Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network

Xiaohong Sheng, Yu-Hen Hu, Parameswaran Ramanathan
School of Electrical and Computer Engineering
University of Wisconsin-Madison, USA
WI, Madison, 53705

***Abstract*— Two novel distributed particle filters with Gaussian Mixer approximation are proposed to localize and track multiple moving targets in a wireless sensor network. The distributed particle filters run on a set of uncorrelated sensor cliques that are dynamically organized based on moving target trajectories. These two algorithms differ in how the distributive computing is performed. In the first algorithm, partial results are updated at each sensor clique sequentially based on partial results forwarded from a neighboring clique and local observations. In the second algorithm, all individual cliques compute partial estimates based only on local observations in parallel, and forward their estimates to a fusion center to obtain final output. In order to conserve bandwidth and power, the local sufficient statistics (belief) is approximated by a low dimensional Gaussian mixture model(GMM) before propagating among sensor cliques. We further prove that the posterior distribution estimated by distributed particle filter convergence almost surely to the posterior distribution estimated from a centralized bayesian formula. Moreover, a data-adaptive application layer communication protocol is proposed to facilitate sensor self-organization and collaboration. Simulation results show that the proposed DPF with GMM approximation algorithms provide robust localization and tracking performance at much reduced communication overhead.**

I. Introduction

In this paper, we present a novel framework for distributive, multiple-target localization and tracking in a wireless sensor network. This framework consists of two major components: (a) a family of distributive, sequential Bayesian joint localization and tracking algorithms based on the particle filter approach; and (b) an application-layer, data-adaptive sensor self-organization and communication protocol that facilitates efficient re-organization of sensors according to the predicted target trajectory. Specifically, important technical contributions made in this paper can be summarized as follows:

Distributive Particle Filter Algorithm Formulations – We proposed two novel distributive implementations of the particle filter algorithms, called DPF-I and DPF-II. With DPF-I, the importance weights of particles are updated in adjacent cliques of sensors sequentially. Each clique receives a partial estimate from its preceding clique, and update the partial estimate with local observations made within the clique. Then, it forwards the updated estimate to the next clique. Hence, the particles' importance weights are updated distributively in a pipelined fashion. With DPF-II, importance weights at each clique are estimated in parallel based only on local observations. These partial estimates are then forwarded over multiple-hop wireless channels to the final clique where the final estimate is computed.

Previously, in [3], a distributive particle filter algorithm was proposed that sought to update the complete particle filter on each individual sensor nodes. To facilitate this kind of distributive computing, it is required that sensor observations are independent. Clearly, DPF-I and DPF-II are quite different from this implementation in terms of the way distributive computation is performed. The distributed algorithms are run over a set of uncorrelated sensor cliques. These uncorrelated sensor cliques are dynamically constructed according to the moving trajectories and specific signal propagation characteristics. There are no independent assumption ahead. In addition, we propose specific learning algorithms for DPF-I and DPF-II. Yet, there are no specific training algorithm proposed in [3] to update the weights of the particle filter. Furthermore, we address the communication issues such as sensor self-organization and wireless bandwidth reduction.

Approximated Belief Function Representation – In a distributive particle filter implementation, partial estimate of importance weights, i.e. the belief estimates, must be communicated among sensor cliques via wireless communication channels. These requirements would defeat the purpose of distributive particle filter processing since it will require transmitting large amount of information. To reduce the communication burden, we approximate the belief estimates with a low dimension Gaussian mixture model (GMM). Instead of transmitting raw estimates of particles or the importance weights, we transmit GMM parameters. This approximation scheme significantly reduces the demand on the communication bandwidth while improves the estimation performance as GMM smooths out the particle distribution.

Data-Adaptive Application Layer Communication Protocol for Sensor Self-Organization and Collaboration – We propose a novel approach that can separate the targets into different independent groups. As such, localizing and tracking of the targets can be performed at each group independently using different distributed particle filters. An efficient communication strategy is proposed to collaborate the broadcasting communication with peer to peer communication to further reduce the communication burden.

The remaining of this paper is organized as follows: In section II, two approximated distributed particle filters are presented. Specifically, we present a way to construct disjoint uncorrelated cliques. Two distributed particle filter algorithms operate on the uncorrelated cliques. Local GMM is proposed to represent the distributed posterior distribution to reduce the communication burden. Section III provides several ways to reduce the DPF computation complexity and communication requirement. Specifically, we present a novel sensor self-organization approach to dynamically grouping targets and sensor clique generation. Data-adaptive application layer communication protocol is proposed to cooperate the broadcasting with peer to peer communication. A new backoff schedule is used to solve the interferences among the broadcasting sensor nodes. Simulations for target tracking using acoustic energy decay model are described in section IV. Very promising simulation results are given. A conclusion is given in section V. In appendix, we show that the posterior distribution estimated by the distributed particle filters convergence almost surely to the true posterior distribution estimated from the centralized Bayesian formula.

II. Distributed Particle Filter

A. Notation

Let $X = \{x_t, t \in N\} \in \Re^{n_x}$ be a Markov random process defined over a probability space $(\Omega, \mathcal{F}, P)$, where n_x is the dimension of X. The transition kernel of the Markov chain is defined as: $K_t(x, A) = P\left(x_{t+1} \in A | x_t = x\right)$

The observation model is: $y_t = h_t(x_t, w_t)$, where the observation noise w_t is assumed to be independent of the state vector x_t. The density of w_t is denoted by $g_t(\cdot)$. In certain conditions (refer to section II-C.1), the observation y_t can be divided into a set of M disjoint uncorrelated cliques, i.e. $y_t = \left[y_{t,1}, y_{t,2}, ..., y_{t,M}\right]^T$.

We denote $\hat{p}_t = P(x_t|Y_{0:t-1} = y_{0:t-1})$, $\pi_t = P(x_t|Y_{0:t} = y_{0:t})$, $\pi_{t,k} = P(x_t|Y_{0:t-1} = y_{0:t-1}, Y_{t,1:k} = y_{t,1:k})$. Similarly, we denote $\hat{p}_t^n$ and π_t^n as the prior and posterior probability estimated by centralized particle filter, and $\hat{p}_{t,k}^n$ and $\pi_{t,k}^n$ as the distributed prior and posterior probability respectively.

B. Centralized Particle Filter Tracking Algorithm

In [1], we proposed a centralized particle filter tracking algorithm. It consists of three steps: initialization, prediction and update.

During initialization, n random particles are uniformly drawn from the initial prior distribution π_0. According to the strong law of large numbers, $\lim_{n\to\infty} \pi_0^n \to \pi_0$ a.s.

