

# A fast atmospheric turbulent parameters estimation using particle filtering. Application to LIDAR observations.

**Suzat Florian, Baehr Christophe, Dabas Alain**

Meteo France, 42 av. Gaspard Coriolis 31057 Toulouse cedex 1, France

E-mail: [florian.suzat@meteo.fr](mailto:florian.suzat@meteo.fr)

## **Abstract.**

Estimating fast turbulence fluctuations in the boundary layer of the atmosphere, using remote detection instrument is an important scientific issue. Doppler LIDAR, is typically used to get this kind of information because it can make fast, distant, precise, and non-intrusive measurements of the wind field by giving the radial component in any direction. The objective of those measurements is to evaluate as precisely as possible the wind structure using the partial wind information provided, in order to estimate turbulent parameters. The approach presented in this paper, consist in coupling the remote detection system and a stochastic Lagrangian model of the atmosphere. The fluid is represented by a set of interacting particles, evolving according to an evolution system based on S.B Pope work. Data provided by the instrument are assimilated in real time in the model using a particle filtering algorithm. The purpose is to locally correct the properties of particles using measurements, to fit the real fluid observed. A precise real time estimation of the wind field, allows then to estimate turbulent parameters. The methodology has produced convincing results on simulated Doppler LIDAR measurements, in tree-dimensional modeling.

## **1. Introduction**

In various activities, scientists use numerical computation in order to predict the evolution of physical systems. The initialization is a critical problem to ensure that the numerical system keeps modeling reality. In many cases, optimizing the model state compared to the true system state is very difficult. Practically, this estimation consist in using measurement instruments, to guess as precisely as possible the state of the real system at a given time. This process is called data assimilation and plays a very important role in the quality of numerical model predictions.

The more classical assimilation methods are based on Kalman Filters. They are widely used in different scientific fields. Nevertheless, they present some disadvantages. The use of the Kalman filter requires the random errors of the model and the observations to be Gaussian which is a strong hypothesis that is not always verified in complex systems. Moreover, this assimilation technique is slow, in other words, it is not very efficient when the system studied has fast and important variations. Lastly, especially on high dimensional systems, such as in geophysical sciences, Kalman Filter becomes numerically very expensive.

Innovative methods, based on applied probability theory, has been pointed out by researchers such as P. Del Moral (Del Moral (2004)). These methods are called particles filters. If they

are suited efficiently, particles filtering algorithms do not present the disadvantages explained before example, and therefore have strong potentiality. They are often used in robotic, telecommunication, but not very much in meteorology. Atmospheric surface layer turbulence phenomena, are typically fast evolving, complex and non-linear, that is why particles filters seems to be suitable, event if they are not yet used for operational purposes.

Doppler LIDAR, is an interesting instrument to estimate turbulence as it can make distant radial wind speed measurements in all direction with high frequency and high accuracy. It emits a pulse of wavelength  $\lambda_e$  along a direction and analyses the signal diffused by tracers encountered by the light beam. In fact, the wavelength  $\lambda_e$  is chosen close to typical surface layer aerosols (  $\sim 1 \mu\text{m}$  ), such that they radiate, like oscillating electromagnetic dipole,  $\lambda_r$  wavelength. The reflected signal is analyzed by the instrument, such that  $\Delta\lambda_d = \lambda_e - \lambda_r$  the Doppler difference, leads to the radial velocity component of the wind field. Because the light is emitted by pulses, and using signal processing tools, a Doppler LIDAR can give radial wind speed in several points in a given direction. The advantages are high level of precision and accuracy, and the fact that the LIDAR can target every spatial directions.

The distance between two consecutive measurement points is the spatial resolution of the LIDAR. In general, the spatial resolution is around  $50\text{m}$  and the range of a LIDAR shoot reach several kilometers in good meteorological conditions. The sampling frequency is close to  $1\text{Hz}$  and the accuracy is close to  $1\text{cm.s}^{-1}$  for the radial velocity.

The standard method to estimate turbulence with this instrument consists in statistic computation during a scan of the LIDAR (see for example Frehlich *et al.* (2006)). The approach presented here is significantly different. The method consist in coupling a turbulent model and LIDAR measurements using particle filtering tools. Radial velocities measurements in several direction provided by the LIDAR are assimilated in the model with its highest frequency. Thus the model provides a real time estimation of the wind field, and by extension, of turbulent parameter fields such as turbulent kinetic energy.

This paper describes how particle filtering can be used efficiently with a simplified turbulence model. It presents a method to apply particle filtering formalism in a particular system. But it also provides ideas to apply it in more general situation. Theoretical aspects and others applications of this technology can be found in Del Moral (2004), or Baehr (2008).

