Grid Based Solution of Zakai Equation with Adaptive Local Refinement for Bearings-only Tracking

Huilong Zhang, Dann Laneuville

Grid Based Solution of Zakai Equation with Adaptive Local Refinement for Bearings-only Tracking

H. Zhang
IMB, UMR 5251, Université Bordeaux 1
351, Cours de la libération
F-33405 Talence Cedex
Huilong.Zhang@math.u-bordeaux1.fr

D. Laneuville
DCNS – SIS Division
BP 403 (Le Mourillon)
F-83055 Toulon Cedex
dann.laneuville@dcnsgroup.com

Abstract—The solution of the Zakai equation provides the complete conditional probability density of the state, given the observations. Numerical solution of this equation by the finite difference method usually leads to large systems of equations which have to be solved at each time step, especially when the dimension of state space is more than two. We propose in this paper, for the first time to our best knowledge, a grid-based four dimensional algorithm to solve the Zakai equation. Our approach is based on an adaptive local grid refinement method and is illustrated with a bearings-only target motion analysis example. [1][2]

TABLE OF CONTENTS

Keys words—Zakai Equation, FAC, local refinement, grid based method, Bearings-Only TMA, Unscented Kalman Filter.

1. INTRODUCTION

Conceptually, the basic problem in bearings-only tracking is to estimate the trajectory of a target (i.e. position and velocity) from noise-corrupted sensor bearing data. In the case of a single-sensor problem, these bearing data are obtained from a single-moving observer (ownship). We have then a four dimension state space problem: (x,y,u,v), where (x,y) and (u,v) are the position and velocity components, respectively. Contrary to Kalman type filters which calculate only the expected value and covariance matrix of the state system, the Zakai equation, a stochastic partial differential equation (PDE) whose solution is a conditional law given the past available observations, permit us not only to obtain an optimal non linear filter, but also to have a probability distribution in both position and velocity fields (confidence regions) at every time step. Tracking with hard constraints is not only possible but also inherent to the method. Important progresses have been made in the numerical resolution of this equation, for instance by using particle filters. In this article, we are interested in a grid-based approach.

In the literature, there exist five different methods for bearing-only tracking filtering: Kalman type filters [4] [8], particle filtering [14], convolution method [1] [7], quantization method [11] and Zakai filter [2]. In the first three approaches the system is considered discrete in time and continuous in state space; in the fourth approach, both time and state space is discrete; in our method, we consider that the system is continuous both in time and in state space.

In spite of the development of computer capability in the recent years, grid-based approximation has found little application in tracking problems. The question of how to solve the Zakai equation efficiently, so as to satisfy as many real-time constraints as possible, still remains, in particular when the space dimension is greater than two. Just like the difficulty encountered in 3-D fluid dynamic computation, the principal problem for multidimensional applications is that a large-scale system has to be solved at each time step. It makes the approach unusable, because the number of points in a multidimensional regular grid is considerable.

The basic remark that we are going to exploit is that in most applications, in particular when the observation noise is small, the conditional density is well localized in some small region in the state space. This is our motivation for using local refinement techniques. To track the solution of noisy state equations, whose shape may be arbitrary and in general could not be predicted in advance, we use an a posteriori criterion. In order to overcome the resulting programming difficulties and optimize software efficiency, we have used the C++ programming language and convenient data structure.

1

[1] 1-4244-1488-1/08/$25.00 ©2008 IEEE.

[2] IEEEAC paper#1217, Version 4, Updated 2007:10:28

The paper is organized as follows: first, a discretization scheme is presented for numerical approximation. An Euler implicit scheme proposed by Le Gland [9] is used for time discretization and an up-wind finite difference scheme due to Kushner [5] [6] is applied on a composite grid for space discretization. Second, the problem of bearings-only tracking is modeled by the Zakai equation; a fast adaptive composite multigrid algorithm will be detailed. Then, using a typical testing scenario, some numerical results are presented and compared with an Unscented Kalman Filter (UKF). Numerical experiments confirm feasibility and efficiency of the method. z

2. THE ZAKAI EQUATION AND DISCRETIZATION

Let's consider the following model, with state equation

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t \qquad (1)$$

and observation process

$$dY_t = h(X_t)dt + dV_t \qquad (2)$$

Here, $\{X_t, t \geq 0\}$ and $\{Y_t, t \geq 0\}$ take values in R^m and R^d respectively with X_0 distributed as $p_0(x)dx$. $\{W_t, t \geq 0\}$ and $\{V_t, t \geq 0\}$ are independent Wiener processes with the appropriate dimensions and non-singular covariance matrix I(identity) and R respectively. Let $y_t = \sigma(Y_s, 0 \leq s \leq t)$ denote the σ-field generated by the observations up to time t. The objective is to compute at each time t, the conditional density $p_t(x)$ of the state X_t, given the past available observations y_t, so as to compute

$$E[\phi(X_t) \mid y_t] = c_t \cdot \int_{R^m} \phi(x) p_t(x) dx$$

for any test function ϕ (here c_t is a normalization constant).