During the prediction step, the new positions of the particles are computed based on the transition kernel, i.e., $\bar{x}_{t+1}^i \sim K_t\left(x_t^i, \cdot\right)$, where x_t^i and $\bar{x}_{t+1}^i$ are respectively, the current and predicted particle positions.

In the update step, the weight w_{t+1}^i for each particle is calculated based on the newly received observations, $w_{t+1}^i = g_{t+1}\left(\bar{x}_{t+1}^i\right) / \sum_{j=1}^n g_{t+1}\left(\bar{x}_{t+1}^j\right)$. The particles $\bar{x}_{t+1}^i$ are resampled according to a multinomial distribution with parameters w_{t+1}^i. The resampled particle x_{t+1}^i has mass $1/n$.

C. Distributed Particle Filter Algorithms

In a centralized Particle Filter (CPF), the posterior probability distribution is updated based on the entire set of new measurements at all sensors in the sensor network. This would require tremendous communication overhead to transmit the raw sensor data to a centralized location.

A distributed Particle Filter (DPF) differs from the CPF in the update step. In particular, a DPF algorithm updates the state distribution sequentially or in parallel using only local measurements. In [3], a DPF algorithm is proposed to update the state distribution sensor by sensor. Coates' method consists of the following features: (a) Each sensor node will maintain a separate particle filters, *synchronized* using the same set of prior distribution. (b) It is assumed that observations at each sensor nodes are independent so that the likelihood function can be factored into products of partial likelihood functions. (c) Each partial likelihood function will be updated at individual sensors using only local observations and partial likelihood functions estimated in a preceding sensors. (d) The partial likelihood function will be represented by an some kind of parametric model whose parameters requires training using locally generated particles. Neither the parametric model nor the training method is not specified. (e) The final importance distribution then will be back-propagated to all preceding sensors and a new set of particles will be generated at each sensor using this final distribution. The resulting distributed particle filter tracking results may not be the same as the centralized particle filter. Hence, it is difficult to predict its theoretical performance.

1) Uncorrelated sensor cliques in sensor network: We argue that in a densely deployed sensor network, the assumption of independent observations at individual sensors would be unrealistic. Instead, in this work, we propose to partition sensors into groups (cliques) based on sinal propagation model.

Information sensed by the nodes in sensor network is governed by the events in the sensor field. Each signal received by the sensor located at (a_x, a_y) corresponds to a space-time signal with spatial bandwidth B_{a_x}, B_{a_y} and temporal bandwidth B. The coherence distance $D_{a_x} = \frac{1}{B_{a_x}}$ and $D_{a_y} = \frac{1}{B_{a_y}}$ denotes the spatial dimension over which the signals are strongly correlated [4]. $D_{a_x} \times D_{a_y}$ determines spatial coherence region, i.e., clique. The signals are approximately uncorrelated in distinct clique, i.e. $\pi_t \approx \prod_{k=1}^M \pi_{t,k}$, where M is the number of uncorrelated cliques.

For signal produced by point source such as targets, the spatial signal bandwidth is determined by the temporal bandwidth via the speed of signal propagation. For isotropic spatial propagated signal, $y(a_x, a_y, t) = y(0, 0, t - \frac{\sqrt{{a_x}^2 + {a_y}^2}}{v})$, where v is the speed of propagation. The spatial bandwidth and coherence distance in the radial coherence dimension are:

$$B_r = \frac{B}{v}, \quad D_r = \frac{1}{B_r} = \frac{v}{B} \tag{1}$$

where $r = \sqrt{{a_x}^2 + {a_y}^2}$, B is the signal temporal bandwidth and v is the propagation speed.

Define the cluster distance $Cd(i,j)$ as:

$$Cd(i,j) = min.d(s_x, s_y) \ \ \forall s_x \in C(i) \ \text{ and } \ s_y \in C(j)$$

The sensors in the network field can be clustered into disjoint uncorrelated sensor cliques based on the minimum spanning tree structure [5] obtained using hierarchical cluster algorithm. Initially each individual sensor is defined as a cluster. Two clusters can be merged into one if the condition $Cd_{i,j} \le D_r$ is satisfied.

Because acoustic signal propagation speed is relatively slow, the coherence distance of the acoustic signals is quite small. For example, D_r is less than $15M$ for all acoustic signals with bandwidth bigger than $20HZ$. This is a very loose condition because most of the acoustic signals have the bandwidth bigger than $20HZ$. This feature provides a good chance to divide the sensor field into uncorrelated small cliques because, as long as some of the sensors have distances more than D_r, we have chances to cluster the sensors into a set of disjoint uncorrelated cliques.

2) Distributed Particle Filter Algorithm I: DPF-I: Let $x_{t-1,M}^i$ and $x_{t,0}^i$ be respectively, the current and predicted particle positions, and $K_t\left(x_{t-1,M}^i, \cdot\right)$ be the transition kernel. Then the predicted particles has a distribution:

$$x_{t,0}^i \sim K_t\left(x_{t-1,M}^i, \cdot\right).$$

Distributed algorithm 1 updates the posterior distribution sequentially by a sequence of local particle filters, i.e.

For $k = 1$ to M,

Denote $\bar{x}_{t,k}^i = x_{t,k-1}^i$,

Compute

$$w_{t,k}^i = n g_{t,k}\left(\bar{x}_{t,k}^i\right) / \sum_{j=1}^n g_{t,k}\left(\bar{x}_{t,k}^j\right) \tag{2}$$

where $g_{t,k}(\bar{x}_{t,k}^i) = P(w = y_{t,k} - h_{t,k}(\bar{x}_{t,k}^i))$.

Resample particles by replacing $\bar{x}_{t,k}^i$ with a number of offspring $n_{t,k}^i$ according to a multinomial distribution of parameters $w_{t,k}^i$

and $\sum_{i=1}^{n} n_{t,k}^{i} = n$. Denote the resampled particles as $x_{t,k}^{i}$, the approximated posterior distribution before and after resampling are:

$$\begin{aligned} \bar{\pi}_{t,k}^{n} &= \frac{1}{n} \sum_{i=1}^{n} w_{t,k}^{i} \delta_{\{\bar{x}_t^i\}} \\ \pi_{t,k}^{n} &= \frac{1}{n} \sum_{i=1}^{n} n_{t,k}^{i} \delta_{\{\bar{x}_t^i\}} = \frac{1}{n} \sum_{i=k}^{n} \delta_{\{x_{t,k}^i\}} \end{aligned} \tag{3}$$

It can be proved that $\lim_{n\to\infty} \pi_{t,M}^{n} \to \pi_t$ almost surely and the convergence rate is $\frac{M}{\sqrt{n}}$. Detailed proof is deferred to Appendix A.