The next section is a description of the turbulence model considered. It is a stochastic Lagrangian model inspired by the work of S.B Pope. Section 3 is an overview of non linear filtering principles and an algorithmic description of a localized particle filtering process. In the last section, the complete system behavior is studied on an experiment. The objective is to demonstrate that the system constructed is able to estimate correctly the wind field characteristics (strength, direction, and turbulence) and to react quickly in case of strong variations.

## 2. A stochastic Lagrangian model for turbulent fluids

This section is a presentation of the turbulence model considered. Let  $\mathcal{X}$  and  $\mathcal{V}$  be general Euclidean spaces.  $\mathcal{X}$  is called the position space.  $\mathcal{V}$  is the value or parameter space. In this case, the model only takes velocities into account, then  $\mathcal{V}$  is the space of velocities. But, most of the following definitions can be adapted for more complex model (for example taking account temperature or humidity). Let be  $\mathcal{Z} = \mathcal{X} \times \mathcal{V}$ .

In this paper, the experience described has been performed in a three-dimensional frame. Then  $\mathcal{X}$  is a subspace of  $\mathbb{R}^{d_x}$  and  $\mathcal{V}$  is a subset of  $\mathbb{R}^{d_v}$ , where  $d_x$  and  $d_v = 3$ . In the following sections, the variable  $x$  refers to position ( in  $\mathcal{X}$  ),  $v$  to velocity ( in  $\mathcal{V}$  ), and  $z = (x, v)$  to an element of  $\mathcal{Z}$ .

### 2.1. Time continuous system

The initial equations used to form our dynamical model are the system (1) & (2). It is a standard stochastic differential equation, representing the evolution of the position and the velocity of a fluid particle put inside a turbulent flow. The solution of the system is not a deterministic function but a random process.

$$dx_t = v_t dt \quad (1)$$

$$dv_t = a_t dt - \frac{1}{T_t^L} (v_t - \langle v_t \rangle) dt + \sigma_t dw_t \quad (2)$$

where  $a_t$  is an acceleration term,  $\langle v_t \rangle$  represents the spatial Eulerian average velocity and  $T_t^L$  the Lagrangian time of the model.  $w_t$  is a standard Brownian motion ( a Gaussian random process) and  $\sigma_t$  is a covariance term.

The construction of a turbulence equation system based on these equations consists in finding the expression of the terms  $a_t$ ,  $\sigma_t$  and  $T_t^L$  such that the random process solution behave as turbulent atmosphere flow. S.B Pope proposes a construction. Here, his model is simplified and adapted for surface layer turbulence. To be consistent with Kolmogorov K41 theory, Pope proposes to model  $\sigma_t$  by (3):  $C_0$  is the Kolmogorov constant and  $\mathcal{E}_t$  is the dissipation rate tensor. Using standard turbulence relation (Pope (2000)), a way to model the Lagrangian time  $T_t^L$  is (5), where  $\mathcal{K}_t$  is the turbulent kinetic energy (TKE). With this form,  $\frac{1}{T_t^L}$  acts as a isotropic return term. The TKE can be deduced from the velocity field by the formula (4). The operator  $\langle . \rangle$  is for the Eulerian average. The turbulent energy is then equal to the half of the variance of the Eulerian velocity. The term acceleration  $a_t$ , is given by (6).  $\rho$  is the mass density of the fluid and  $\nabla_x \langle P \rangle$  the gradient of the spatial mean pressure. In this work, this pressure term is voluntary ignored. In fact, this model is used in a filter, then this gradient pressure term will be learned from the observation by the update step, and the incompressibility assumption will be restored.

$$\sigma_t = \sqrt{C_0 \mathcal{E}_t} \quad (3) \quad \frac{1}{T_t^L} = \left(\frac{1}{2} + \frac{3}{4}C_0\right) \frac{\mathcal{E}_t}{\mathcal{K}_t} \quad (5)$$

$$\mathcal{K}_t = \frac{1}{2} \langle (v_t - \langle v_t \rangle)^2 \rangle \quad (4) \quad a_t = \frac{1}{\rho} \nabla_x \langle P \rangle \quad (6)$$

