REAL-TIME PARALLEL PARTICLE FILTERING

Olivier Brun and Jean-Marie Garcia

LAAS-CNRS
7, Avenue du Colonel Roche
31077 Toulouse Cedex, France
e-mail : brun@laas.fr, jmg@laas.fr
Tel : 05-61-33-63-04

1.Introduction

Nonlinear filtering is the natural framework to state global estimation problems where a dynamic stochastic process is partially observed by a nonlinear measuring device (e.g. a RADAR) and corrupted by an additional stochastic process (the noise). The difficulty with nonlinear filtering formulation comes from the infinite dimensional character of the solution, which can not be derived in closed form. Approximate solutions based on local linearisation of system and measuring equations and/or moment truncation of density probability can not maintain garanteed performance or even stability.

The particle filtering technique may cope with nonlinear models as well as non-Gaussian dynamic and observation noises. It recursivelly constructs the conditional probability density $p(x_t|y_0 \dots y_t)$ of the state variables x_t, with respect to all available measurements $\{y_\tau, \tau \le t\}$, through a random exploration of the state space by entities called particles. A weight is assigned to each particle by a Bayesian correction term based on measurements. Its main advantage relies on global properties of the procedure, which leads to garanteed convergence.

However, the main drawback of such an approach to the non-linear filtering problem is its heavy computation cost. In practice, implementations of the particle filtering method can seldom cope with the real-time constraints of applications such as RADAR signal processing. Therefore, parallelism appears as a natural approach to real-time particle filtering.

The paper addresses the parallelization of the particle filtering technique. It is organized as follows. In section 2, we describe the particle filter principles. In section 3, we analyze the parallelism capabilities of the algorithm. Section 4 is devoted to results and concluding remarks.

2.Particle Filter

Let the following equation represent a discrete dynamic system x with its observation y :

$$x_{t+1} = f(t, x_t, w_t), \quad x_t \in R^n, w_t \in R^m, t \in N$$

$$y_t = h(x_t) + v_t$$

where w and v_t are independent white noises with known probability distributions. It is well known that the recursive solution of the conditionnal probability density $p(x_t|y_0 \dots y_t)$ involves two steps : correction and prediction. This recursive solution can be stated as follows :

$$p(x_t|y_0 \dots y_t) = \frac{p(y_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|y_0 \dots y_{t-1}) dx_{t-1}}{\int p(y_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|y_0 \dots y_{t-1}) dx_{t-1} dx_t}$$

Unfortunately, such a recursive computation lies in an infinite dimensional span and is thus unrealizable, with the noticeable exception of the well known unconstrained linear Gaussian problem.

Particle filter is a procedure to solve the general nonlinear non-Gaussian problem, as previously developped in [1]. Particle filtering approximates the a priori probability $p(x_0)$ at initial time by a set of N Dirac distributions and applies the dynamic of the system to this set. In other words, one "randomly selects N particles" from the probability distribution $p(x_0)$ to represent it as:

$$p(x_0) \cong \sum_{i=1}^{N} \frac{1}{N} \delta_{x_0^i}(x_0)$$

where x_0^i is the position of particle i. Next, each particle i follows the system dynamic $x_t^i = f(x_{t-1}^i, w_t^i)$ with noise samples w_t^i generated from its *a priori* probability. Time evolution of Dirac point measures is given by sequential application of this procedure :

$$p(x_t) \cong \sum_{i=1}^{N} \frac{1}{N} \delta_{x_t^i}(x_t)$$

Fig. 1.a shows the evolution of a set of particles (dot lines represent the trajectory of each particle).

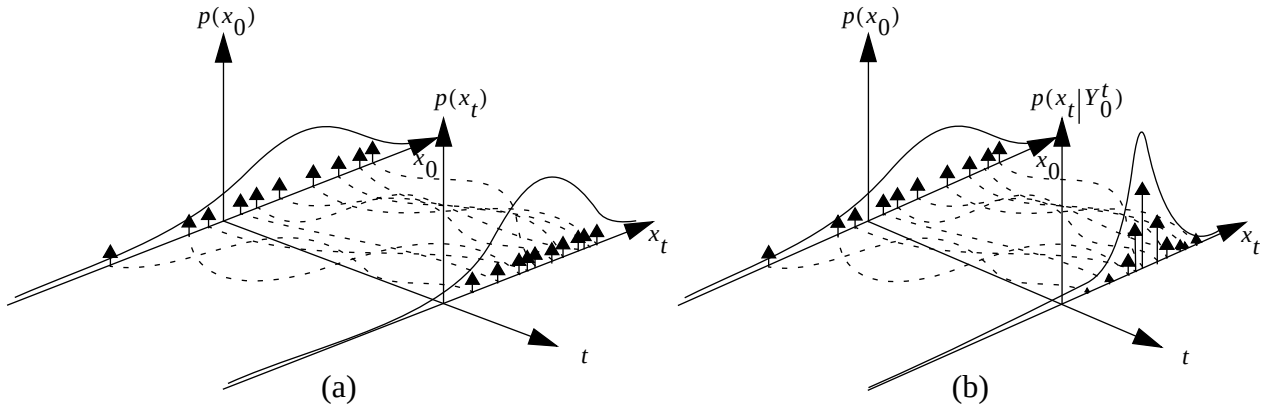

Fig. 1 : (a) a priori evolution and (b) conditionnal evolution.

Next step is to introduce information carried by the measure $\{y_\tau, \tau \le t\}$:

$$p(x_t|y_0^t) \cong \sum_{i=1}^{N} \rho_t^i \delta_{x_t^i}(x_t) \qquad \text{with} \qquad \rho_t^i = (p(y_t|x_t^i)\ldots p(y_0|x_0^i)) \Big/ \left(\sum_{i=1}^{N} p(y_t|x_t^i)\ldots p(y_0|x_0^i) \right)$$

The weight ρ_t^i corrects the representation given in figure 1.a, where all particles had the same weight $1/N$. The particle estimation of the conditional probability is depicted in figure 1.b (dot lines are the trajectories and the arrows' amplitude represents the weight of each particle).

When the observation noise is gaussian, we have the following simple expression of the weight ρ_t^i :

$$\rho_t^i = e^{V_t^i} \Big/ \left(\sum e^{V_t^k} \right) \qquad \text{with} \qquad V_t^i = V_{t-1}^i - \frac{1}{2}((y_t - h(x_t^i))^2 / R)$$

V_t^i is the likelihood of the trajectory $\{x_0^i, \ldots, x_t^i\}$. The estimate is then given by :

$$\hat{x}_t = \sum_{i=1}^{N} \rho_t^i x_t^i$$

The main drawback of the particle filter is that, if the state space is unbounded, particle trajectories diverge. Furthermore, in the absence of regularisation, particle weights degenerate and the law of large numbers is no more satisfied. A resampling technique can be used to solve both difficulties at the same time [2]. The algorithm is restarted at an instant t using the estimated conditional probability as an initial distribution. All particles are redistributed among the states x_t^i according to the weight

attributed to them and take the new same weight $1/N$ after the redistribution. Therefore, most probable states, corresponding to “heavy” particles, give rise to several new particles while least probable ones are “killed”. Resampling locates particles where they are needed, in a probabilistic way. Note that particle resampling is not done all time. Particles need to ``learn'' from measurements about the likelihood of their paths. Redistributions can be done in a periodic way or in an adaptive way defined by the particles' weigth distribution.