3) Distributed Particle Filter Algorithm II: DPF-II: Instead of updating the posterior distribution sequentially using a sequence of local particle filters, the DPF-II runs the local particle filters in parallel at individual sensor cliques. Then the final posterior distribution can be computed based on these local sufficient statistics. Specifically, at every sensor clique, compute the following concurrently:

$$w_{t,k}^{i} = \frac{g_{t,k}\left(\bar{x}_t^i\right)}{\sum_{j=1}^{n} g_{t,k}\left(\bar{x}_t^j\right)} \tag{4}$$

where $\bar{x}_t^i \sim K_{t-1}(x_{t-1}^i, .)$

Note that $g_{t,1:M}\left(x_t\right) = \prod_{k=1}^{M} g_{t,k}\left(x_t\right)$. Since signals received at different cliques are uncorrelated to each other, the updated weight at time t is:

$$\begin{aligned} {}^{i}_{t} &= \frac{{}_{t,1:M}\left(^{-i}_{t}\right)}{\sum_{j=1}^{n} {}_{t,1:M}\left(^{-j}_{t}\right)} = \frac{{}_{t,1}\left(^{-i}_{t}\right) \prod_{k=2}^{M} {}_{t,k}\left(^{-i}_{t}\right)}{\sum_{l=1}^{n} {}_{t,1}\left(^{-l}_{t}\right) \sum_{j=1}^{n} \frac{\prod_{k=1}^{M} g_{t,k}\left(\bar{x}_t^j\right)}{\sum_{l=1}^{n} g_{t,1}\left(\bar{x}_t^l\right)}} \\ &= \frac{{}^{i}_{t,1} \prod_{k=2}^{M} {}_{t,k}\left(^{-i}_{t}\right)}{\sum_{j=1}^{n} {}^{j}_{t,1} \prod_{k=2}^{M} {}_{t,k}\left(^{-j}_{t}\right)} = \quad = \frac{\prod_{k=1}^{M} {}^{i}_{t,k}}{\sum_{j=1}^{n} \prod_{k=1}^{M} {}^{j}_{t,k}} \end{aligned} \tag{5}$$

Eq. (5) shows that DPF-II yields identical posterior estimates as that of the centralized particle filtering algorithm. The convergence rate for DPF-II is $1/\sqrt{n}$.

D. Convergence of DPF Algorithms

Theorem 1 (Convergence of DPF): $\lim_{n\to\infty} \pi_{t,M}^{n} \to \pi_t$ almost surely.

The proof of this theorem is given in Appendix A.

The convergence rate for DPF-I and DPF-II algorithms are $M/\sqrt{n}$ and $1/\sqrt{n}$ respectively. Details of convergence rate will be reported in a manuscript under preparation [10].

E. Gaussian Mixture Approximation

Both DPF-I and DPF-II require belief propagation, which need to transmit more bits than raw observation data. In this paper, we propose to approximate the estimated distribution from the DPFs with a low dimensional Gaussian Mixture Model (GMM).

$$\pi_{t,k}^{n} \simeq \hat{\pi}_{t,k}^{n} = \sum_{m=1}^{c} \hat{\lambda}_{t,k}^{m} \mathcal{N}(\hat{\mu}_{t,k}^{m}, \hat{\sigma}_{t,k}^{m}) \tag{6}$$

where c is the number of Gaussian mixtures.

Extended Kalman filter and GMM have been used for non Gaussian or nonlinear system whose analytic solutions are not available. Yet, both the algorithms fail to take into account all the salient statistical features of the processes under consideration, leading quite often to poor results [2].

With the GMM model, the belief can be propagated through the transmission of the parameters of $\text{GMM}\left(\hat{\lambda}_{t,k}^{m}, \hat{\mu}_{t,k}^{m}, \hat{\sigma}_{t,k}^{m}\right)$ rather than the particles. This approximation promises dramatic reduction of the communication overhead of the DPF algorithms. The parameters of GMM can be estimated using EM algorithm which is summarized as follows:

Expectation step:

$$\hat{\lambda}_{t,k}\left(m|x_{t,k}^{i}\right) = \frac{\mathcal{N}\left(x_{t,k}^{i}, \hat{\mu}_{t,k}^{m}, \hat{\sigma}_{t,k}^{m}\right) \hat{\lambda}_{t,k}^{m}}{\sum_{l=1}^{c} \mathcal{N}\left(x_{t,k}^{i}, \hat{\mu}_{t,k}^{l}, \hat{\sigma}_{t,k}^{l}\right) \hat{\lambda}_{t,k}^{l}}$$

$$\hat{\lambda}_{t,k}^{m} = \frac{1}{n} \sum_{i=1}^{n} \hat{\lambda}_{t,k}\left(m|x_{t,k}^{i}\right) \tag{7}$$

Maximization step:

$$\hat{\mu}_{t,k}^{m} = \frac{\sum_{i=1}^{n} x_{t,k}^{i} \hat{\lambda}_{t,k}\left(m|x_{t,k}^{i}\right)}{\sum_{i=1}^{n} \hat{\lambda}_{t,k}\left(m|x_{t,k}^{i}\right)}$$

$$\hat{\sigma}_{t,k}^{m2} = \frac{\sum_{i=1}^{n} \left(x_{t,k}^{i} - \hat{\mu}_{t,k}^{m}\right)^2 \hat{\lambda}_{t,k}\left(m|x_{t,k}^{i}\right)}{\sum_{i=1}^{n} \hat{\lambda}_{t,k}\left(m|x_{t,k}^{i}\right)} \tag{8}$$

The number of c can be selected as a function of the number of targets which we assume is known in this work. We note that the EM iterations converge to a local minimum solution. However, in our simulation, the results seem to be quite robust to the selection of initial conditions.