$\mathcal{E}_t$  is a term not simply computable. That is why a classical  $\mathcal{K} - \mathcal{E}$  parametrization, which links the dissipation rate and the TKE, is used. The term  $\mathcal{E}_t$  is expressed as a function of  $\mathcal{K}_t$  (8), where  $C_1$  is a constant, and  $L_{scale}$  is a turbulence scale length. To sum up, the continuous model equation system used in this work is the system (1) & (2) with

$$\sigma_t = \sqrt{\frac{C_0 C_1}{L_{scale}} \mathcal{K}_t^{\frac{2}{3}}} \quad (7) \quad \frac{1}{T_t^L} = \frac{(\frac{1}{2} + \frac{3}{4}C_0) C_1}{L_{scale}} \mathcal{K}_t^{-\frac{1}{3}} \quad (9)$$

$$\mathcal{E}_t = C_1 \frac{\mathcal{K}_t^{\frac{2}{3}}}{L_{scale}} \quad (8) \quad a_t = 0 \quad (10)$$

The advantage is that this equation system is closed by the Eulerian velocity variance, each term is constant, or computable from the velocity field.

## 2.2. Discrete interacting particle system

It is necessary to deduce a time and space discrete model from the previous one. Let note  $\Delta t$  a small real number, the elementary time unit. Now, each quantity is denoted with the integer  $n \in \mathbb{N}$ . Instants considered are  $t_0, t_1, \dots, t_n$  where  $t_n = n\Delta t$ .  $Q_n$  refers to the variable  $Q$  at time  $n\Delta t$ . A Lagrangian representation for the flow is chosen. At time  $t_n$ , the fluid is represented with a set of  $N$  particles belonging to the space  $\mathcal{Z}$ .  $p_n^i = (x_n^i, v_n^i) \in \mathcal{Z}$  is the  $i^{\text{th}}$  particle.  $x_n^i$  is the position,  $v_n^i$  the velocity of the particle. Let us formalize the model state in terms of particle network denoted  $R_n^N$ .

$$R_n^N = \sum_{i=1}^N \delta_{(x_n^i, v_n^i)}^{\mathcal{Z}} \quad (11)$$

where  $\delta^{\mathcal{Z}}$  represents the standard Dirac distribution on  $\mathcal{Z}$ .  $R_n^N$  is a distribution on  $\mathcal{Z}$  containing all the informations of the model at time  $t_n$ .

With this notation, the model evolution process consists in an transformation of  $R_n^N$  into  $R_{n+1}^N$ . The key point is to set an expression to  $\langle \cdot \rangle$  terms in this discrete frame. In other words, it is necessary to define for all  $x \in E$ ,  $\langle V_n^N(x) \rangle$ , the mean velocity at point  $x$  at time  $t_n$ , from the particle network  $R_n^N$ . The goal is to spatially interpolate the discrete distribution  $R_n^N$  to get something similar to the mean and the variance of velocity at any point  $x \in E$ . Let define the standard Gaussian spatial interpolation function of length  $\lambda$ , noted  $g^\lambda$  by:

$$\begin{aligned} g^\lambda : \mathcal{X} &\longrightarrow \mathbb{R}_+^* \\ x &\longrightarrow g^\lambda(x) = \exp\left[-\left(\frac{\|x\|_{\mathcal{X}}}{\lambda}\right)^2\right] \end{aligned}$$

$\|\cdot\|_{\mathcal{X}}$  represents the standard Euclidean norm on  $\mathcal{X}$ . With this function, an interpolation of the velocity for any point  $x$  can be formalized. The choice of the spatial interpolation function is arbitrary. This one is one of the simplest but other functions  $g$  are possible. Therefore, let define:

$$\langle V_n^N(x) \rangle \stackrel{\text{def}}{=} \frac{\int_{z \in \mathcal{Z}} R_n^N(z) g^\lambda(x - \Pi_{\mathcal{X}}(z)) \Pi_{\mathcal{V}}(z) dz}{\int_{z \in \mathcal{Z}} R_n^N(z) g^\lambda(x - \Pi_{\mathcal{X}}(z)) dz} \quad (12)$$

$$= \frac{\sum_{j=1}^N g^\lambda(x - x_n^j) v_n^j}{\sum_{j=1}^N g^\lambda(x - x_n^j)} \quad (13)$$

where  $\Pi_{\mathcal{X}}$  (resp.  $\Pi_{\mathcal{V}}$ ) represents the canonical projector from  $\mathcal{Z}$  to  $\mathcal{X}$  (resp.  $\mathcal{V}$ ). For  $z = (x, v) \in \mathcal{Z}$ ,  $\Pi_{\mathcal{X}}(z) = x$  and  $\Pi_{\mathcal{V}}(z) = v$ . The equation (12) is easiest to understand in its discrete form (13). The equality between both comes from elementary distribution theory.  $\langle V_n^N(x) \rangle$  can be viewed as the barycenter of the velocity of network  $R_n^N$  at point  $x$ , pondered by the function  $g^\lambda$ . Because the mean is done on a neighborhood of  $x$ ,  $\langle V_n^N(x) \rangle$  is an approximation in discrete time of the spatial Eulerian mean  $\langle v_t \rangle$  introduced before. In the same way, the TKE can be spatially discretized and approximated at any location  $x$  by:

$$\mathcal{K}_n^N(x) \stackrel{\text{def}}{=} \frac{1}{2} \frac{\sum_{j=1}^N g(x - x_n^j) (v_n^j - \langle V_n^N(x) \rangle)^2}{\sum_{j=1}^N g(x - x_n^j)} \quad (14)$$

With formulas (13) & (14), it is possible to compute the spatial average velocity and the TKE for any position  $x$  from the particle network  $R_n^N$ . The Brownian motion  $dw_t$  is time-discretised using the fact that:

$$dw_t \approx w_{t+\Delta t} - w_t = \xi \sqrt{\Delta t} \quad (15)$$

Where  $\xi$  is a realization of a standard normal Gaussian variable.

Then, using the standard Euler time discretization scheme, the discrete evolution model system at time  $t_n$  can be written for any  $i \in \{1, \dots, N\}$  :

$$x_{n+1}^i = x_n^i + v_n^i \Delta t \quad (16)$$

$$v_{n+1}^i = v_n^i - \frac{1}{T_n^L(x_n^i)} [v_n^i - \langle V_n^N(x_n^i) \rangle] \Delta t + \sigma_n(x_n^i) \xi_n^i \sqrt{\Delta t} \quad (17)$$

$$\frac{1}{T_n^L(x_n^i)} = \frac{(\frac{1}{2} + \frac{3}{4}C_0)C_1}{L_{scale}} \mathcal{K}_n^N(x_n^i)^{-\frac{1}{3}} \quad (18)$$

$$\sigma_n(x_n^i) = \sqrt{\frac{C_0 C_1}{L_{scale}} \mathcal{K}_n^N(x_n^i)^{\frac{2}{3}}} \quad (19)$$

$$\mathcal{K}_n^N(x_n^i) = \frac{1}{2} \frac{\sum_{j=1}^N g(x_n^i - x_n^j) (v_n^j - \langle V_n^N(x_n^i) \rangle)^2}{\sum_{j=1}^N g(x_n^i - x_n^j)} \quad (20)$$

where  $\xi_n^i$  are independent standard normal Gaussian variables. The evolution algorithm consists in calculating the values  $\langle V_n^N(x_n^i) \rangle$  and  $\mathcal{K}_n^N(x_n^i)$  for each particle  $(x_n^i, v_n^i)$  using all the other particles of the network  $R_n^N$ , and then deducing  $(x_{n+1}^i, v_{n+1}^i)$  for every particles to get  $R_{n+1}^N$ .

### 2.3. Discussion about this model

It is important to keep in mind that our system is an interacting particle system. In fact, interactions between particles are contained in terms containing mean operators such as  $\langle V_n^N(x) \rangle$  and  $\mathcal{K}_n^N(x)$  because the computation for a particle depends on all the others. This fact reveals a very important point of this discretization. The choice of the spatial interpolation function  $g$  has a strong influence on the model. For example, in the case of  $g^\lambda$ , high values of  $\lambda$  greatly enlarge the range of the interaction, whereas small values create week interactions between particles which are far one from another. Therefore, this choice has to be done very carefully depending the scale of the turbulent phenomenon wanted to be modeled. In the same way, the choice of the constants  $C_1$  and  $L_{scale}$ , strongly impact on the model behavior.

Even if the model presented is simplified, it provides an interesting frame for experiments on particles filtering algorithm, which is the main point of this work.

## 3. Non linear filtering

This section shows how a non-linear data assimilation method works, and how particle filtering tools can be used on a simple example. Usually, simulation processes with data assimilation consist is an succession of two steps which can be symbolized by :

$$R_n^N \xrightarrow{\text{update}} \widehat{R}_n^N \xrightarrow{\text{evolution}} R_{n+1}^N$$

The evolution step consists in computing the particle network  $R_n^N$  into  $R_{n+1}^N$ . This step is explained in section 2. The update step (selection, in particle filtering vocabulary), consists in a transformation of  $R_n^N$  (called *a priori* particle network) into  $\widehat{R}_n^N$  (*a posteriori*) using measurements made on the real fluid. The goal is to make the numerical model better fit the reality.