Under broad assumptions, the conditional density $p_t(x)$ is the unique solution of the following stochastic PDE, called the Zakai equation [15]

$$dp_t = L^* p_t dt + p_t h^* R^{-1} dY_t \qquad (3)$$

where * is the transpose operator, L is the infinitesimal generator associated with the diffusion process X_t, i.e.

$$L = \frac{1}{2}\sum_{i,j=1}^{m} a_{i,j}(\bullet)\frac{\partial^2}{\partial x_i \partial x_j} + \sum_{i=1}^{m} b_i(\bullet)\frac{\partial^2}{\partial x_i} \qquad (4)$$

with $a = (a_{i,j}) = \sigma\sigma^*$. For basic references about nonlinear filtering and the Zakai equation, see [3], [12] and [13].

Time Discretization

We introduce a uniform partition of the time interval $[0, +\infty)$: $0 = t_0 < t_1 < \cdots < t_n < \cdots$, with constant time step $\delta = t_{n+1} - t_n$. We use the splitting-up algorithm studied in Le Gland [9], which combines an implicit Euler scheme for the prediction step and a Bayes formula based on sampled observations

$$z_n^\delta = \frac{1}{\delta}\left[Y_{t_n} - Y_{t_{n-1}}\right] = \frac{1}{\delta}\int_{t_{n-1}}^{t_n} h(X_s)ds + \frac{1}{\delta}\left[V_{t_n} - V_{t_{n-1}}\right]$$

for the correction step. Following [9], the conditional density p_{t_n} is approximated by p_n^δ, where the transition from p_{n-1}^δ to p_n^δ is divided into the following two steps:

- Prediction step: we solve the following Fokker-Planck equation, between times t_{n-1} and t_n

$$\frac{\partial p_t^n}{\partial t} = L^* p_t^n$$

with initial condition $p_{t_{n-1}}^n = p_{n-1}^\delta$ at time $t = t_{n-1}$. The final value at time $t = t_n$ gives the prior estimate $p_{n-1/2}^\delta = p_{t_n}^n$. The Fokker-Planck equation is further discretized by the Euler implicit scheme, which is unconditionally stable, i.e. we solve

$$\underline{[I - \delta L]^* p_{n-1/2}^\delta = p_{n-1}^\delta}$$

- Correction step: we use the new observation z_n^δ and the Bayes formula, to update the prior estimate $p_{n-1/2}^\delta$, i.e. we compute

$$p_n^\delta = c_n \Psi_n^\delta p_{n-1/2}^\delta$$

where

$$\Psi_n^\delta(x) = \exp\left\{-\frac{1}{2}\delta\left|z_n^\delta - h(x)\right|_R^2\right\}$$

and c_n is a normalization constant and the notation $|\bullet|_R$ denotes the norm in R^d associated with the definite positive matrix R^{-1}, i.e. $|u|_R^2 = u^* R^{-1} u$.

Finite difference on a regular grid

On a given m-dimensional regular grid Ω^h with mesh $h=(h_1,\ldots,h_m)$, we use the upwind finite difference scheme introduced by Kushner [5] (see also Kushner and Dupuis [6]) to approximate the partial differential operator L defined in (4). In this scheme, the first order derivatives are approximated as

$$\frac{d\phi}{dx_i}(x) \cong \begin{cases} \dfrac{\phi(x+e_i h_i)-\phi(x)}{h_i} & \text{if } b_i(x) \geq 0 \\ \dfrac{\phi(x)-\phi(x-e_i h_i)}{h_i} & \text{if } b_i(x) < 0 \end{cases}$$

where e_i denotes the unit vector in the i-th coordinate direction. The central second order derivatives are approximated as