3.Parallel Particle Algorithm

We propose a data parallel approach that consists in partitionning the set of particles in several separate sub-sets of the same size [5]. Each subset is affected to one processor. This approach allows to exploit efficiently the inherent parallelism of the evolution and ponderation steps of the particle algorithm. Since the subsets are separate, one can expect a linear speedup if the communication overhead is keep negligeable.

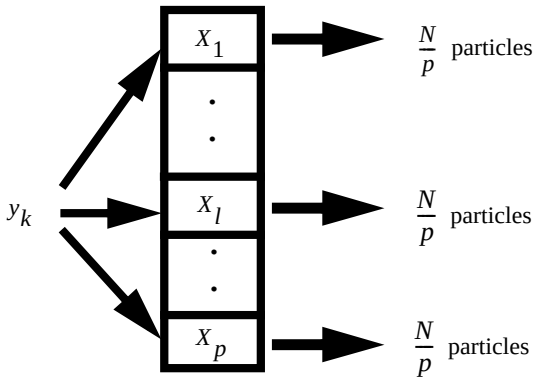

Fig. 2: Principle of the data parallel approach.

However, we are faced with two difficulties. The first one is related to the parallel computation of the estimate $\hat{x}_t$ from the distributed set of particles. The second one concerns the control of the degenerascy of the distributed set of particles.

3.1. Parallel Computation of the Estimate

To solve the first problem, we observe that the estimate needs not be calculated on each processor. Hence, only one task, hereafter called master task, is devoted to this computation. The other tasks, which are referred to as slave tasks in the sequel, compute and send to the master task the informations necessary to compute the estimate. To determine which informations are needed at the master task level, observe that the estimate $\hat{x}_t$ can be re-written as below:

$$\hat{x}_t = \sum_{i=1}^{p} E_i^t / \sum_{k=1}^{p} P_k^t \qquad \text{with} \qquad E_l^t = \sum_{i=1}^{N/P} e^{V_t^i(l)}\, x_t^i(l) \qquad \text{and} \qquad P_l^t = \sum_{i=1}^{N/P} e^{V_t^i(l)}$$

where P is the number of slave tasks. The quantity E_l^t corresponds to a non-normalized local estimate while the quantity P_l^t corresponds to a non-normalized local weight.

This formulation allows to compute the parallel estimate as depicted on Fig. 3.

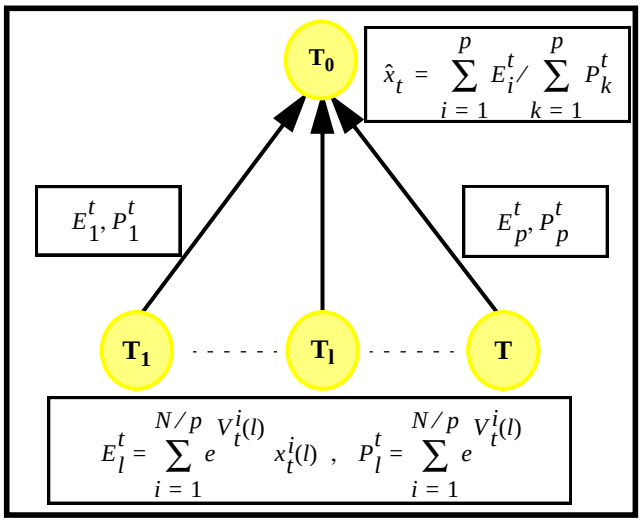

Fig. 3: Communications involved in the parallel estimate computation.

3.2. Degeneracy Control of the Distributed Set of Particles

Several approaches can be used to cope with the degeneracy of the distributed set of particles.

The first approach is to realize a total exchange of the agregated informations E_l^t and P_l^t at each step of the algorithm. It allows each processor to detect the degeneracy of the subset of particles of any other processor. A total exchange of the particle subsets can then be done to reconstruct the global set of particles on each processor. This global set of particles is then redistributed and repartitionned independantly on each processor. This approach has the interesting feature of decreasing the estimation error as the number of processor increases. However, the cost of particle subset total exchanges is too important to cope with the real-time constraints of applications such as RADAR.

The second approach we propose consists in operating local redistributions on the subset of particles. Such redistributions can be done independently on each processor. It can be shown using birth and death processes theory that local redistribution allows to stabilize the subset of particles. This property holds even if the number of particles in each subset is as lower as 2, as shown on figure 4.

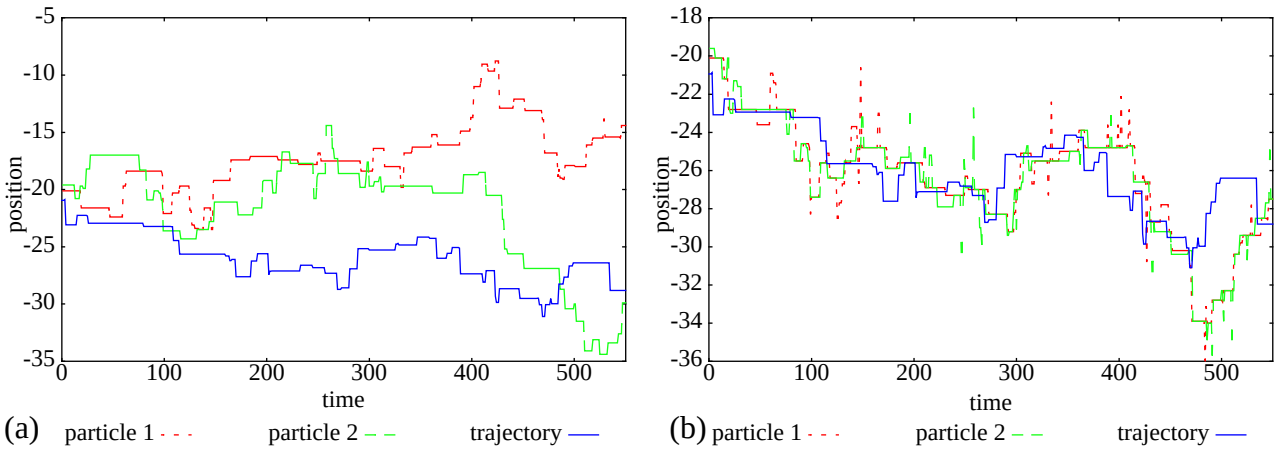

Fig. 4 : (a) without local redistribution and (b) with local redistribution.