### 3.1. An example of particle filtering algorithm

In this case, the measurement instrument is a Doppler LIDAR. The principle is then to re-sample particles of the network to make them coherent with the radial velocity informations provided by the LIDAR, and then to force the particle network to estimate the true fluid.

Particle filtering methods are based on applied probability theory. The precise notations, formalism and justifications can be found in P. Del Moral references. Here, only the basic ideas are presented, with a simplified formalism.

The operation described in this part is the transformation of the particle network from the *a priori* network  $R_n^N$  into  $\widehat{R_n^N}$ , the *a posteriori* one. Let us consider that the measurement made by the LIDAR at time  $t_n$ , noted  $\mathcal{L}_n$ , consists in a set of  $K$  couples (position, radial velocity)  $(y_n^1, r_n^1) \dots (y_n^K, r_n^K)$ . We set  $u_n \in \mathcal{X}$  the direction targeted by the LIDAR.  $u_n$  is chosen unitary. With the LIDAR measurements description given in introduction, let us consider that  $y_n^k = kl_r u_n$ ,  $l_r$  represent the spatial resolution of the LIDAR shoot.

The first thing to do is to extract from  $R_n^N$  the set of particle affected by the shoot  $\mathcal{L}_n$ . For each couple  $(y_n^k, r_n^k)$ , it is necessary to build  $\mathcal{C}_n^k$  the set of particles concerned. Without detailing more, once may consider that a particle  $p_n^i = (x_n^i, v_n^i)$  is in  $\mathcal{C}_n^k$  if  $x_n^i$  belongs to a certain neighborhood of  $y_n^k$ . For example let consider that:

$$p_n^i \in \mathcal{C}_n^k \text{ if } \|x_n^i - y_n^k\|_\infty < \frac{l_r}{2} \quad (21)$$

$\|x\|_\infty$  is equal to  $\max(|x_x|, |x_y|, |x_z|)$ . The choice of the neighborhood is of course not unique. In fact the term *localized* concerning particle filtering indicates that the selection step is done in a spatially limited domain. From this point the position of the particle will not impact anymore during the process.

The second part consists in giving a potential to the particles  $(p_n^{(k,1)}, \dots, p_n^{(k,c_n^k)})$  of  $\mathcal{C}_n^k$  (where  $c_n^k$  denotes the cardinal of  $\mathcal{C}_n^k$ ). The more a particle is coherent with a measurement, the higher its potential is. In this example, because the LIDAR measurement concern radial velocities, the potential of each particle  $p_n^{(k,j)}$ , denoted  $s_n^{(k,j)}$ , is computed from the velocity  $v_n^{(k,j)}$  of particles (equation (22)).  $\Pi_{u_n}$  represents the canonical projection in the direction  $u_n$ . Theoretically, the expression of the potential function for a measurement depends on the laws of the measurement error. The example given in (22) corresponds to a normal distribution with variance  $\sigma_l^2$ . Thus, according to the form of the measurement error, the potential function can be inferred. It is clear that with this expression, the more the velocity of a particle is close to the measurement, the higher the potential is. Let compute  $\mu_n^{(k,j)}$  (equation (24)). This term is the re-normalization of  $s_n^{(k,j)}$ .  $\mu_n^{(k,j)}$  may be seen as the probability of particle  $p_n^{(k,j)}$  given the observation  $(y_n^k, r_n^k)$ . The last step consists in a random re-sampling of particles depending on their potential. In other words, it is a re-sampling conditionally to the measurements. Before the filtering step, the law  $\eta_n^k$  of the particles velocity in  $\mathcal{C}_n^k$  is given in (23). Before the observation, each particle of  $\mathcal{C}_n^k$  has the same weight  $\frac{1}{c_n^k}$ . Equation (25) give the theoretical expression of  $\widehat{\eta}_n^k$ , the updated particle velocity law in  $\mathcal{C}_n^k$ .

$$s_n^{(k,j)} \stackrel{def}{=} e^{-\frac{(\Pi_{u_n}(v_n^{(k,j)}) - r_n^k)^2}{2\sigma_l^2}} \quad (22) \quad \mu_n^{(k,j)} \stackrel{def}{=} \frac{s_n^{(k,j)}}{\sum_{i=1}^{c_n^k} s_n^{(k,i)}} \quad (24)$$

$$\eta_n^k \stackrel{def}{=} \frac{1}{c_n^k} \sum_{j=1}^{c_n^k} \delta_{v_n^{(k,j)}} \quad (23) \quad \widehat{\eta}_n^k \stackrel{def}{=} \sum_{j=1}^{c_n^k} \mu_n^{(k,j)} \delta_{v_n^{(k,j)}} \quad (25)$$