$$\frac{\partial^2\phi}{\partial x_i^2}(x) \cong \frac{\phi(x+e_i h_i)-2\phi(x)+\phi(x-e_i h_i)}{h_i^2}$$

and the mixed second order derivatives as

$$\frac{\partial^2\phi}{\partial x_i \partial x_j} \cong \begin{cases} \dfrac{1}{2h_i}\left[\dfrac{\phi(x+e_i h_i+e_j h_j)-\phi(x+e_i h_i)}{h_j}-\dfrac{\phi(x+e_j h_j)-\phi(x)}{h_j}\right. \\ \left.+\dfrac{\phi(x)-\phi(x-e_j h_j)}{h_j}-\dfrac{\phi(x-e_i h_i)-\phi(x-e_i h_i-e_j h_j)}{h_j}\right], \\ \qquad \text{if } a_{i,j}(x) \geq 0 \\ \dfrac{1}{2h_i}\left[\dfrac{\phi(x+e_i h_i)-\phi(x+e_i h_i-e_j h_j)}{h_j}-\dfrac{\phi(x)-\phi(x-e_j h_j)}{h_j}\right. \\ \left.+\dfrac{\phi(x+e_j h_j)-\phi(x)}{h_j}-\dfrac{\phi(x-e_i h_i+e_j h_j)-\phi(x-e_i h_i)}{h_j}\right], \\ \qquad \text{if } a_{i,j}(x) < 0 \end{cases}$$

As a result, the infinitesimal generator L is approximated as

$$L\phi(x) \cong L^h\phi(x) = \sum_{y \in A^h(x)} L^h(x,y)\phi(y)$$

where for all $x \in \Omega^h$, $A^h(x) \subset N^h(x)$ denotes the set of points in the grid Ω^h which are accessible from x, i.e.

$$A^h(x) = \left\{ x+\varepsilon_i e_i h_i + \varepsilon_j e_j h_j, \text{ for all } \varepsilon_i, \varepsilon_j \in \{0,\pm 1\}, i \neq j \right\}$$

and $N^h(x)$ denotes the set of nearest neighbours of x, including x itself, i.e.

$$N^h(x) = \left\{ x+\varepsilon_1 e_1 h_1 + \ldots + \varepsilon_m e_m h_m, \text{ for all } \varepsilon_1, \ldots \varepsilon_m \in \{0,\pm 1\} \right\}$$

By identifying the coefficients of the matrix L^h, we have for all $x \in \Omega^h$

$$L^h(x,x) = -\sum_{i=1}^{m}\left[\frac{1}{h_i^2}a_{i,i}(x) - \sum_{j,j\neq i}\frac{1}{2h_i h_j}\left|a_{i,j}(x)\right|\right] - \sum_{i=1}^{m}\frac{1}{h_i}\left|b_i(x)\right|$$

$$L^h(x, x \pm e_i h_i) = \frac{1}{2h_i^2}a_{i,i}(x) - \sum_{j,j\neq i}\frac{1}{2h_i h_j}\left|a_{i,j}(x)\right| + \frac{1}{h_i}b_i^{\pm}(x)$$

$$L^h\left(x, x+e_i h_i \pm e_j h_j\right) = \frac{1}{2h_i h_j}a_{i,j}^{\pm}(x)$$

$$L^h\left(x, x-e_i h_i \mp e_j h_j\right) = \frac{1}{2h_i h_j}a_{i,j}^{\pm}(x)$$

$$L^h(x,y) = 0, \quad \text{otherwise}$$

One can see [5] for the definition of finite difference approximation on the boundary. In any case, we can check that

$$\forall x \in \overline{\Omega}^h, \sum_{y \in A^h(x)} L^h(x,y) = 0$$

and under the additional assumption

$$\forall x \in R^m, \frac{1}{h_j^2}a_{i,i}(x) - \sum\frac{1}{2h_i h_j}\left|a_{i,j}(x)\right| \geq 0 \qquad (5)$$

the matrix L^h satisfies

$$\forall x \in \overline{\Omega}^h, L^h(x,x) \leq 0 \text{ and } L^h(x,y) \geq 0 \text{ for each } y \in A^h(x), y \neq x$$

Therefore, under the additional assumption (), the finite difference matrix L^h obtained by this method can be interpreted as the infinitesimal generator of a pure jump Markov process taking values in the discretization grid Ω^h. In addition, the matrix $\left[I - \delta L^h\right]$ is a M-matrix. As a consequence, the resulting approximation to the Fokker-Planck equation will automatically be a discrete probability distribution, i.e. non-negative everywhere, and adding up to one.