4.Results and concluding remarks

The academic non-linear filtering problems considered for the application are the followings. The first model is relative to a jump process $x_{t+1} = x_t + \pi_t$ with a poisson additive perturbation π_t of random gaussian magnitude and an observation process $y_t = x_t + v_t$ corrupted by a gaussian noise. The second model describes the evolution of a dynamic mobile (position, velocity and acceleration) with the same poisson perturbation on the acceleration and the same gaussian noise on the observation y.

$$a_t = \alpha \cdot a_{t-1} + \pi_t ; \qquad v_t = (1 - k_1) \cdot v_{t-1} + a_t ; \qquad p_t = p_{t-1} + v_t$$

$$y_t = p_t + \gamma_t$$

On both models, we generated four instances of the particle algorithm. Instances differ with respect to the number of particles (4000 and 16000) and the magnitude of noise variables. The parallel algorithm has been implemented on an Origin2000 (32 processors) with MPI message passing library.

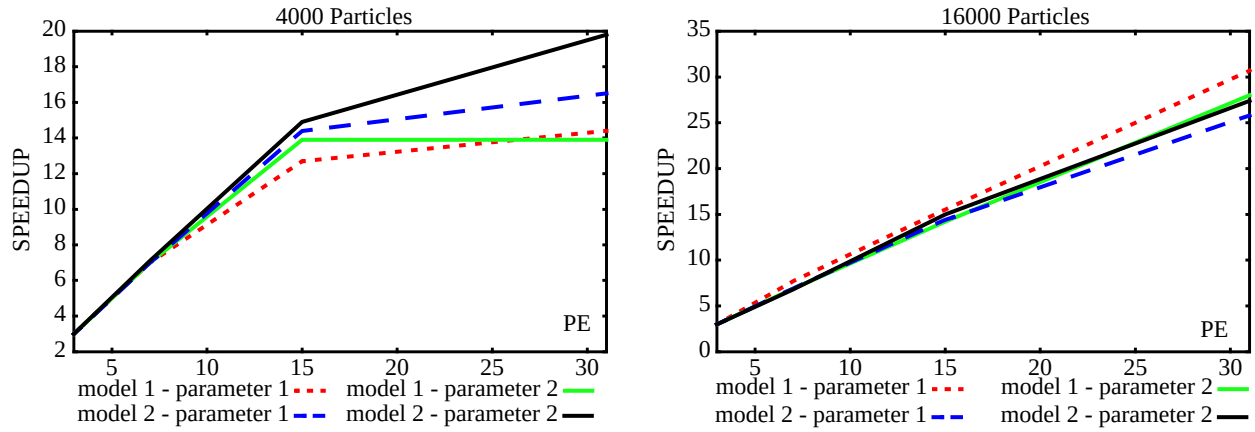

Fig. 5: Speedups obtained on the Origin2000 using 4000 and 16000 particles.

The results plotted on figure 5 show the very good scalability (near linear speedup) of the parallel algorithm for 16000 particles. The resulting parallel computation time per iteration is compatible with real-time applications (smaller than 1 ms).

We are developping new approaches to extend the scalability of the method to hundreds of processors to handle the larger number of particles (greater than 50000) necessary in real application problems such as the LORAN navigation signal processing [4] and RADAR defense applications with very low signal-noise ratio [3].

In parallel, we also investigate the capabilities of PC clusters with high-speed communication links (myrinet) to handle efficiently such parallel method.

5.References

[1] P. Del Moral and G. Salut. Filtrage non-linéaire: résolution particulaire à la Monte-Carlo. C.R. Acad. Sci. Paris, t.320, Série I:p1147-1152,1995.

[2] G. Rigal. Filtrage non-linéaire, résolution particulaire et application au traitement du signal. Thèse de doctorat, Université Paul Sabatier, Toulouse, juillet 1993.

[3] J-C Noyer and G. Salut. Optimal non-linear radar tracking of non-Gaussian manoeuvering targets. research report 96113, LAAS-CNRS, 1996.

[4] A. Monin. Evaluation de la limite de portée du LORAN-C par filtrage particulaire, JDA'99, Journées Doctorales d'Automatique, September 1999.

[5] O. Brun and J.M. Garcia. Parallélisation de la méthode de filtrage particulaire, research report LAAS-CNRS, September 1999.

REAL-TIME PARALLEL PARTICLE FILTERING

Olivier Brun and Jean-Marie Garcia

LAAS-CNRS
7, Avenue du Colonel Roche
31077 Toulouse Cedex, France
e-mail : brun@laas.fr, jmg@laas.fr
Tel : 05-61-33-63-04

1.Introduction

Nonlinear filtering is the natural framework to state global estimation problems where a dynamic stochastic process is partially observed by a nonlinear measuring device (e.g. a RADAR) and corrupted by an additional stochastic process (the noise). The difficulty with nonlinear filtering formulation comes from the infinite dimensional character of the solution, which can not be derived in closed form. Approximate solutions based on local linearisation of system and measuring equations and/or moment truncation of density probability can not maintain garanteed performance or even stability.

The particle filtering technique may cope with nonlinear models as well as non-Gaussian dynamic and observation noises. It recursivelly constructs the conditional probability density $p(x_t|y_0 \dots y_t)$ of the state variables x_t, with respect to all available measurements $\{y_\tau, \tau \le t\}$, through a random exploration of the state space by entities called particles. A weight is assigned to each particle by a Bayesian correction term based on measurements. Its main advantage relies on global properties of the procedure, which leads to garanteed convergence.

However, the main drawback of such an approach to the non-linear filtering problem is its heavy computation cost. In practice, implementations of the particle filtering method can seldom cope with the real-time constraints of applications such as RADAR signal processing. Therefore, parallelism appears as a natural approach to real-time particle filtering.

The paper addresses the parallelization of the particle filtering technique. It is organized as follows. In section 2, we describe the particle filter principles. In section 3, we analyze the parallelism capabilities of the algorithm. Section 4 is devoted to results and concluding remarks.

2.Particle Filter

Let the following equation represent a discrete dynamic system x with its observation y :

$$x_{t+1} = f(t, x_t, w_t), \quad x_t \in R^n, w_t \in R^m, t \in N$$

$$y_t = h(x_t) + v_t$$

where w and v_t are independent white noises with known probability distributions. It is well known that the recursive solution of the conditionnal probability density $p(x_t|y_0 \dots y_t)$ involves two steps : correction and prediction. This recursive solution can be stated as follows :

$$p(x_t|y_0 \dots y_t) = \frac{p(y_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|y_0 \dots y_{t-1}) dx_{t-1}}{\int p(y_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|y_0 \dots y_{t-1}) dx_{t-1} dx_t}$$

Unfortunately, such a recursive computation lies in an infinite dimensional span and is thus unrealizable, with the noticeable exception of the well known unconstrained linear Gaussian problem.