An implementation detail that needs to be emphasized is that for DPF-II, a copy of the local particle filter algorithm will run in parallel at each sensor clique. The clique center passes the local GMM to its neighborhood, and calculates:

$$\begin{aligned} \hat{\pi}_{t,j+1}^{n} &= \prod_{k=j}^{j+1} \sum_{m=1}^{c} \hat{\lambda}_{t,k}^{m} \mathcal{N}\left(\hat{\mu}_{t,k}^{m}, \hat{\sigma}_{t,k}^{m}\right) \\ &= \frac{\sum_{m_j=1}^{c} \sum_{m_{j+1}=1}^{c} \prod_{k=j}^{j+1} \hat{\lambda}_{t,k}^{m_k} \mathcal{N}\left(\hat{\mu}_{t,k}^{m_k}, \hat{\sigma}_{t,k}^{m_k}\right)}{\sum_{m_j=1}^{c} \sum_{m_{j+1}=1}^{c} \prod_{k=j}^{i+1} \hat{\lambda}_{t,k}^{m_k}} \end{aligned} \tag{9}$$

$$\prod_{k=j}^{j+1} \hat{\lambda}_{t,k}^{m_k} \mathcal{N}\left(\hat{\mu}_{t,k}^{m_k}, \hat{\sigma}_{t,k}^{m_k}\right) = \left(\prod_{k=j}^{j+1} \hat{\lambda}_{t,k}^{m_k}\right) \mathcal{N}(\bar{\mu}_t, \bar{\sigma}_t) \tag{10}$$

$$\bar{\sigma}_t = \left(\sum_{k=j}^{j+1} \left(\hat{\sigma}_{t,k}^{m_k}\right)^{-1}\right)^{-1}, \quad \bar{\mu}_t = \bar{\sigma}_t \sum_{k=j}^{j+1} \frac{\hat{\mu}_{t,k}^{m_k}}{\hat{\sigma}_{t,k}^{i_k}}$$

$$m_k \in [1:c]$$

Use k-mean algorithm to cluster c^2 mixture to c mixture of Gaussian and denote the clustered GMM as $\hat{\pi}_{t,j+1}^{n}$ and pass the clustered GMM parameters to the next clique.

The DPF-I and DPF-II with GMM approximation algorithms are summarized in Table 1 and Table 2.

F. Comparison of the two algorithms

We have shown that both DPF-I and DPF-II algorithms convergence almost surely to the true posterior probability. The convergence rate of DPF-II is faster than that of DPF-I. From Eq.(5), we note that the posterior probability distribution estimated using DPF-II should be identical to that estimated using the CPF algorithm. Also, from Eq.(19) and Eq.(21), it appears that DPF-I would require more particles than DPF-II to achieve the same performance. However, the communication required by DPF-II is higher than DPF-I.

TABLE I

DPF-I WITH GMM SMOOTHING ALGORITHM

1) Initialization:
   - Draw n samples $^{i}_{0}$ uniformly from distribution $_0$
2) Repeat at every time instant t:
   - If $\neq 1$, draw samples $^{i}_{t-1}$ from GMM with distribution $\hat{}^{n}_{t-1}$
   - Predict $\bar{}_{t,0}$ according to the transition kernel $_t(\ ^{i}_{t-1}\ \cdot)$.
   - For $= 1:$
     - Compute $^{i}_{t,k}$ according to Eq. (2)
     - Resample $\bar{}^{i}_{t,k}$ using multinomial distribution with weights $^{i}_{t,k}$, denote the new particles as $^{i}_{t,k}$
     - Approximate $^{n}_{t,k}$ with $\hat{}^{n}_{t,k}$ using Gaussian mixture model.
     - Pass parameters of local GMM to the center node of its neighbor set, $(\ = \ +1)$
   - $\hat{}^{n}_{t} = \hat{}^{n}_{t,M}$
     $(\hat{}_t) = \sum_{m=1}^{c} \hat{}^{m}_{t,M} \hat{}^{m}_{t,M}$

TABLE II

DPF-II WITH GMM SMOOTHING ALGORITHM:

1) Initialization:
   - Draw n samples $^{i}_{0}$ uniformly from distribution $_0$
2) Repeat at every time instant t:
   - Run the following algorithm in parallel at each clique
     - If $\neq 1$, draw samples $^{i}_{t-1}$ from GMM with distribution $\hat{}^{n}_{t-1}$
     - Predict $\bar{}_{t}$ according to the transition kernel $_t(\ ^{i}_{t-1}\ \cdot)$.
     - Compute $^{i}_{t,k}$ according to Eq.(4)
     - Approximate $^{n}_{t,k}$ with $\hat{}^{n}_{t,k}$ using Gaussian mixture model.
   - Pass parameters of local GMM clique by clique
     For j=1:M
     - Compute $\hat{}^{n}_{t,j+1}$ according to Eq. (9) (10) which will produce ^{2} mixtures of Gaussian
     - Use k-mean algorithm to cluster ^{2} mixtures to c- mean mixtures
     - Denote the clustered GMM as $\hat{}^{n}_{t,j+1}$ and pass the clustered GMM parameters to its neighboring clique.
   - $(\hat{}_t) = \sum_{m=1}^{c} \hat{}^{m}_{t} \hat{}^{m}_{t}$

III. SENSOR SELF-ORGANIZATION AND DATA ADAPTIVE COMMUNICATION -SYSTEM DESIGN

A. Computation Complexity

Computation complexity of DPF increases: 1) exponentially with the number of jointed sources; 2)linearly with the number of states; and 3) linearly with the number of particles.

We proposed a way to reduce the number of states by eliminating the nuisance parameters based on a local ML estimator. The number of particle filters can be reduced through a deterministic kernel placement [2]. The number of jointed sources is reduced by separating the sources into different independent groups and merging the very close sources according to their relative interference. DPFs with GMM approximation run independently at each group. Details of source grouping and merging will be delivered in the [10] [11].

Fig. 1 shows the two groups of sources we have separated at time t. The left group is composed of 5 sources. Four of them are merged into two single sources. Therefore, they can be estimated as three joint sources. The right group has three joint sources. The computation complexity is reduced from $O(D^8)$ to $O(2D^3)$, where D is the dimension of location position

Table III shows the pseudo code of the hierarchical cluster algorithm for target grouping.

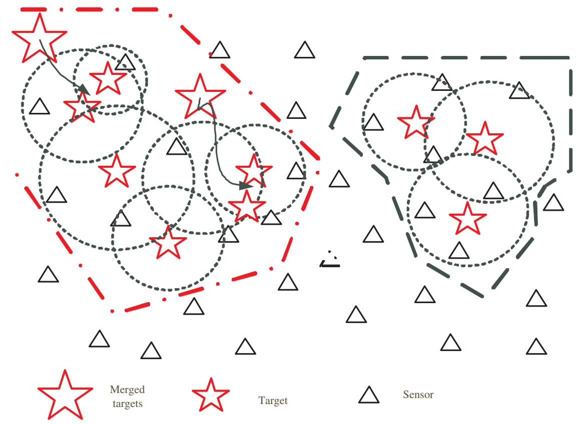

Fig. 1. Source grouping and merging

TABLE III

SOURCE GROUPING WITH HIERARCHICAL CLUSTERING ALGORITHM

1) Initial step,
   - Set each target as a group,
     $$_i = [\hat{}_{ix}\ \hat{}_{iy}] \quad = 1\ 2$$
     where $[\hat{}_{ix}\ \hat{}_{iy}]$ is the predicted target location at next time interval.
   - Set the number of group as K, i.e. $_G =$
2) Iteration step,
   for $= 1:\ _G$
   - for $= \ +1:\ _G$
     - if the minimum distance between two group (i,j) doesn't satisfy the source separation condition [10] [11]:
       * Merge $_i$ $_j$
       * if $_G$
         · Set $_j = \ _{N_G}$, $_G = \ _G - 1$
       * else
         · Set $_G = \ _G - 1$

B. Dynamic Sensor Self-Organization

Sensor self-organization has been investigated to improve signal processing and communication efficiency in wireless sensor network[12] [13] [14]. In this paper, we propose a novel sensor self organization algorithm based on predicted moving target trajectories and relative signal interference. With sensor self organized, the targets can be separated and be localized independently at each group with reduced targets at each group. Hence, the computation complexity can be reduced exponentially.