Finite difference on a composite grid

For simplicity, we consider the situation with two meshes: a fine mesh $h=(h_1,\ldots,h_m)$, and a coarse mesh $H=2h$. We assume that a composite grid Ω^h is given as

$$\Omega^h = \Omega^H \cup \Omega^h_{loc} \quad \text{where } \Omega^h_{loc} = \bigcup_{x \in \Omega^H, \text{refined}} N^h(x)$$

This defines a first partition of the composite grid Ω^h into the set Ω^h_{loc} of fine grid points, and the set $\Omega^H_{loc} = \Omega^h \setminus \Omega^h_{loc}$ of coarse grid points: see Figure 1.

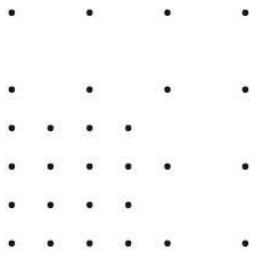

Figure 1 – A composite grid of two levels

A fine grid point $x \in \Omega^h_{loc}$ is said to be *regular* if $A^h(x) \subset \Omega^h_{loc}$. Similarly, a coarse grid point $x \in \Omega^H_{loc}$ is said to be *regular* if $A^h(x) \cap \Omega^h_{loc} = \varnothing$.

This defines a second partition of the composite grid Ω^h into the set Ω^h_C of regular coarse grid points, the set Ω^h_F of regular fine grid points, and the set $\Omega^h_I = \Omega^h \setminus \Omega^h_R$ of *interface* points, where $\Omega^h_R = \Omega^h_C \cup \Omega^h_F$: see Figure 2.

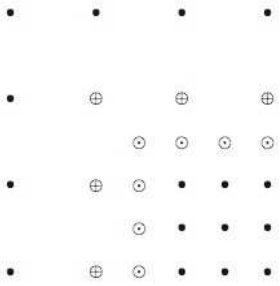

Figure 2 Regular point $(\bullet)$: Ω^h_C and Ω^h_F, coarse and fine interface point $(\oplus, \ominus)$: Ω^h_I

For any point $x \in \Omega^h$, we need to specify the set $A^h(x)$ of points in the composite grid Ω^h which are *accessible* from x, and to specify the corresponding finite difference coefficients $L^h(x,y)$, $y \in A^h(x)$.

At a regular point $x \in \Omega^h_R = \Omega^h_C \cup \Omega^h_F$, we use the original finite difference scheme. At an interface point Ω^h_I, a special technique is used: (i) First, as depicted in Figures 3 and 5, we consider the set $A^h(x)$ of points in the grid Ω^h which are accessible from x, by adding some slave points (+) if necessary, and we define the corresponding finite difference coefficients $L^h(x,y)$, $y \in A^h(x)$, using the expressions given in the section above. (ii) Then, we redistribute the finite difference coefficients of each slave point to points that actually exist in the composite grid, by linear interpolation.

We get in the end a composite scheme:

$$L^h\phi(x) = \sum_{y \in A^h(x)} L^h(x,y)\phi(y)$$

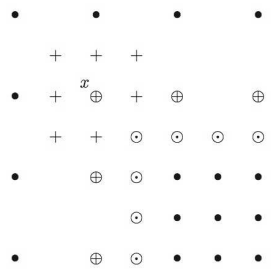

Figure 3 – Coarse interface points $(\oplus)$ and their slave points $(+)$

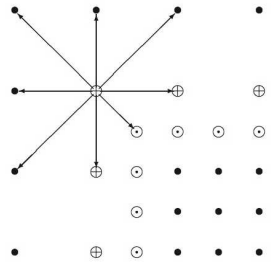

Figure 4 – Coarse interface points $(\oplus)$ and their pointers

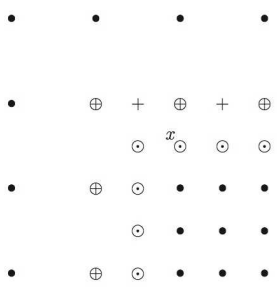

Figure 5 – Fine interface points $(\ominus)$ and their slave points $(+)$

With this technique, we can prove that the resulting finite difference matrix $L^{\underline{h}}$ on the composite grid $\Omega^{\underline{h}}$, is such that the matrix $\left[I - \delta L^{\underline{h}} \right]$ is a M -matrix.