After the filtering step, the weight of particle  $\hat{p}_n^{(k,j)}$  is  $\mu_n^{(k,j)}$ . Therefore an high potential implies, good correspondence with the measurements, and then large probability for the particle. It is necessary to compute a equi-weighted probability law because a particle must have a single velocity after this step. In practice, the operation consist in re-sampling velocities of particles of  $\mathcal{C}_n^k$ ,  $v_n^{(k,j)}$  transforms randomly into  $\hat{v}_n^{(k,j)} \stackrel{def}{=} v_n^{(k,\hat{j})}$  with probability  $\mu_n^{(k,\hat{j})}$ . Then, a particle which has a realistic velocity given the measurements survives with an high probability, whereas a particle with a low potential is killed and regenerates into a velocity with high potential.

The particles shaped after the filtering are noted  $\hat{p}_n^i$ . For a particle not concerned by a measurement,  $\hat{p}_n^i \stackrel{def}{=} p_n^i$ . Then,  $\widehat{R}_n^N \stackrel{def}{=} \frac{1}{N} \sum_{i=1}^N \hat{p}_n^i$ , the *a posteriori* distribution is computed. After this step, the population of particles is naturally closer to the truth, because coherent particles have survived and not adapted ones have been killed.

### 3.2. Discussion

The efficiency of non linear filtering is mathematically demonstrated in continuous spaces. Thus, to ensure that the algorithm presented here is still efficient, it is necessary to verify some conditions. It is very important that the number of particles concerned by each observation is sufficient. In other word, the phase space  $\mathcal{V}$  have to be sufficiently explored. In fact if no particle is close to the observation, it is clear that the filtering step explained before will not be efficient.

Particularly, in the model presented in 2, the exploration of the phase space is due to the random term coming from the Brownian motion. This problem of model exploration can appears problematic in no random models, but works are in progress to test a generalization of this algorithm applied to Eulerian deterministic models.

It is important to note that the algorithm presented here is very general and can be applied with others instruments or in more complicated spaces. For example, the particle space is  $\mathcal{Z} = \mathcal{X} \times \mathcal{V}$ , but in more complex models it might be  $\mathcal{Z}' = \mathcal{X} \times \mathcal{V}'$ , with a set of values  $\mathcal{V}'$  taking account the temperature or the humidity. In such case, it would be possible to assimilate data provided by others instruments. The only requirement to assimilate those data would be to formalize the observation operator. In this algorithm, measurements are supposed punctual with an influence neighborhood. A lot of instruments can be represented this way, but the method can even be adapted in more complicated situations. With the error law of the measurement and the appropriated projector in the observation space, the potential function can easily be computed. At last, the re-sampling step have to be properly implemented.

A great advantage of this formalization in term of particle network is that it allows to strongly reduce the complexity of the filtering problem. The step of localization, and the fact that the scoring and re-sampling processes do not depends on particles positions, allows in fact the filtering operation to be of dimension  $d_v = \dim(\mathcal{V})$ , the number of model variables. This technique reduces the dimension of filtering operation to low values, even for complicated models. Moreover, the computational complexity of the particle filtering process only increase linearly with the number of particle concerned. At last, this filtering process is naturally adapted for parallelization.

On the contrary, in Kalman filter based algorithms, it is necessary to make very expensive algebra operations as the optimization process implies manipulations (inversions, multiplications,...) on high dimension matrix.

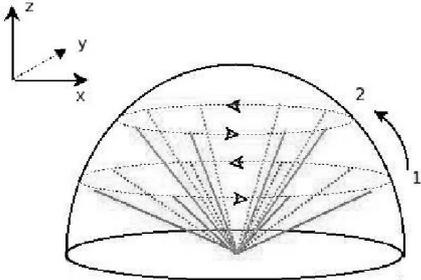
Particle filtering technique presents many advantages, and has an important potential in various field. The next section consists in demonstrating that the algorithm produce interesting result in a simple frame, and seem well adapted for the estimation of the turbulence with a remote detection instrument in the boundary layer.

