Verlag, 2002.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc.*, ser. 38, pp. 1–38, 1977.

[4] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 2000.

[5] S. Umesh and D. W. Tufts, "Estimation of parameters of exponentially damped sinusoids using fast maximum likelihood estimation with applications to NMR spectroscopy data," *IEEE Trans. Signal Process.*, vol. 44, no. 9, Sep. 1996.

[6] D. W. Tufts, G. Hongya, and S. Umesh, "Fast maximum likelihood estimation of signal parameters using the shape of the compressed likelihood function," *IEEE J. Ocean. Eng.*, vol. 18, no. 4, pp. 388–400, Oct. 1993.

[7] K. Matsumoto and T. Hattori, "Fast hierarchical searching algorithm for real time location tracking with maximum likelihood estimation," in *Proc. Vehicular Technology Conf. (VTC)*, Jun. 2005, vol. 5, pp. 2820–2824.

[8] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. AeroSense*, 1997.

[9] E. A. Wan and R. van der Merwe, "The unscented Kalman filters for nonlinear estimation," presented at the IEEE Symp. (AS-SPCC), Lake Louise, AB, Canada, Oct. 2000.

[10] G. F. Ma and X. Y. Jiang, "Unscented Kalman filter for spacecraft attitude estimation and calibration using magnetometer measurements," in *Proc. Int. Conf. Learning Machine Cybernetics*, Guangzhou, Aug. 2005.

[11] E. A. R. van der Merwe, "The square-soot unscented Kalman filter for state and parameter estimation," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 2001.

[12] J. H. Clements, "Recursive maximum likelihood estimation of aircraft position using multiple range and bearings measurements," in *Proc. IEEE Position Location Navigation Symp.*, Apr. 1996, pp. 199–204.

[13] J. Li and Y. Zhong, "Comparison of three Kalman filters for speed estimation of induction machines," in *Proc. Industry Applications Conf.*, Oct. 2005.

[14] A. Doucet, N. De Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York: Springer, 2001.

[15] M. Yeddanapudi, Y. Bar-Shalom, K. R. Pattipati, and S. Deb, "Ballistic missile track initiation from satellite observations," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 31, no. 3, pp. 1054–1071, Jul. 1995.

[16] A. Farina, S. Immediata, L. Timmoneri, M. Meloni, and D. Vigilante, "Comparison of recursive and batch processing for impact point prediction of ballistic targets," in *Proc. IEEE Int. Radar Conf.*, May 2005.

[17] K. Doganay, "Online optimization of receiver trajectories for scan-based emitter localization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 3, pp. 1117–1125, Jul. 2007.

[18] X.-J. Tao, C.-R. Zou, and Z.-Y. He, "Passive target tracking using maximum likelihood estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 32, no. 4, pp. 1348–1354, Oct. 1996.

[19] B. K. Natarajan, "Occam's razor for functions," in *Proc. 6th Annu. Conf. Computational Learning Theory*, 1993, pp. 370–376.

[20] B. K. Natarajan, "Filtering random noise via data compression," in *Proc. Data Compression Conf. (DCC)*, 1993.

[21] K. Konstantinides and B. K. Natarajan, "Algorithm and architecture for non-linear noise filtering via piecewise linear compression," *IEEE Trans. Signal Process.*, vol. 42, no. 9, Sep. 1994.

[22] M. A. Budin, "Parameter estimation using least-squares polynomial smoothing," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 4, pp. 371–381, Jul. 1973.

[23] C. J. Musso and C. G. Jauffret, "Linear maximum likelihood estimator," in *Proc. Acoustics, Speech, Signal Processing (ICASSP)*, 1991, pp. 1301–1304.

[24] Z. Zhang and T. Berger, "Estimation via compressed information," *IEEE Trans. Inf. Theory*, vol. 34, no. 2, pp. 198–211, Mar. 1988.

[25] M. L. Fowler and M. Chen, "Fisher-information-based data compression for estimation using two sensors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 3, pp. 1131–1137, Jul. 2005.

[26] P. Venkitasubramaniam, L. Tong, and A. Swami, "Quantization for maxmin ARE in distributed estimation," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3596–3607, Jul. 2005.

[27] G. Szegö, *Orthogonal Polynomials*. Providence, RI: Amer. Math. Soc., 1989.

[28] A. Ghizzetti and A. Ossicini, *Quadrature Formulae*. Cambridge, MA: Birkhäuser Verlag, 1970.

[29] A. Mate, P. Nevai, and V. Totik, "Szegö's extremum problem on the unit circle," *Ann. Math.*, vol. 134, pp. 433–453, 1991.

[30] G. H. Golub and J. H. Welsch, "Calculation of Gauss quadrature rules," *Math. Comput.*, no. 23, pp. 221–230, 1963.

[31] A. M. Mughal, X. Ye, and K. Iqbal, "Computational algorithm for higher order Legendre polynomial and Gaussian quadrature method," in *Proc. 2006 Int. Conf. Scientific Computing (CSC)*, 2006.

[32] T. J. Meyer, "Active protection systems: Impregnable armor or simply enhanced survivability," *Armor Mag.*, May–Jun. 1998.

[33] C. Villien, V. Fleck, E. Ostertag, and P. Raymond, "3-D short range localization device by low-cost CW-Doppler radar," presented at the IEEE Int. Radar Conf., Arlington, VA, May 2005.
[34] C. Villien, "Prévision de Trajectories 3-D en Temps Réel," Ph.D. dissertation, Strasbourg University, Alsace, France, 2006.
[35] C. Villien, F. T. Talom, and N. Daniele, "Comparison of receiver architecture in emergency scenario," presented at the UWB Wokshop, Grenoble, France, May 2007.
[36] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: Possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 41–53, Jul. 2005.

Christophe Villien was born in 1979. He received the Engineer degree from the Ecole Nationale Supérieure d'Electronique Informatique et Radioéléectricité (ENSEIRB), Bordeaux, France, in 2002 and the French Doctorate degree in control systems and signal processing from the University Louis Pasteur (ULP), Strasbourg, France, in 2006.

In 2006, he joined the French CEA, Grenoble, France, where he is working on UWB localization. His research interests are on estimation theory, radar systems, UWB localization, and embedded processing.

Eric P. Ostertag (M'89) received the Lic.es Sci. degree in physics from the University of Strasbourg, France, in 1961, the Electrical Engineering degree from the Ecole Supérieure d'Electricité, now Supélec, Gif-sur-Yvette, France, in 1963, and the Doctorat d'Etat (Ph.D.) degree in solid state physics from the University Louis Pasteur (ULP), Strasbourg, France, in 1977.

After having worked for several years as a Research and Development Engineer in nuclear electronics, first with the French Centre National de la Recherche Scientifique (CNRS), then with ORTEC, Inc., Oak Ridge, TN, and its subsidiary in Munich, Germany, in 1981, he joined the Ecole Nationale Supérieure de Physique de Strasbourg (ENSPS), where he was a Full Professor and taught in the area of control systems until his retirement in 2004. Since then he has become an Emeritus Professor of the ULP. In 1987, he initiated a research group on Automation, Vision and Robotics at the Laboratoire des Sciences de l'Image et de la Télédétection (LSIIT), Strasbourg, Germany. His research interests include computer aided design of control systems, fuzzy control, digital computer controlled systems, and RST structure, and optimal control.

Prof. Ostertag was a member of the Groupe de Recherche en Automatique of the CNRS and of the Société des Electriciens et des Electroniciens (SEE) for many years and has been a member of the Société des Ingénieurs Supélec since 1981 and of the Association Régionale des Ingénieurs et Scientifiques d'Alsace since 2006.