Particle filter is a procedure to solve the general nonlinear non-Gaussian problem, as previously developped in [1]. Particle filtering approximates the a priori probability $p(x_0)$ at initial time by a set of N Dirac distributions and applies the dynamic of the system to this set. In other words, one "randomly selects N particles" from the probability distribution $p(x_0)$ to represent it as:

$$p(x_0) \cong \sum_{i=1}^{N} \frac{1}{N} \delta_{x_0^i}(x_0)$$

where x_0^i is the position of particle i. Next, each particle i follows the system dynamic $x_t^i = f(x_{t-1}^i, w_t^i)$ with noise samples w_t^i generated from its *a priori* probability. Time evolution of Dirac point measures is given by sequential application of this procedure :

$$p(x_t) \cong \sum_{i=1}^{N} \frac{1}{N} \delta_{x_t^i}(x_t)$$

Fig. 1.a shows the evolution of a set of particles (dot lines represent the trajectory of each particle).

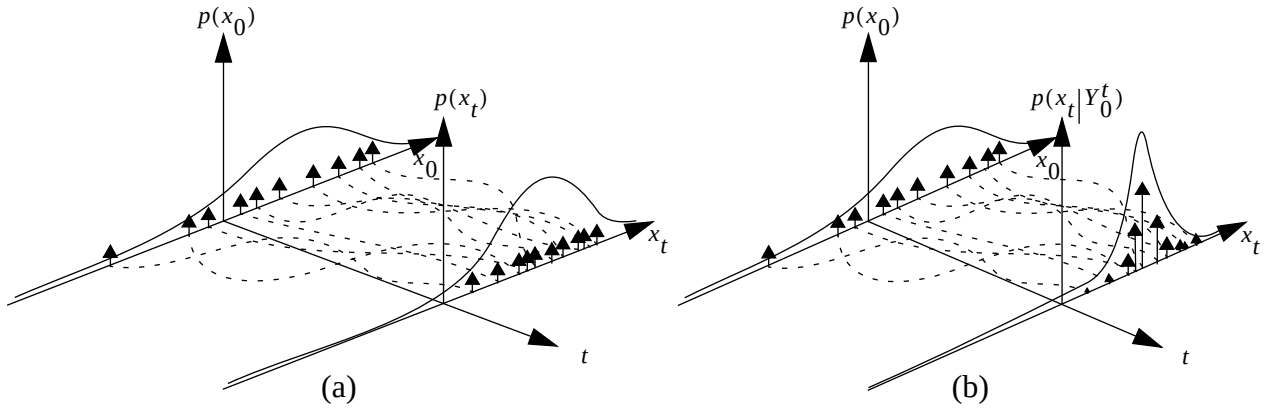

Fig. 1 : (a) a priori evolution and (b) conditionnal evolution.

Next step is to introduce information carried by the measure $\{y_\tau, \tau \le t\}$:

$$p(x_t|y_0^t) \cong \sum_{i=1}^{N} \rho_t^i \delta_{x_t^i}(x_t) \qquad \text{with} \qquad \rho_t^i = (p(y_t|x_t^i)\ldots p(y_0|x_0^i)) \Big/ \left(\sum_{i=1}^{N} p(y_t|x_t^i)\ldots p(y_0|x_0^i) \right)$$

The weight ρ_t^i corrects the representation given in figure 1.a, where all particles had the same weight $1/N$. The particle estimation of the conditional probability is depicted in figure 1.b (dot lines are the trajectories and the arrows' amplitude represents the weight of each particle).

When the observation noise is gaussian, we have the following simple expression of the weight ρ_t^i :

$$\rho_t^i = e^{V_t^i} \Big/ \left(\sum e^{V_t^k} \right) \qquad \text{with} \qquad V_t^i = V_{t-1}^i - \frac{1}{2}((y_t - h(x_t^i))^2 / R)$$

V_t^i is the likelihood of the trajectory $\{x_0^i, \ldots, x_t^i\}$. The estimate is then given by :

$$\hat{x}_t = \sum_{i=1}^{N} \rho_t^i x_t^i$$

The main drawback of the particle filter is that, if the state space is unbounded, particle trajectories diverge. Furthermore, in the absence of regularisation, particle weights degenerate and the law of large numbers is no more satisfied. A resampling technique can be used to solve both difficulties at the same time [2]. The algorithm is restarted at an instant t using the estimated conditional probability as an initial distribution. All particles are redistributed among the states x_t^i according to the weight

attributed to them and take the new same weight $1/N$ after the redistribution. Therefore, most probable states, corresponding to “heavy” particles, give rise to several new particles while least probable ones are “killed”. Resampling locates particles where they are needed, in a probabilistic way. Note that particle resampling is not done all time. Particles need to ``learn'' from measurements about the likelihood of their paths. Redistributions can be done in a periodic way or in an adaptive way defined by the particles' weigth distribution.

3.Parallel Particle Algorithm

We propose a data parallel approach that consists in partitionning the set of particles in several separate sub-sets of the same size [5]. Each subset is affected to one processor. This approach allows to exploit efficiently the inherent parallelism of the evolution and ponderation steps of the particle algorithm. Since the subsets are separate, one can expect a linear speedup if the communication overhead is keep negligeable.

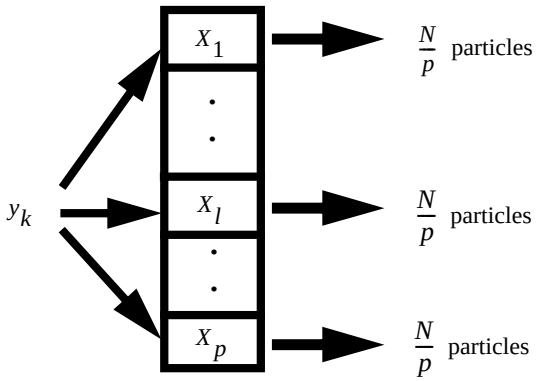

Fig. 2: Principle of the data parallel approach.

However, we are faced with two difficulties. The first one is related to the parallel computation of the estimate $\hat{x}_t$ from the distributed set of particles. The second one concerns the control of the degenerascy of the distributed set of particles.