## 4. Experiments

This section is devoted to detail an experiment performed with the model presented in 2, with the particle filtering of LIDAR measurements presented in 3. In, this simulation, LIDAR observations are simulated from an artificial reference wind field. The principle is to study how the numerical model react with the assimilation of LIDAR measurement in a controlled environment. The area considered is

$$\mathcal{B} = \{x = (x_x, x_y, x_z) \in \mathcal{X}, \|x\|_{\mathcal{X}} < \mathcal{R}, x_z \geq 0\}$$

The LIDAR is located at the center of this half bowl. The radius of the half bowl is fixed to  $\mathcal{R} = 1km$ . Figure 1, shows a schematic 3D view of the measurement process. Segments represents the direction targeted by the LIDAR. The measurement process consists in a succession of circular scans of the LIDAR over two elevation angles.



**Figure 1.** Description of the scanning process

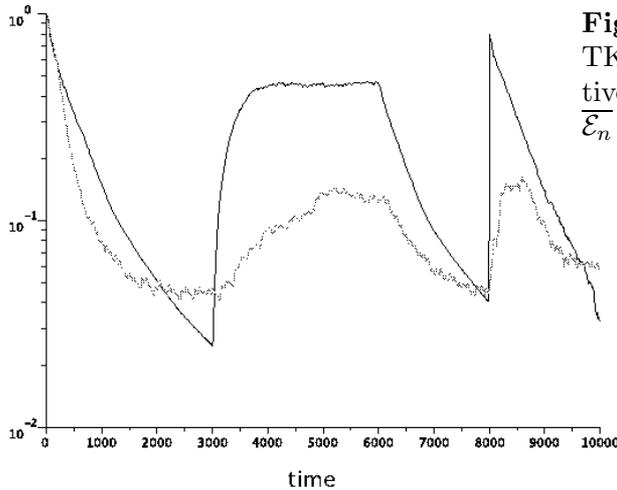
*In this experiment a LIDAR shoot consist in  $K = 10$  radial velocity measurements. The spatial resolution of the LIDAR is  $l_r = 100m$ . The condition (21) is chosen for the influence zone of an observation. For each elevation angle,  $\phi_1 = \frac{\pi}{8}$  and  $\phi_2 = \frac{3\pi}{8}$ , the LIDAR makes  $N_\theta = 8$  measurements turning anticlockwise. Measurements errors are supposed standard Gaussian with variance  $\sigma_l = 0,01m.s^{-1}$ . The frequency of LIDAR shoots is chosen equal to  $0,2Hz$  (at this frequency the time spent by the LIDAR to change direction is taken into account). The period of the LIDAR scan process is then equal to  $T = 80$  seconds. The time discretization  $\Delta t$  is chosen equal to 5 seconds. For  $n \in \mathbb{N}$ ,  $t_n = n\Delta t$ . In this experiment, the LIDAR measurement are simulated from a reference wind  $v_n^{ref}$ . The simulation is divided in 4 phases. For  $t_n \in [0, \tau_1[$ ,  $v_n^{ref} = (2, 0, 0)$ . Then, for  $t_n \in [\tau_1, \tau_2[$ , the wind turn regularly anticlockwise from  $(2, 0, 0)$  to  $(-2, 0, 0)$ . During the phase 3, for  $t_n \in [\tau_2, \tau_3[$  the wind stay constant with value  $v_n^{ref} = (-2, 0, 0)$ . A time  $\tau_3$ , the wind change instantaneously to  $v_n^{ref} = (-\sqrt{2}, -\sqrt{2}, 0)$  and keep this value to the end of the simulation. Here,  $\tau_1 = 3000s$ ,  $\tau_2 = 6000s$  and  $\tau_3 = 8000s$ .*

A limit condition is put on the ground. It consist in assimilating a virtual measurement  $v_z = 0$  with the filtering method presented in section 3. In this experiment, particles are concerned by this observation if their altitude is less than  $15m$ . This form of limit condition is satisfying for the purpose, and easier to implement than a real soil.

The model particles are initialized with an uniform random position on  $\mathcal{B}$  and with a three-dimensional normal law of zero mean and covariance matrix  $\sigma_{init}^2 = Id_3$  for the velocity.

Results are presented in figure 2. The evolution of two parameters is shown in logarithmic coordinate. The dotted line represents the mean of the TKE  $\overline{\mathcal{K}_n}$  whereas the plain line represents the relative error compared to the reference wind,  $\overline{\mathcal{E}_n}$ . At the beginning of the simulation, particles learn quite fast the wind tendency (shown in figure 2), because the decrease is exponential while the turbulence stabilizes. In the second part of the simulation, for  $t_n \in [\tau_1, \tau_2[$ , the reference wind direction change continuously from  $v_{n_1}^{ref} = (2, 0, 0)$  to  $v_{n_2}^{ref} = (-2, 0, 0)$  in

3000 seconds. The error and the turbulence increase during this phase, but remain stable. This mean that the filtering system keeps tracking the wind signal. At time  $t_{n_3} = n_3\Delta t$  the reference wind change instantaneously to  $v_{n_3}^{ref} = (-\sqrt{2}, -\sqrt{2}, 0)$  (as confirmed by figure). It is interesting to notice that the response in the TKE (dotted line) is immediate. Figure 3 show a three-dimensional view of the TKE field at a given instant during this transition phase. High turbulent zones are identified by dark color. The model parameters used in this simulation are summarized in a tabular.