One unique feature of our DPF algorithm is that our DPFs run over a sequence of uncorrelated sensor cliques. These uncorrelated sensor cliques are generated dynamically according to the predicted target trajectories.

According to our DPF design, the final clique center at the current time interval has the latest complete state information. It predicts state distribution and target locations at next time interval. Based on the predicted target locations, the current clique center can figure out the sensors that are closest to the predicted target locations. Using the grouping algorithm described in Table III, we can determine 1) which targets can be grouped together and which targets can be separated, and 2) the effective sensors at each group.

The effective sensors in the group can be clustered into different uncorrelated cliques according to what we described in section II-C.1.

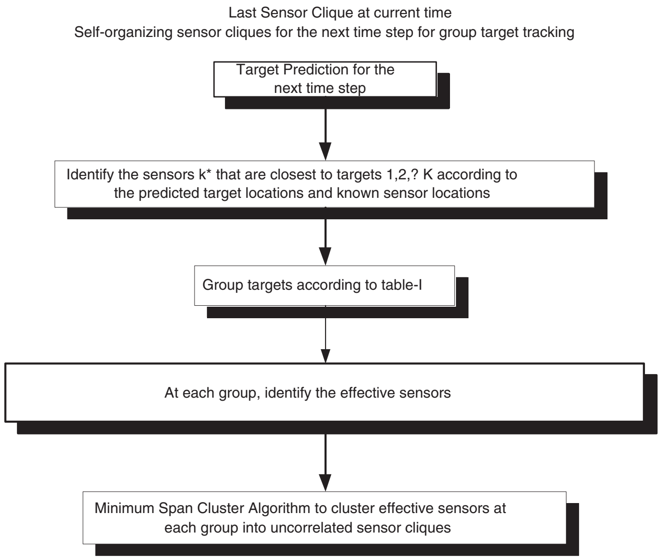

Fig. 2. Dynamic sensor self-organization for next time step operated at the last sensor clique at current time step

After clustering the effective nodes at each group into uncorrelated sensor cliques, we pick the center node of each clique as the master node. Other effective nodes are defined as slave nodes. Fig. 2 shows the flowchart for dynamic sensor self-organization and group target tracking using our distributed particle filter.

C. Data-Adaptive Communication

Data-adaptive application layer communication protocol is needed for sensor self-organization and collaboration in order to dynamic target grouping and sensor clique generation. Data adaptive communication collaborates the broadcasting communication with peer to peer communication. A new backoff schedule is used to solve the interferences among the broadcasting nodes.

Message being exchanged in wireless sensor network for group targets tracking using distributed particle filtering includes:

1) Message exchange in the current cliques
   - message exchange within the clique.
   - message exchange among the cliques.
2) Message exchange for constructing next cliques
   - Notification of selected clique centers (master nodes) for next time interval.
   - Selected master nodes for next time interval activate their slave nodes

Message Exchange in the Current Cliques

To reduce the communication cost, the slave nodes within the clique broadcast their measurements. The master node only accepts the packets broadcasted by its own slave nodes and ignores the others. The broadcasting power can be low since the clique size is small. Under the constraint that slave node transmission signal can be heard by its own master node, the transmission power should be as low as possible so that the interference among the broadcasting signals can be minimized. Meanwhile, low transmission power design also saves the master node power. This is because the master nodes of the cliques will have less probability to hear the transmitted signals from other cliques. Hence, the master node won't take their power to decode the unnecessarily received packets.

Due to the considerable correlation and redundancy of information within a clique, the slave nodes are not required to guarantee delivery of their measurements to the master node. As a result, it is not necessary to use acknowledgements and retransmissions for this communication. Each slave node will be set a backoff time before broadcasting to avoid the broadcasting interference. The backoff time is designed to be inverse proportional to its SNR.

Peer to peer communication is applied to exchange the local GMM parameters among the cliques. A reliable peer-to-peer communication protocol will be used to ensure that the information is delivered without loss. The protocol relies on acknowledgements and retransmissions to support the reliability.

Message Exchange for Clique Construction for Next Time Interval

The clique center of last sensor clique at current time step dynamically organizes sensors and generates sensor cliques for the next time step based on the predicted target trajectories. The master node of each sensor clique is selected as the center node of the clique. After that, the current clique master nodes are responsible to notify the selected master nodes of the sensor cliques of next time step. Because the notification to the selected master nodes for next time interval is very important, we apply peer to peer communication. A reliable peer-to-peer communication protocol will be used to ensure that the information is delivered without loss. If there is no ack from a selected master node, a nearby sensor node to the un-acked master node will be selected as a replacement and it will be notified. The notified master nodes for the next time interval will broadcast a 'wake-up' signal to its slave nodes.

D. Communication Requirement

The communication cost in bytes per time step is:

$$N_e N_f + 3cN_c N_s N_f + N_n N_i \quad \text{for DPF-I GMM}$$

$$N_e N_f + 3cN_c N_s N_f + N_n(N_i + 3cN_s N_f) \quad \text{for DPF-II GMM}$$

where N_f and N_i are the number of bytes to represent the float data and integer data respectively. N_e is the total number of slave nodes within the cliques, N_c is the total number of hops that master nodes transmit their parameters among the cliques. N_n is the number of hops to notify the clique centers at next time interval. N_s is the number of states and c is the number of mixers we applied.

Since $max(N_e, N_c, N_n) = N$, where N is the number of sensors in the sensor field, the maximum communication cost in bytes for DPF-I and DPF-II are O(N). DPF-I has less communication requirement than DPF-II.