3.1. Parallel Computation of the Estimate

To solve the first problem, we observe that the estimate needs not be calculated on each processor. Hence, only one task, hereafter called master task, is devoted to this computation. The other tasks, which are referred to as slave tasks in the sequel, compute and send to the master task the informations necessary to compute the estimate. To determine which informations are needed at the master task level, observe that the estimate $\hat{x}_t$ can be re-written as below:

$$\hat{x}_t = \sum_{i=1}^{p} E_i^t / \sum_{k=1}^{p} P_k^t \qquad \text{with} \qquad E_l^t = \sum_{i=1}^{N/P} e^{V_t^i(l)}\, x_t^i(l) \qquad \text{and} \qquad P_l^t = \sum_{i=1}^{N/P} e^{V_t^i(l)}$$

where P is the number of slave tasks. The quantity E_l^t corresponds to a non-normalized local estimate while the quantity P_l^t corresponds to a non-normalized local weight.

This formulation allows to compute the parallel estimate as depicted on Fig. 3.

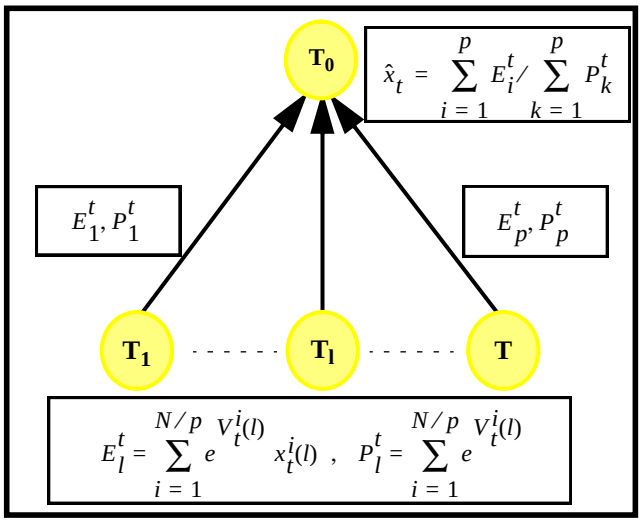

Fig. 3: Communications involved in the parallel estimate computation.

3.2. Degeneracy Control of the Distributed Set of Particles

Several approaches can be used to cope with the degeneracy of the distributed set of particles.

The first approach is to realize a total exchange of the agregated informations E_l^t and P_l^t at each step of the algorithm. It allows each processor to detect the degeneracy of the subset of particles of any other processor. A total exchange of the particle subsets can then be done to reconstruct the global set of particles on each processor. This global set of particles is then redistributed and repartitionned independantly on each processor. This approach has the interesting feature of decreasing the estimation error as the number of processor increases. However, the cost of particle subset total exchanges is too important to cope with the real-time constraints of applications such as RADAR.

The second approach we propose consists in operating local redistributions on the subset of particles. Such redistributions can be done independently on each processor. It can be shown using birth and death processes theory that local redistribution allows to stabilize the subset of particles. This property holds even if the number of particles in each subset is as lower as 2, as shown on figure 4.

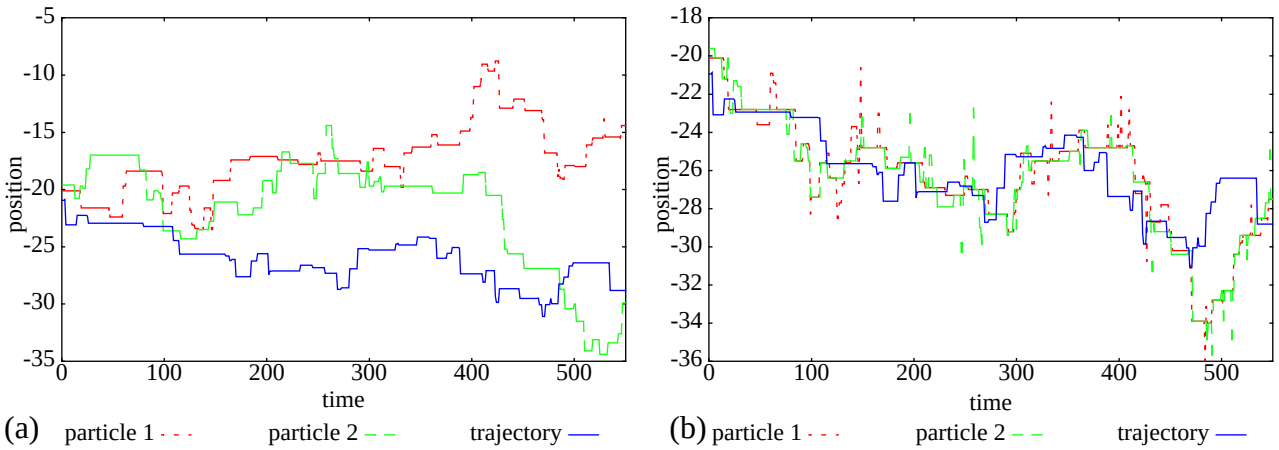

Fig. 4 : (a) without local redistribution and (b) with local redistribution.

4.Results and concluding remarks

The academic non-linear filtering problems considered for the application are the followings. The first model is relative to a jump process $x_{t+1} = x_t + \pi_t$ with a poisson additive perturbation π_t of random gaussian magnitude and an observation process $y_t = x_t + v_t$ corrupted by a gaussian noise. The second model describes the evolution of a dynamic mobile (position, velocity and acceleration) with the same poisson perturbation on the acceleration and the same gaussian noise on the observation y.

$$a_t = \alpha \cdot a_{t-1} + \pi_t ; \qquad v_t = (1 - k_1) \cdot v_{t-1} + a_t ; \qquad p_t = p_{t-1} + v_t$$

$$y_t = p_t + \gamma_t$$

On both models, we generated four instances of the particle algorithm. Instances differ with respect to the number of particles (4000 and 16000) and the magnitude of noise variables. The parallel algorithm has been implemented on an Origin2000 (32 processors) with MPI message passing library.

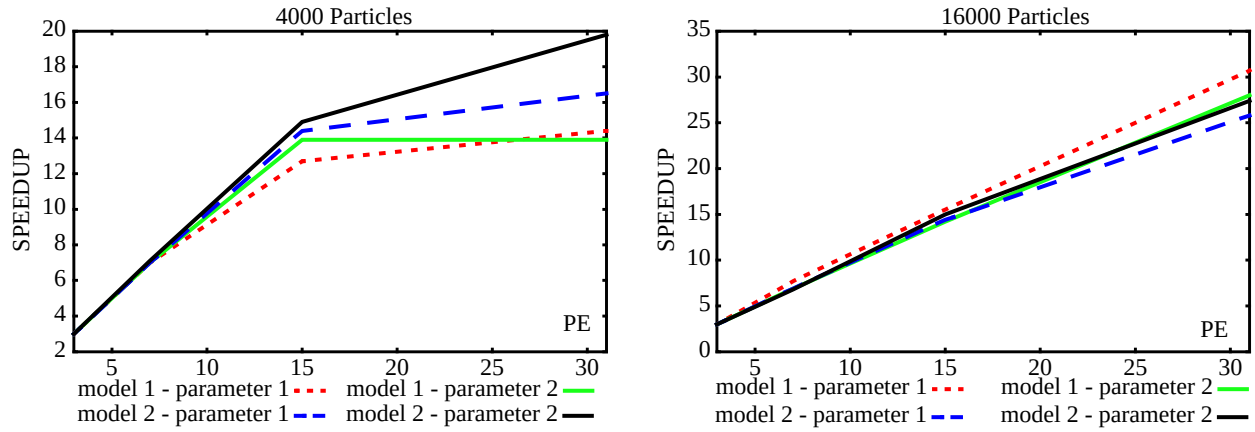

Fig. 5: Speedups obtained on the Origin2000 using 4000 and 16000 particles.