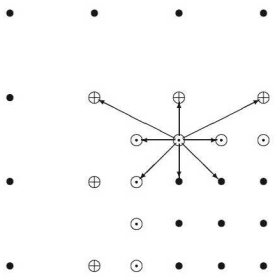

Figure 6 – Fine interface points ( ) and their pointers

3. ALGORITHM FOR THE FILTERING PROBLEM

For simplicity, we present our algorithm in the simple case of two meshes: a fine mesh $h = (h_1, \ldots, h_m)$, and a coarse mesh $H = 2h$. To obtain our approximation $p_n^{\delta,\underline{h}}$ at time t_n, we proceed in three steps.

- We first consider the following linear system on the coarse grid Ω^H

$$\left[I - \delta L^H \right]^* p_{n-1/2}^{\delta,H} = p_{n-1}^{\delta,H}$$

The right hand side vector $p_{n-1}^{\delta,H}$ is just the restriction to Ω^H of the fine grid approximation $p_{n-1}^{\delta,h}$ obtained at the previous time t_{n-1}. To solve this linear system, we use the BCG (BiConjugate Gradient) method. Based on this rough approximation $p_{n-1/2}^{\delta,H}$ of the solution $p_{n-1/2}$, we have then to decide where to refine the grid. From a theoretical point of view, the determination of the refinement region is a difficult task. Here, we propose a widely used heuristic a posteriori. criterion, based on the magnitude of the Zakai equation. Using a simple threshold decision rule, we decide whether this point is going to be refined or not. As a result, we obtain a composite grid

$$\Omega^{\underline{h}} = \Omega^H \cup \Omega_{loc}^h \quad \text{where } \Omega_{loc}^h = \bigcup_{x \in \Omega^H, \text{refined}} N^h(x)$$

Notice that the nested iteration of the multigrid method offers a numerically natural way to do this step.

- In the second step we construct a new linear system

$$\left[I - \delta L^{\underline{h}} \right]^* p_{n-1/2}^{\delta,\underline{h}} = p_{n-1}^{\delta,h}$$

on the composite grid. The matrix $L^{\underline{h}}$ is defined as in section above. The right hand side vector $p_{n-1}^{\delta,\underline{h}}$ is again the restriction to $\Omega^{\underline{h}}$ of the fine grid approximation $p_{n-1}^{\delta,h}$ obtained at the previous time t_{n-1}. To solve this system, we use the FAC method .

- The last step consists of the correction step

$$p_n^{\delta,h} = c_n \Psi_n^{\delta} I_{\underline{h}}^h p_{n-1/2}^{\delta,\underline{h}}$$

on the global fine grid Ω^h. Here c_n is a normalization constant, to make sure that the vector $p_n^{\delta,h}$ is a discrete probability distribution (i.e. non-negative everywhere, and adding up to one).

FAC method

The FAC method (fast adaptive composite multigrid method) introduced in McCormick [10] is a natural extension of the conventional multigrid methods to problems discretized on composite grids. For simplicity, we give here a two level algorithm for solution of equation above. i.e.

$$M^{\underline{h}} u^{\underline{h}} = f^{\underline{h}}$$

with $M^{\underline{h}} = \left[I - \delta L^{\underline{h}} \right]^*$ and $f^{\underline{h}} = p_{n-1}^{\delta,\underline{h}} = I_h^{\underline{h}} p_{n-1}^{\delta,h}$

begin	
$f^{2h} = I_{\underline{h}}^{2h}(f^{\underline{h}} - M^{\underline{h}} u^{\underline{h}})$	residual restriction to coarse grid
$u^{2h} = \left(M^{2h}\right)^{-1} f^{2h}$	solution of coarse grid problem
$u^{\underline{h}} = u^{\underline{h}} + I_{2h}^{\underline{h}} u^{2h}$	global correction
$f_{loc}^h = I_{\underline{h}}^{h,loc}(f^{\underline{h}} - M^{\underline{h}} u^{\underline{h}})$	residual restriction to local grid
$u_{loc}^h = (M_{loc}^h)^{-1} f_{loc}^h$	solution of local fine grid problem
$u^{\underline{h}} = u^{\underline{h}} + I_{\underline{h},loc}^{\underline{h}} u_{loc}^h$	local correction
end	

Figure 7 – Two-level FAC algorithm

The grid transfer operators $I_{\underline{h}}^h$, $I_h^{\underline{h}}$, $I_{\underline{h}}^{2h}$, $I_{2h}^{\underline{h}}$, $I_{\underline{h}}^{loc}$ and $I_{\underline{h},loc}^{\underline{h}}$ operators are defined in an obvious way. A full multigrid algorithm (FMG) is used in our approach. At coarsest level we replace the exact solution by BCG

method. As we have already pointed out, the FMG approach gives us a cheap way to find the refined regions.