**Figure 2.** Evolution of the mean of the TKE  $\bar{\mathcal{K}}_n$  (dotted line), and the mean relative error compared to the reference wind  $\bar{\mathcal{E}}_n$  (plain line), in logarithmic coordinate.

$$\bar{\mathcal{K}}_n \stackrel{def}{=} \frac{1}{N} \sum_{i=1}^N \mathcal{K}_n^i(x_n^i) \quad (26)$$

$$\bar{\mathcal{E}}_n \stackrel{def}{=} \frac{\|v_n^{ref} - \bar{v}_n\|_{\mathcal{V}}}{\|v_n^{ref}\|_{\mathcal{V}}} \quad (27)$$

$$\bar{v}_n \stackrel{def}{=} \frac{1}{N} \sum_{i=1}^N v_n^i \quad (28)$$

This simple experiment aims at demonstrating that the system is able to react properly and quickly to the observation even if the information provided by the LIDAR is partial. Moreover, it is important to notice that the filtering process is efficient even if the number of particle concerned by each LIDAR shoot is not very important ( almost 500 particle, about  $\sim 2,5\%$ ). The interactions term between particles, are very useful for both.

## 5. Conclusion

The results of the experiments, prove that the methodology and the algorithm of particle filtering is efficient in this simplified model. Conditioning the wind field by this non linear filtering process allows to estimate precisely the wind direction. Then, TKE field, can be estimated, because the numerical fluid react to the wind field and generate turbulence.

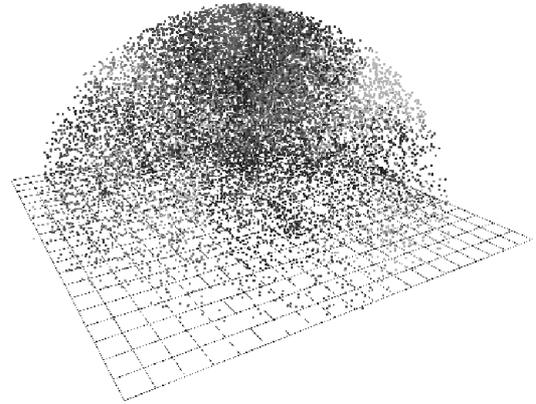
The next step of this work is to test the algorithm in with real LIDAR measurements on a physical turbulent fluid. More complete experiences are currently being prepared. One consists in testing the model in real boundary layer atmosphere, with a complete independent set of other observations (for example with a network of *in situ* instruments). This should allow to validate the methodology and to improve the model, for example by adjusting the different constants. Another necessary experience is to use the numerical model in a simulation laboratory. In such totally controlled environments, it would be possible to quantify precisely the estimations errors made by the model.

More generally, if the results are satisfying, it would be interesting to implements particle filtering algorithms in more complex models, with many types of observations and to compare particle filtering assimilation methods to more classical ones such as Kalman filter technique for example.

Model constants:

$N$	$\Delta t$	$C_0$	$\frac{C_1}{L_{scale}}$	$\lambda$	$\mathcal{R}$
20000	5s	2.1	0.001	100m	1000m

*It is possible to adjust those constants to modify the model behavior. For example, a high value of  $\lambda$  make the interactions more strong. In this case, the model learning time will be short. But, it would be impossible to observe little turbulences structure.*



**Figure 3.** 3D representation of the particles. The color represents the TKE, black for high values.

## References

- BAEHR, C. 2008 Nonlinear filtering for observations on a random vector field along a random path. *M2AN - ESAIM* **44** (5), 921–945.
- DEL MORAL, P. 2004 *Feynman-Kac Formulae, genealogical and interacting particle system with applications*, springer edn.
- FREHLICH, ROD, SHARMAN, ROBERT, MEILLIER, YANNICK, JENSEN, MICHAEL L. & BALSLEY, BEN 2006 Measurements of boundary layer profiles in an urban environment. *Journal of Applied Meteorology and Climatology* **45**, Issue 6, p.821–837.
- POPE, S.B 2000 *Turbulent Flows*. Cambridge University Press.