The results plotted on figure 5 show the very good scalability (near linear speedup) of the parallel algorithm for 16000 particles. The resulting parallel computation time per iteration is compatible with real-time applications (smaller than 1 ms).

We are developping new approaches to extend the scalability of the method to hundreds of processors to handle the larger number of particles (greater than 50000) necessary in real application problems such as the LORAN navigation signal processing [4] and RADAR defense applications with very low signal-noise ratio [3].

In parallel, we also investigate the capabilities of PC clusters with high-speed communication links (myrinet) to handle efficiently such parallel method.

5.References

[1] P. Del Moral and G. Salut. Filtrage non-linéaire: résolution particulaire à la Monte-Carlo. C.R. Acad. Sci. Paris, t.320, Série I:p1147-1152,1995.

[2] G. Rigal. Filtrage non-linéaire, résolution particulaire et application au traitement du signal. Thèse de doctorat, Université Paul Sabatier, Toulouse, juillet 1993.

[3] J-C Noyer and G. Salut. Optimal non-linear radar tracking of non-Gaussian manoeuvering targets. research report 96113, LAAS-CNRS, 1996.

[4] A. Monin. Evaluation de la limite de portée du LORAN-C par filtrage particulaire, JDA'99, Journées Doctorales d'Automatique, September 1999.

[5] O. Brun and J.M. Garcia. Parallélisation de la méthode de filtrage particulaire, research report LAAS-CNRS, September 1999.

REAL-TIME PARALLEL PARTICLE FILTERING

Olivier Brun and Jean-Marie Garcia

LAAS-CNRS
7, Avenue du Colonel Roche
31077 Toulouse Cedex, France
e-mail : brun@laas.fr, jmg@laas.fr
Tel : 05-61-33-63-04

1.Introduction

Nonlinear filtering is the natural framework to state global estimation problems where a dynamic stochastic process is partially observed by a nonlinear measuring device (e.g. a RADAR) and corrupted by an additional stochastic process (the noise). The difficulty with nonlinear filtering formulation comes from the infinite dimensional character of the solution, which can not be derived in closed form. Approximate solutions based on local linearisation of system and measuring equations and/or moment truncation of density probability can not maintain garanteed performance or even stability.

The particle filtering technique may cope with nonlinear models as well as non-Gaussian dynamic and observation noises. It recursivelly constructs the conditional probability density $p(x_t|y_0 \dots y_t)$ of the state variables x_t, with respect to all available measurements $\{y_\tau, \tau \le t\}$, through a random exploration of the state space by entities called particles. A weight is assigned to each particle by a Bayesian correction term based on measurements. Its main advantage relies on global properties of the procedure, which leads to garanteed convergence.

However, the main drawback of such an approach to the non-linear filtering problem is its heavy computation cost. In practice, implementations of the particle filtering method can seldom cope with the real-time constraints of applications such as RADAR signal processing. Therefore, parallelism appears as a natural approach to real-time particle filtering.

The paper addresses the parallelization of the particle filtering technique. It is organized as follows. In section 2, we describe the particle filter principles. In section 3, we analyze the parallelism capabilities of the algorithm. Section 4 is devoted to results and concluding remarks.

2.Particle Filter

Let the following equation represent a discrete dynamic system x with its observation y :

$$x_{t+1} = f(t, x_t, w_t), \quad x_t \in R^n, w_t \in R^m, t \in N$$

$$y_t = h(x_t) + v_t$$

where w and v_t are independent white noises with known probability distributions. It is well known that the recursive solution of the conditionnal probability density $p(x_t|y_0 \dots y_t)$ involves two steps : correction and prediction. This recursive solution can be stated as follows :

$$p(x_t|y_0 \dots y_t) = \frac{p(y_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|y_0 \dots y_{t-1}) dx_{t-1}}{\int p(y_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|y_0 \dots y_{t-1}) dx_{t-1} dx_t}$$

Unfortunately, such a recursive computation lies in an infinite dimensional span and is thus unrealizable, with the noticeable exception of the well known unconstrained linear Gaussian problem.

Particle filter is a procedure to solve the general nonlinear non-Gaussian problem, as previously developped in [1]. Particle filtering approximates the a priori probability $p(x_0)$ at initial time by a set of N Dirac distributions and applies the dynamic of the system to this set. In other words, one "randomly selects N particles" from the probability distribution $p(x_0)$ to represent it as:

$$p(x_0) \cong \sum_{i=1}^{N} \frac{1}{N} \delta_{x_0^i}(x_0)$$

where x_0^i is the position of particle i. Next, each particle i follows the system dynamic $x_t^i = f(x_{t-1}^i, w_t^i)$ with noise samples w_t^i generated from its *a priori* probability. Time evolution of Dirac point measures is given by sequential application of this procedure :

$$p(x_t) \cong \sum_{i=1}^{N} \frac{1}{N} \delta_{x_t^i}(x_t)$$

Fig. 1.a shows the evolution of a set of particles (dot lines represent the trajectory of each particle).

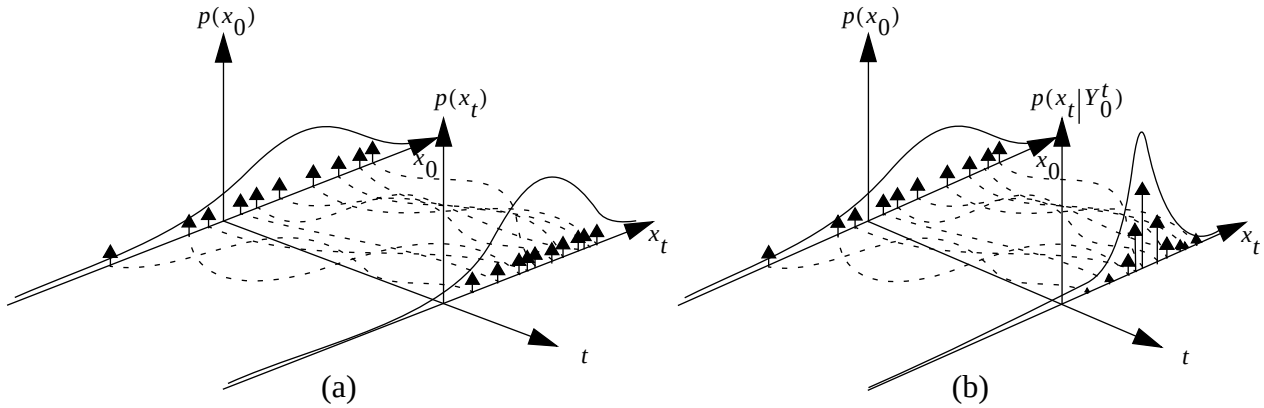

Fig. 1 : (a) a priori evolution and (b) conditionnal evolution.