From the point of view of implementation, solving a 4-dimension PDE on a composite grid numerically is quite difficult Special data structure and dynamic memory allocation have to be used. For the implementation of the Zakai equation, we chose the C++ language because it provides an efficient realization of object oriented programming techniques. This is discussed in [2] in the case of two dimensions. The idea is then extended for four dimensions, and this is our main contribution.

4. BEARINGS-ONLY TRACKING

The problem of bearings-only tracking arises in a variety of important practical applications, for example tracking using a passive sonar. Here the target state is a four-dimensional vector:

$$X_t = (x_t, y_{t,}, u_t, v_t)^T$$

where (x, y) and (u, v) are the position and velocity components, respectively. We consider a classical CV model presented in [1,page 86], where after the discretization of equation (1), we have

$$X_{k+1} = FX_k + \Gamma\gamma_k \qquad (6)$$

where γ_k a scalar zero mean white noise sequence with $E(\gamma_k\gamma_j) = \sigma_\gamma{}^2\delta_{k,j}$ and the covariance matrix of process $V_k = \Gamma\gamma_k$ is defined by

$$Q = \sigma_\gamma^2\Gamma\Gamma^* = \sigma_\gamma^2\begin{pmatrix} \frac{\delta^4}{4} & 0 & \frac{\delta^3}{2} & 0 \\ 0 & \frac{\delta^4}{4} & 0 & \frac{\delta^3}{2} \\ \frac{\delta^3}{2} & 0 & \delta^2 & 0 \\ 0 & \frac{\delta^3}{2} & 0 & \delta^2 \end{pmatrix}$$

So we have $a = \sigma_\gamma^2\Gamma\Gamma^*$. The vector $b(\bullet)$, which is the coefficient before $\frac{\partial}{\partial x_i}$ is,

$$b(x, y, u, v) = (u, v, 0, 0)$$

The observation process is

$$y_k = h(X_k) + v_k$$

with

$$h(X_k) = \tan^{-1}\left(\frac{x_k - x_k^o}{y_k - y_k^o}\right) \qquad (7)$$

and v_k : $N(0, \sigma_v^2)$. The classical UKF equations ([4]) are then applied to (7), whereas the standard Kalman filter equations are applied to (6).

5. NUMERICAL RESULTS

In this section, we present the numerical results of a scenario. The target-observer configuration for this case is shown in Figure 8. The target, whose state vector is initialized with small error (10%) in position and null velocity, moves with an initial velocity of 0.4 m/s, and a initial heading of -270^0; its trajectory is disturbed by a white noise acceleration of magnitude $\sigma_\gamma = 0.005$ m/s^2. The ownship executes a circle maneuver disturbed by a white noise acceleration of the same magnitude; its initial position is (0,0) and its velocity is 2 m/s with initial heading of 90^0 and turn rate of 0.3 deg/s. Bearing measurements with accuracy $\sigma_v = 1^0$ are received every $\delta = 1$ second for an observation period of 800 seconds.

We will present the numerical results in the following figures. Figure 8 shows the Zakai filter in target-observer plan. Figure 9 gives the estimated target kinematics features deduced from the state vector: relative distance (between ownship and target), heading and course. Same estimation is also calculated by the UKF filter with Cartesian representation (for more details, see [8]).

Figures 10, 11 and 12 give the solutions of the Zakai equation at instants 40 s, 160 s and 800 s. Each figure has four subplots; at the top, we give marginal density in (x, y) and (u, v) fields with the true position indicated by a cross mark; at the bottom we present their corresponding grids obtained by a projection of 4D composite grid onto (x, y) and (u, v) plans. Of course this projection is done for illustration purposes of our local refinement grid method, and need not to be done in practice.