The least communication for CPF algorithm is when the sensors are uniformly deployed and the center node is at the center of the sensor field. The communication in bytes is: $\sqrt{N}(1 + 2 + ... + \frac{\sqrt{N}}{2})N_f = O(N^{3/2})$. The worst case for the CPF algorithm is when the center node is at the end of the sensor field. The communication in bytes is: $(N - 1 + N - 2 + ... + 1)N_f = O(N^2)$. Hence, the communication requirement is reduced using DPF algorithm.

IV. SIMULATION ON TWO TARGETS TRACKING IN WIRELESS SENSOR NETWORK

A. Acoustic Energy Based Source Localization

We apply acoustic energy based source localization model that we have proposed in [6] as our measurement model.

$$y_i(t) = \gamma_i \sum_{l=1}^{L} \frac{s_l(t)}{\| \boldsymbol{\rho}_l(t) - \mathbf{r}_i \|^2} + \varepsilon_i(t) \tag{11}$$

$$i = 1, 2, ..., N$$

where L is the number of targets (assumed to be known), $y_i(t)$ is the acoustic energy received by the i^{th} sensor. $\varepsilon_i(t)$ is a perturbation term that summarizes the net effects of background additive noise and the parameter modelling error. γ_i and $\mathbf{r}_i$ are the gain factor and location of the i^{th} sensor respectively, $s_l(t)$ and $\boldsymbol{\rho}_l(t)$ are the energy emitted by the l^{th} source (measured at 1 meter from the source) and its location during the t^{th} time interval. N is the number of sensors in the activated region. The mean and variance of each $\varepsilon_i(t)$ can be empirically estimated from constant false alarm (CFAR) detector.

Collaborating the particle filter with maximum likelihood algorithm, we eliminate the nuisance parameter[10][11]. The reduced state updating weight can be expressed as:

$$w_{t,k}^{i} = \begin{cases} (\mathbf{z}_{t,k}|^{-}{}_{t,k}(\ )) = \mathbf{z}_{t,k}^{T}\mathbf{P}_{t,k}(\bar{x}_{t,k}(i))\mathbf{z}_{t,k} & \text{for DPF-I} \\ (\mathbf{z}_{t,k}|^{-}{}_{t}(\ )) = \mathbf{z}_{t,k}^{T}\mathbf{P}_{t}(\bar{x}_{t}(i))\mathbf{z}_{t,k} & \text{for DPF-II} \end{cases} \quad (12)$$

where $\mathbf{z}_{t,k}$ is the normalized energies at the k^{th} sensor clique. $\mathbf{P}_{t,k}$ is a projection function of $\bar{x}_{t,k}(i)$. $\mathbf{P}_t$ is a projection function of $\bar{x}_t(i)$. For DPF-I algorithm, $\bar{x}_{t,k}(i)$ is the i^{th} prior sample distributed as $\bar{x}_{t,k}(i) \sim P(x_t|\mathbf{z}_{1:t-1}, \mathbf{z}_{t,1:k})$. For DPF-II algorithm, $\bar{x}_t(i)$ is the i^{th} prior sample distributed as $\bar{x}_t(i) \sim P(x_t|\mathbf{z}_{1:t-1})$. η is defined to normalize the posterior probability.

In this work, the state transition is modeled by a a Gaussian Markov jump process:

$$\begin{aligned} \theta_l(t) &= \theta_l(t-1) + d\theta_l(t) \\ \mathbf{a}_l(t) &= \zeta_l(t)[cos\theta_l(t) \ \ sin\theta_l(t)]^T \\ \mathbf{u}_l(t) &= \mathbf{u}_l(t-1) + \mathbf{a}_l(t)T \\ \boldsymbol{\rho}_l(t) &= \boldsymbol{\rho}_l(t-1) + \mathbf{u}_l(t-1)T + \frac{1}{2}\mathbf{a}_l(t)T^2 \end{aligned} \quad (13)$$

where $\boldsymbol{\rho}_l(t)$ stands for the location of source l, $\mathbf{u}_l(t)$ is the velocity of source l and $\mathbf{a}_l(t)$ is the acceleration of the source l. $\theta_l(t)$ is the acceleration direction. The change of the acceleration direction $d\theta_l(t)$ is modeled as a Gaussian Markov jump model:

$$d\theta_l(t) = v_l(t) + \begin{cases} -15^o & \text{with p.d.f. } 0.1 \\ 0^o & \text{with p.d.f. } 0.8 \\ 15^o & \text{with p.d.f. } 0.1 \end{cases} \quad (14)$$

where noise $v_l(t)$ is assumed to be a Gaussian random variable, $v_l(t) \sim N(0,1) * 5^o$.
The absolute acceleration velocity $\zeta_l(t)$ is assumed to be uniformly distributed over $[-A_{max} \ A_{max}]$ where A_{max} is the maximum acceleration rate; and T is the time interval.

B. Simulation

Simulations have been conducted to compare the performance of target tracking using Distributed Particle Filter with GMM approximation (DPF-GMM), Distributed Particle Filter (DPF) and Centralized Particle Filter (CPF).

Two targets move in a sensor field of size $100 \times 100M^2$. The sensor node labels and locations are depicted in Fig.3. The initial velocity of each targets is assumed to be 20 m/s. The velocity and acceleration for the two targets are changed according to our state transition model. The background noise level is set as $\sigma_i = 2$ for all sensors in the sensor field. The number of particles is chosen to be 500. 500 repeated trials are simulated for each sequential running point.

The bias and variance of target location estimation are shown in Fig. 4∼Fig. 5. From the simulation results, we can see that both DPF algorithms with GMM approximation have comparable performance as the CPF algorithm. The estimation error doesn't accumulate with

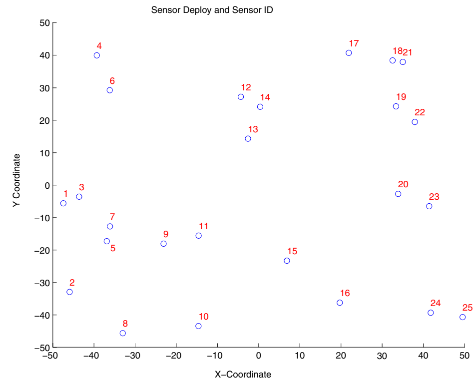

Fig. 3. Sensor Placement and Sensor ID for Target Localization and Tracking

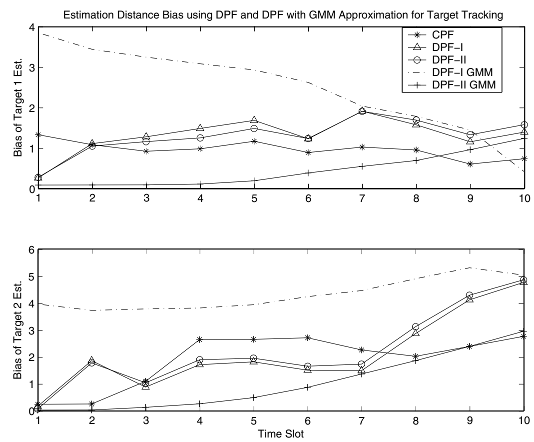

Fig. 4. Estimation bias using CPF, DPF and DPF with GMM Approximation for sequential target tracking

time increment. Furthermore, DPF with GMM approximation can produce better performance than DPF only. As our prediction in section II-C.3, DPF-II has better performance than DPF-I.