Next step is to introduce information carried by the measure $\{y_\tau, \tau \le t\}$:

$$p(x_t|y_0^t) \cong \sum_{i=1}^{N} \rho_t^i \delta_{x_t^i}(x_t) \qquad \text{with} \qquad \rho_t^i = (p(y_t|x_t^i)\ldots p(y_0|x_0^i)) \Big/ \left(\sum_{i=1}^{N} p(y_t|x_t^i)\ldots p(y_0|x_0^i) \right)$$

The weight ρ_t^i corrects the representation given in figure 1.a, where all particles had the same weight $1/N$. The particle estimation of the conditional probability is depicted in figure 1.b (dot lines are the trajectories and the arrows' amplitude represents the weight of each particle).

When the observation noise is gaussian, we have the following simple expression of the weight ρ_t^i :

$$\rho_t^i = e^{V_t^i} \Big/ \left(\sum e^{V_t^k} \right) \qquad \text{with} \qquad V_t^i = V_{t-1}^i - \frac{1}{2}((y_t - h(x_t^i))^2 / R)$$

V_t^i is the likelihood of the trajectory $\{x_0^i, \ldots, x_t^i\}$. The estimate is then given by :

$$\hat{x}_t = \sum_{i=1}^{N} \rho_t^i x_t^i$$

The main drawback of the particle filter is that, if the state space is unbounded, particle trajectories diverge. Furthermore, in the absence of regularisation, particle weights degenerate and the law of large numbers is no more satisfied. A resampling technique can be used to solve both difficulties at the same time [2]. The algorithm is restarted at an instant t using the estimated conditional probability as an initial distribution. All particles are redistributed among the states x_t^i according to the weight

attributed to them and take the new same weight $1/N$ after the redistribution. Therefore, most probable states, corresponding to “heavy” particles, give rise to several new particles while least probable ones are “killed”. Resampling locates particles where they are needed, in a probabilistic way. Note that particle resampling is not done all time. Particles need to ``learn'' from measurements about the likelihood of their paths. Redistributions can be done in a periodic way or in an adaptive way defined by the particles' weigth distribution.

3.Parallel Particle Algorithm

We propose a data parallel approach that consists in partitionning the set of particles in several separate sub-sets of the same size [5]. Each subset is affected to one processor. This approach allows to exploit efficiently the inherent parallelism of the evolution and ponderation steps of the particle algorithm. Since the subsets are separate, one can expect a linear speedup if the communication overhead is keep negligeable.

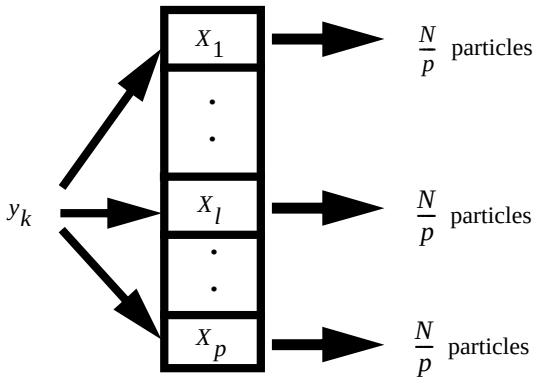

Fig. 2: Principle of the data parallel approach.

However, we are faced with two difficulties. The first one is related to the parallel computation of the estimate $\hat{x}_t$ from the distributed set of particles. The second one concerns the control of the degenerascy of the distributed set of particles.

3.1. Parallel Computation of the Estimate

To solve the first problem, we observe that the estimate needs not be calculated on each processor. Hence, only one task, hereafter called master task, is devoted to this computation. The other tasks, which are referred to as slave tasks in the sequel, compute and send to the master task the informations necessary to compute the estimate. To determine which informations are needed at the master task level, observe that the estimate $\hat{x}_t$ can be re-written as below:

$$\hat{x}_t = \sum_{i=1}^{p} E_i^t / \sum_{k=1}^{p} P_k^t \qquad \text{with} \qquad E_l^t = \sum_{i=1}^{N/P} e^{V_t^i(l)}\, x_t^i(l) \qquad \text{and} \qquad P_l^t = \sum_{i=1}^{N/P} e^{V_t^i(l)}$$

where P is the number of slave tasks. The quantity E_l^t corresponds to a non-normalized local estimate while the quantity P_l^t corresponds to a non-normalized local weight.

This formulation allows to compute the parallel estimate as depicted on Fig. 3.

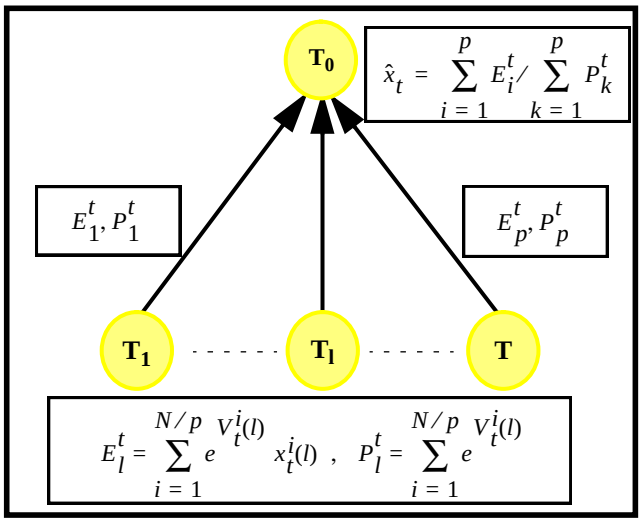

Fig. 3: Communications involved in the parallel estimate computation.

3.2. Degeneracy Control of the Distributed Set of Particles

Several approaches can be used to cope with the degeneracy of the distributed set of particles.

The first approach is to realize a total exchange of the agregated informations E_l^t and P_l^t at each step of the algorithm. It allows each processor to detect the degeneracy of the subset of particles of any other processor. A total exchange of the particle subsets can then be done to reconstruct the global set of particles on each processor. This global set of particles is then redistributed and repartitionned independantly on each processor. This approach has the interesting feature of decreasing the estimation error as the number of processor increases. However, the cost of particle subset total exchanges is too important to cope with the real-time constraints of applications such as RADAR.