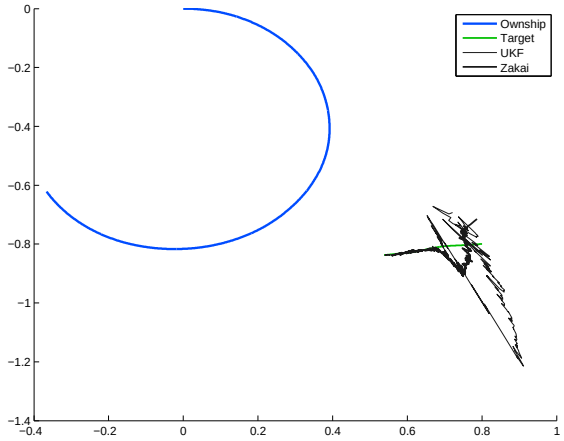

Figure 8 – Scenario, Zakai and UKF estimated trajectories

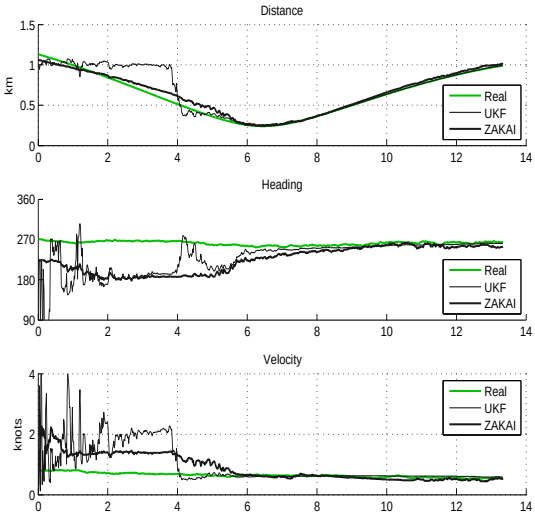

Figure 9 – Kinematics features

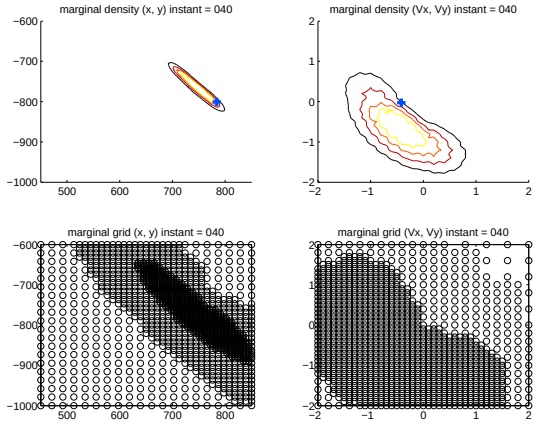

Figure 10 – Zakai solution at t=40 s, marginal density and corresponding composite grid

Three levels are used in this example. The coarsest grid is a $20 \times 20 \times 11 \times 11$ regular grid so that the finest level has a resolution of $77 \times 77 \times 41 \times 41$, nearly 10 million grid points. In comparison, there are about 300 thousand points in our composite grid. This test takes 280 minutes with an opteron 2.4 Ghz CPU, consuming 900 Mo Ram. To compare the time performance of our method, we have run this code with a one level $77 \times 77 \times 41 \times 41$ regular grid: the CPU time for this problem took 134.4 hours, and needed 18 Go of memory space.

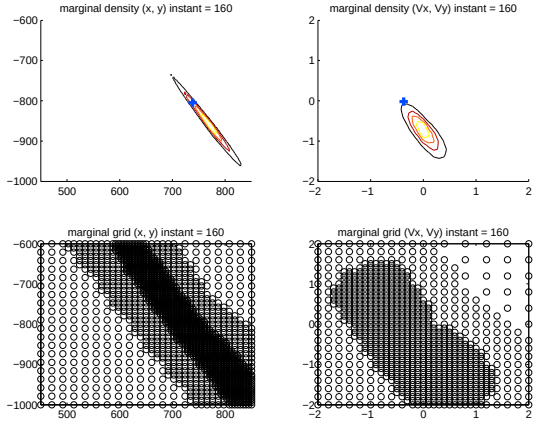

Figure 11 – Zakai solution at t=160 s, marginal density and corresponding composite grid