In section II-C.3, we have shown that DPF-II yields identical posterior estimates as that of the centralized particle filtering algorithm (Refer to Eq.5). Furthermore, GMM approximation regularizes the widely distributed particles and makes these particles distributed more smoothly. Hence, from the simulation results, we find that DPF-II with GMM approximation has even better performance than that of the CPF algorithm. Therefore, DPF with GMM approximation can not only reduces the communication significantly, but also improves the algorithm performance.

The communication requirements at each time step using DPF-I GMM and DPF-II GMM algorithms are shown in Table IV. Here we assume that the float data is represented by 16 bits, the integer is represented by 8 bits, two mixer of gaussian is used to represent the approximated distribution, and 4 states of GMM parameters(2 states for location and 2 states for velocity) are transmitted among the clique centers. This table shows that the maximum communication requirement at each time interval is less than 160 bytes for DPF-I with GMM approximation and 310 bytes for DPF-II with GMM approximation. The total transfer bytes is reduced significantly from 5520 bytes for CPF algorithm to 1000 bytes for DPF-I with GMM approximation and 2200bytes for DPF-II with GMM approximation.

Overall, the simulation shows that DPF with GMM approximation

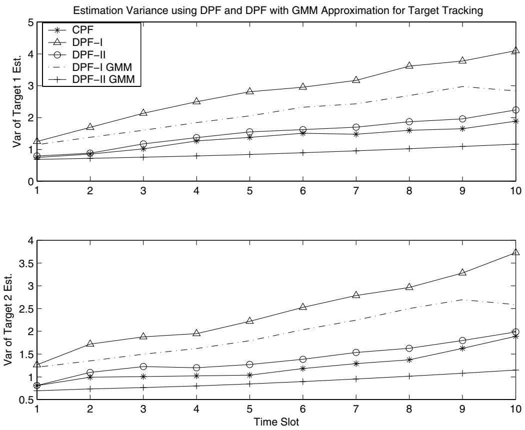

Fig. 5. Estimation variance using CPF, DPF and DPF with GMM Approximation for sequential target tracking

TABLE IV
COMMUNICATION REQUIREMENT USING DPF-I AND DPF-II WITH GMM ALGORITHMS AND CPF ALGORITHM

Time	N_1	N_2	N_3	Total transfer in bytes (DPF-I)	Total transfer bytes (DPF-II)	Total transfer bytes (CPF)
1	5	2	2	108	204	552
2	5	2	3	109	253	552
3	6	3	3	159	303	552
4	4	3	3	155	251	552
5	4	3	2	154	298	552
6	2	1	3	55	199	552
7	8	3	3	163	307	552
8	6	1	2	62	158	552
9	5	0	2	12	108	552
10	5	0	2	12	108	552
$Total$ $(bytes)$	50	18	25	989	2189	5520

N_1: Number of slave nodes within the clique.
N_2: Number of hops for master node transmits the GMM parameters
N_3: Number of hops for notifying the master nodes of cliques at next time period.

algorithms provide robust tracking performance while reduce the communication significantly compared to the CPF algorithm. DPF-I GMM approximation requires much less communication requirement than DPF-II GMM approximation and almost the same estimation performance as the DPF-II GMM approximation. Hence, DPF-I GMM approximation is superior to DPF-II GMM approximation.

V. CONCLUSION

In this paper, we proposed two distributed particle filters cooperated with GMM approximation algorithms to track the multiple targets in wireless sensor network. The algorithms can run distributively at the local sensor cliques. By approximating the local sufficient statistic (belief) with GMM, the communication burden is dramatically reduced while the performance is maintained almost the same as or even better than the CPF algorithm. In addition, we also proved that the states posterior distribution estimated by DPF convergence almost surely to the true posterior distribution formulated by the centralized sequential bayesian estimation. The computation and communication burden of DPF with GMM approximation algorithm can further be reduced using the target grouping and separation approaches and state reduction algorithms. Data adaptive application layer communication protocol is proposed for sensor self-organization and collaboration. The simulation shows that the DPF with GMM approximation algorithms provide high localization and tracking performance and much reduced communication requirements. Reducing the number of particles through a deterministic kernel placement is under development.

REFERENCES

[1] X. Sheng and Y.H. Hu, "Sequential acoustic energy based source localization using particle filter in a distributed sensor network", ICASSP2004, III-972-975
[2] A.Doucet, N. de Freitas, and N. Gordon, editors. Sequential Monte Carlo Methods in Practice. Springer-Verlag, 2001
[3] Mark Coates, "Distributed particle filters for sensor networks" Information Processing in Sensor Networks, IPSN2003, Springer, pp:99-107, 2004
[4] A. Sayeed, "A Statistical Signal Modeling Framework for Sensor Networks" , Technical Report, 2004
[5] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Second Edition, MIT Press, September 2001, ISBN 0-262-03293-7
[6] X. Sheng and Y.H. Hu, "Energy based source localization", Information Processing in Sensor Networks, IPSN2003, Springer, pp:285-300, 2003
[7] Erik B. Sudderth, etal, "Nonparametric Belief Propagation", Proceedings of the 2003 IEEE conference on computer vision and pattern recognition, v.1, pp605-612
[8] Alexander T. lhler, etal," Nonparametric belief propagation for self-calibration in sensor networks", IPSN2004
[9] Xiaohong Sheng and Yu-Hen Hu,"Maximum Likelihood Multiple-Source Localization Using Acoustic Energy Measurements with Wireless Sensor Networks", IEEE Trans. on Signal Processing, Jan. 2005, Vol. 53, pp44-53.
[10] Xiaohong Sheng, Parameswaran Ramanathan and Yu-Hen Hu, "Target localization and tracking using distributed particle filtering with GMM approximation in wireless sensor network, Part I: Signal Processing Algorithms", IEEE Trans. on Signal Processing, to be submitted.
[11] Xiaohong Sheng, Parameswaran Ramanathan and Yu-Hen Hu, "Target localization and tracking using distributed particle filtering with GMM approximation in wireless sensor network, Part II: Sensor Self-Organization and Data Adaptive Communication", to be submitted.
[12] Juan Liu, Jie Lie, James Reich, Patrick Cheung, Feng Zhao, "Distributed group management for track initiation and maintenance in target localization applications", IPSN2003, pp113-128
[13] Liang Zhao; Xiang Hong; Qilian Liang,"Energy-efficient self-organization for wireless sensor networks: a fully distributed approach", Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE , Vol: 5, 2004, Pp:2728 - 2732
[14] Zhihui Chen; Khokhar, A.,"Self organization and energy efficient TDMA MAC protocol by wake up for wireless sensor networks", Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on , 4-7 Oct. 2004 Pp:335 - 341