The second approach we propose consists in operating local redistributions on the subset of particles. Such redistributions can be done independently on each processor. It can be shown using birth and death processes theory that local redistribution allows to stabilize the subset of particles. This property holds even if the number of particles in each subset is as lower as 2, as shown on figure 4.

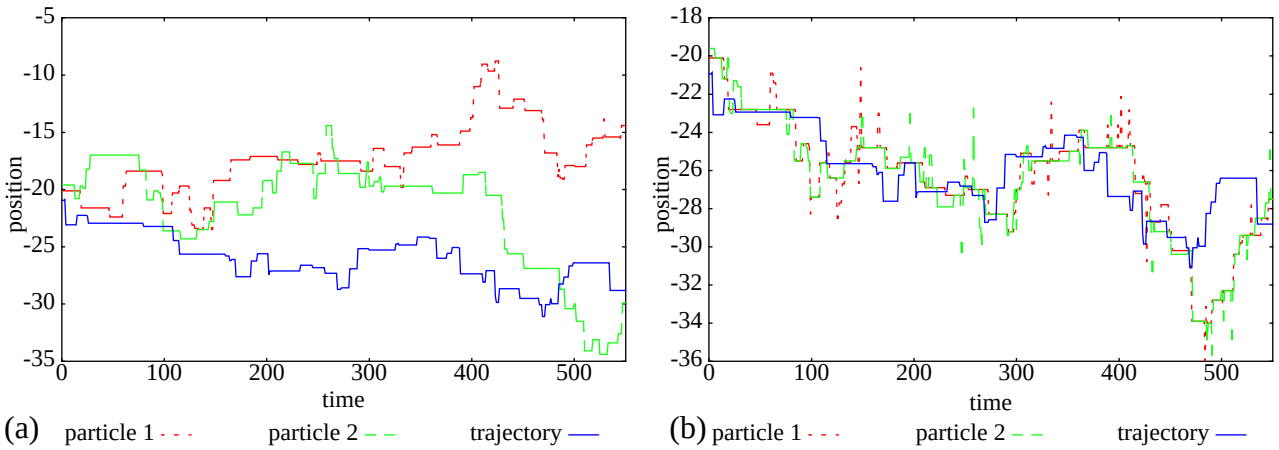

Fig. 4 : (a) without local redistribution and (b) with local redistribution.

4.Results and concluding remarks

The academic non-linear filtering problems considered for the application are the followings. The first model is relative to a jump process $x_{t+1} = x_t + \pi_t$ with a poisson additive perturbation π_t of random gaussian magnitude and an observation process $y_t = x_t + v_t$ corrupted by a gaussian noise. The second model describes the evolution of a dynamic mobile (position, velocity and acceleration) with the same poisson perturbation on the acceleration and the same gaussian noise on the observation y.

$$a_t = \alpha \cdot a_{t-1} + \pi_t ; \qquad v_t = (1 - k_1) \cdot v_{t-1} + a_t ; \qquad p_t = p_{t-1} + v_t$$

$$y_t = p_t + \gamma_t$$

On both models, we generated four instances of the particle algorithm. Instances differ with respect to the number of particles (4000 and 16000) and the magnitude of noise variables. The parallel algorithm has been implemented on an Origin2000 (32 processors) with MPI message passing library.

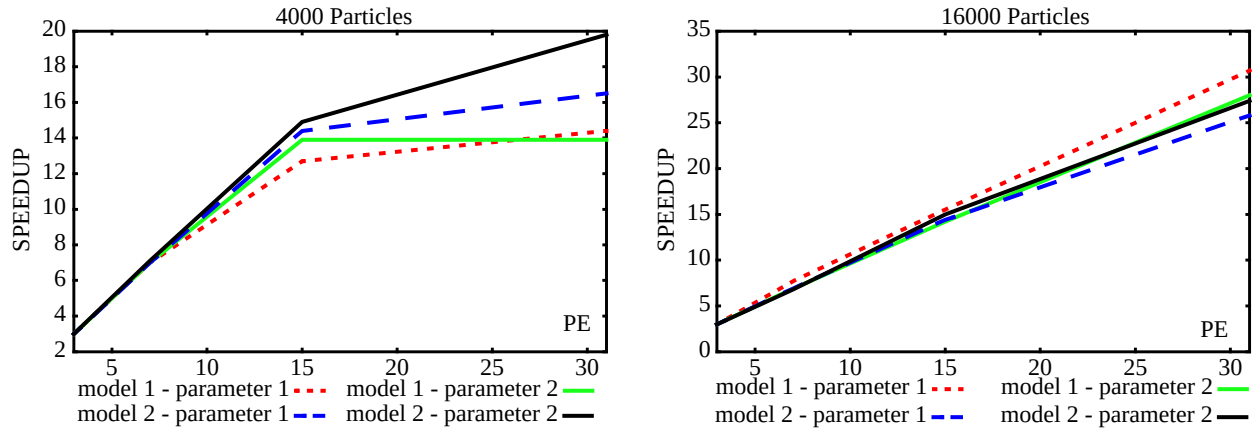

Fig. 5: Speedups obtained on the Origin2000 using 4000 and 16000 particles.

The results plotted on figure 5 show the very good scalability (near linear speedup) of the parallel algorithm for 16000 particles. The resulting parallel computation time per iteration is compatible with real-time applications (smaller than 1 ms).

We are developping new approaches to extend the scalability of the method to hundreds of processors to handle the larger number of particles (greater than 50000) necessary in real application problems such as the LORAN navigation signal processing [4] and RADAR defense applications with very low signal-noise ratio [3].

In parallel, we also investigate the capabilities of PC clusters with high-speed communication links (myrinet) to handle efficiently such parallel method.

5.References

[1] P. Del Moral and G. Salut. Filtrage non-linéaire: résolution particulaire à la Monte-Carlo. C.R. Acad. Sci. Paris, t.320, Série I:p1147-1152,1995.

[2] G. Rigal. Filtrage non-linéaire, résolution particulaire et application au traitement du signal. Thèse de doctorat, Université Paul Sabatier, Toulouse, juillet 1993.

[3] J-C Noyer and G. Salut. Optimal non-linear radar tracking of non-Gaussian manoeuvering targets. research report 96113, LAAS-CNRS, 1996.

[4] A. Monin. Evaluation de la limite de portée du LORAN-C par filtrage particulaire, JDA'99, Journées Doctorales d'Automatique, September 1999.

[5] O. Brun and J.M. Garcia. Parallélisation de la méthode de filtrage particulaire, research report LAAS-CNRS, September 1999.