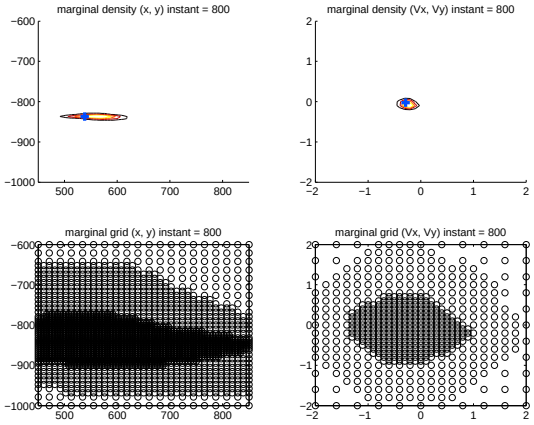

Figure 12 – Zakai solution at t=800 s, marginal density and corresponding composite grid

6. CONCLUSION

We have implemented for the first time an algorithm to solve the Zakai equation of a four dimensional stochastic system with local refinement. Our approach is based upon a local refinement method where the state space is dynamically refined when the density is significant, and this allows for an important time reduction.

7. REFERENCES

[1] Y. Bar-Shalom and T.E. Fortmann, "Tracking and Data Association", Academic Press, 1988

[2] Z. Cai, F. Le Gland and H. Zhang, "An Adaptive Local Grid Refinement Method for Nonlinear Filtering", RR-2679 Rapport de recherché de l'INRIA-Rennes, 1995.

[3] A.H. Jazwinski, "Stochastic Processes and Filtering Theory", volume 64 of *Mathematics in Science and Engineering*. Academic Press, New York, 1970

[4] S. J. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems", *Proc. of AeroSense: the 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Control*, 1997.

[5] J. Kushner, "Probability Methods for Approximations in Stochastic Control and for Elliptic Equations", volume 129 of *Mathematics in Science and Engineering*, EAcademic Press, New York, 1977.

[6] H.J. Kushner and P. Dupuis, "Numerical Methods for Stochastic Control Problems in Continuous Time", volume 24 of *Applications of Mathemathics*. Springer Verlag, New York, 1992.

[7] D. Laneuville and H. Vignal, "Grid based Target Motion Analysis", 2007 IEEE Aerospace Conference Proceedings, March 2007.

[8] D. Laneuville and C. Jauffret, "Recursive Bearings-only TMA via Unscented Kalman Filter : Cartesian vs. Modified Polar Coordinates", 2008 IEEE Aerospace Conference Proceedings, March 2008.

[9] F. Le Gland, "Time discretization of nonlinear filtering equations". *Proceedings of the 28th IEEE Conference on Decision and Control*, pages 2601--2606. IEEE, December 1989.

[10] S.F. Mccormick, "Multilevel Adaptive Methods for Partial Differential Equations", volume 6 of *Frontiers in Applied Mathematics*, SIAM, Philadelphia, 1989

[11] G. Pagès and H. Pham, "Optimal quantization methods for nonlinear filtering with discrete-time observations, *Bernouilli,* 11(5), pages 893-932, 2005

[12] E. Pardoux, "Filtrage non linéaire et équations aux dérivées partielles stochastiques associées" In P.L. Hennequin, editor, *Ecole d'Eté de Probabilités de Saint-Flour XIX* - 1989}, volume 1464 of *Lecture Notes in Mathematics*, pages 67--163. Springer Verlag, Berlin, 1991.

[13] B. Rozovskii "Stochastic Evolution Systems", volume 35 of *Mathematics and its Applications* (Soviet Series). Kluwer Academic Publishers, Dordrecht, 1990

[14] R. Van der Merwe, A. Doucet and N. de Freitas and E. Wan, "The Unscented Particle Filter", Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department, 2000

[15] M. Zakai. "On the optimal filtering of diffusion processes", *Zeitschrift fur Wahrscheinlichkeits theorie und verwandte Gebiete*, 11(3):230--243, 1969

BIOGRAPHY

Huilong Zhang, born in 1964, China, received the B.S. degree in applied mathematics from Northwestern Polytechnical University (China) in 1986 and a Ph.D in Applied Mathematics from University de Provence Aix Marseille 1 in 1992. He is associated professor at University Bordeaux 1 since 1993. His research interests include nonlinear filtering and dynamic reliability.

Dann Laneuville, born in 1964, France, received the diplôme d'ingénieur from Ecole Supérieure d'Electricité (Supélec) in 1987 and a Ph.D. in automatic and signal processing from the Paris XI University in 1998. Before joining DCNS in 2003, where he is working on Target Motion Analysis problems and Multi-Platform Tracking Process, he has been with MBDA for fifteen years where he has developed researches in Multi Sensor Maneuvering Target Tracking, Data Fusion and Guidance.