APPENDIX A

Let $f \in \mathbb{B}(\Re^{n_x})$ be any continuous bounded function defined on $\Re^{n_x}$, then

$$E\left[(\pi_t f - \pi_t^n f)^4\right] \le E\left[E\left[\left(\pi_t f - \bar{\pi}_{t,M}^n f\right)^4 |x_t\right]\right] + E\left[E\left[\left(\bar{\pi}_{t,M}^n f - \pi_{t,M}^n f\right)^4 |x_t\right]\right] \quad (15)$$

$$E\left[E\left[\left(\bar{\pi}_{t,M}^n f - \pi_{t,M}^n f\right)^4 |x_t\right]\right] = E\left[E\left(\frac{1}{n} w_{t,k}^i f(x_{t,k}^i) - \frac{1}{n}\sum_i n_{t,k}^i f(x_{t,k}^i)\right)^4\right] \le \frac{c_{t,M_1}^* ||f||^4}{n^2} \quad (16)$$

where $\bar{\pi}_{t,M}^n$ and $\pi_{t,M}^n$ represent the posterior distribution before and after resampling as shown in (3). c_{t,M_1}^* is a constant.

Define
$\bar{g}^n_{t,k} = \frac{1}{n}\sum_j g_{t,k}(x^j_{t,k-1}) = E^n_{\pi_{t,k-1}}(g_{t,k})$
$\bar{g}_{t,k} = \int g_{t,k}(x_t)\pi_{t,k-1}(x_t)dx_t = E_{\pi_{t,k-1}}(g_{t,k})$
We have

$$\pi_{t,k}f(x_t) = \frac{g_{t,k}(x_t)f(x_t)\pi_{t,k-1}}{\bar{g}_{t,k}}$$

$$\pi^n_{t,k}f(x_t) = \frac{g_{t,k}(x_t)f(x_t)\pi^n_{t,k-1}}{\bar{g}^n_{t,k}}$$

Here $g(\cdot)$ is the density of system noise as defined in the notation section. Then we have:

$$\begin{aligned} &E\left[E\left[\left(\pi_{t+1}f - \bar{\pi}^n_{t,M}f\right)^4 |\mathbf{x}_t\right]\right] \leq \\ &E\left[E\left[\left(\frac{g_{t,M}f\pi_{t,M-1}}{\bar{g}_{t,M}} - \frac{g_{t,M}f\pi^n_{t,M-1}}{\bar{g}_{t,M}}\right)^4\right]\right] + \\ &E\left[E\left(\frac{g_{t,M}f\pi^n_{t,M-1}}{\bar{g}_{t,M}} - \frac{g_{t,M}f\pi^n_{t,M-1}}{\bar{g}^n_{t,M}}\right)^4\right] \leq \\ &c_{t,M_2}E\left[\left(\pi_{t,M-1}f - \pi^n_{t,M-1}f\right)^4\right] + \\ &c_{t,M_3}E\left[\left(\bar{g}_{t,M} - \bar{g}^n_{t,M}\right)^4\right] \\ &= c^*_{t,M_2}E\left[\left(\pi_{t,M-1}f - \pi^n_{t,M-1}f\right)^4\right] \end{aligned} \tag{17}$$

where the last inequality for (17) is from $g < 1$, c^*_{t,M_2}, c_{t,M_2}, and c_{t,M_3} are some constants.

Substituting (16) and (17) into (15), we have:

$$\begin{aligned} &E\left[(\pi_t f - \pi^n_t f)^4\right] \leq \\ &\frac{c^*_{t,M_1}||f||^4}{n^2} + c^*_{t,M_2}E\left[\left(\pi_{t,M-1}f - \pi^n_{t,M-1}f\right)^4\right] \end{aligned} \tag{18}$$

By mathematical induction, we can show that

$$\begin{aligned} &E\left[(\pi_t f - \pi^n_t f)^4\right] \leq \\ &\frac{c^*_{t,M_1}||f||^4}{n^2} + c^*_{t,M_2}E\left[\left(\pi_{t,M-1}f - \pi^n_{t,M-1}f\right)^4\right] \leq \\ &\frac{c^*_{t,M_1}||f||^4}{n^2} + c^*_{t,M_2}\frac{c^*_{t,(M-1)_1}||f||^4}{n^2} \\ &+c^*_{t,M_2}c^*_{t,(M-1)_2}E\left[\left(\pi_{t,M-2}f - \pi^n_{t,M-2}f\right)^4\right] \leq ... \\ &\leq \frac{c^*_t tM||f||^4}{n^2} + c^*_t E\left[(\pi_0 f - \pi^n_0 f)^4\right] \end{aligned} \tag{19}$$

where $c^*_t = max(\prod_{i=1}^{t_0} c^*_{i,(l-1)_1}\prod_{k=l}^{M} c^*_{i,k_2})$ for any $1 \leq t_0 \leq t, 1 \leq l \leq M$.

Since $\lim_{n\to\infty}\pi^n_0 \to \pi_0$ a.s. by our initiation step, we have

$$\begin{aligned} &\sum_{n=1}^{\infty} E\left[(\pi_t f - \pi^n_t f)^4\right] \\ &\leq \sum_{n=1}^{\infty}\frac{c^*_t tM||f||^4}{n^2} + c^*_t\sum_{n=1}^{\infty}E\left[(\pi_0 f - \pi^n_0 f)^4\right] \leq \infty \end{aligned} \tag{20}$$

By Borel Cantelli lemma, $\lim_{n\to\infty}\pi^n_t \to \pi_t$ almost surely if $O(\frac{tM}{n}) < O(1)$, which is true when t is finite.

Similarly, for DPF-II, we can show that

$$E\left[(\pi_t f - \pi^n_t f)^4\right] \leq \frac{c^*_t t||f||^4}{n^2} + c^*_t E\left[(\pi_0 f - \pi^n_0 f)^4\right] \tag{21}$$

Hence, for DPF-II algorithm, we also have

$$\lim_{n\to\infty}\pi^n_t \to \pi_t \text{ a.s.}$$