Communication and Network, 2010, 2, 50-53
doi:10.4236/cn.2010.21007 Published Online February 2010 (http://www.scirp.org/journal/cn)

ADPF Algorithm for Target Tracking in WSN

Chunhe Song, Hai Zhao, Wei Jing, Dan Liu
Institute of Information and Technology, Northeastern University, Shenyang, China
E-mail: *songchunhe@tsinghua.org.cn, {zhhai, jingw, liud}@mail.neuera.com*
Received October 27, 2009; *accepted December* 29, 2009

Abstract: Particle filtering (PF) has been widely used in solving nonlinear/non Gaussian filtering problems. Inferring to the target tracking in a wireless sensor network (WSN), distributed PF (DPF) was used due to the limitation of nodes' computing capacity. In this paper, a novel filtering method—asynchronous DPF (ADPF) for target tracking in WSN is proposed. There are two keys in the proposed algorithm. Firstly, instead of transferring value and weight of particles, Gaussian mixture model (GMM) is used to approximate the posteriori distribution, and only GMM parameters need to be transferred which can reduce the bandwidth and power consumption. Secondly, in order to use sampling information effectively, when target moving to the next cluster head region, the GMM parameters are transfer to the next cluster head, and combine with the new local GMM parameters to compose the new GMM parameters incrementally. The ADPF can also deal with the situation of different number of nodes in different cluster when using the dynamic cluster structure. The proposed ADPF is compared to some other DPF for WSN target tracking, and the experimental results show that not only the precision is improved, but also the bandwidth and power is reduced.

Keywords: WSN, target tracking, asynchronous distributed particle filtering

1. Introduction

One of the major goals of WSN is to detect and track changes. The problem concerned is performing on-line state estimation for multi-dimensional signals that can be modeled using markovian state-space models that are nonlinear and non-Gaussian, Particle filter is one of the widely used tracking algorithms in non-linear/ Gaussian dynamic systems. When using such algorithm in sensor networks the energy cost related to computation in each sensor node and communication between sensor nodes is significant. Currently there are several distributed particle filters [1–3], in which the distributed nature is achieved by either transmitting local statistics of particles to a centralized unit or using the parameters passing method. Transmitting local statistics of particles to a centralized unit is not an efficient approach. Failure of the centralized unit is vital to the entire network. In the parameters passing method, the algorithms construct a path through the networks, which passes through all nodes. Global statistics of particles are accumulated by adding local statistics in each node through a forward pass. Then there needs a backward pass, which runs the important sampling and selection steps in each sensor node by using the accumulated global statistics.

In this paper, a novel filtering method – asynchronous DPF (ADPF) for target tracking in WSN is proposed. There are two keys in the proposed algorithm. Firstly, instead of transferring value and weight of particles, Gaussian mixture model (GMM) is used to approximate the posteriori distribution, and only GMM parameters need to be transferred which can reduce the bandwidth and power consumption. Secondly, in order to use sampling information effectively, when target moving to the next cluster head region, the GMM parameters are transfer to the next cluster head, and combine with the new local GMM parameters to compose the new GMM parameters incrementally. Because of computing asynchronously, this process can be regarded as ADPF.

The remaining of the paper is organized as follows: a brief description of PF and DPF in WSN are presented in Section 2. The details of the new PF this paper proposed – ADPF is presents in Section 3. In Section 4 the proposed algorithm is compared to other DPFs and finally, we give some concluding remarks in Section 5.

2. Tracking in WSN

Issues considered in this paper is tracking a moving target based on position measurements from multiple distributed sensors.

2.1 Target Motion Model

The target motion model used in this paper is the same as [4]:

$$x_k = f(x_{k-1}) + Gv_k, \quad k = 1, 2, \ldots \qquad (1)$$

where the target state vector $x_k = [x, \dot{x}, y, \dot{y}, w]^T$, consist of the position, velocity and the turn rate; $v_k \sim N(0, Q_k)$

is white process nosie: and

$$f(x)=\begin{bmatrix} 1 & \frac{\sin(\varepsilon_k T)}{\varepsilon_k} & 0 & -\frac{(1-\cos(\varepsilon_k T))}{\varepsilon_k} & 0 \\ 0 & \cos(\varepsilon_k T) & 0 & \sin(\varepsilon_k T) & 0 \\ 0 & \frac{(1-\cos(\varepsilon_k T))}{\varepsilon_k} & 1 & \frac{\sin(\varepsilon_k T)}{\varepsilon_k} & 0 \\ 0 & \sin(\varepsilon_k T) & 0 & \cos(\varepsilon_k T) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ w \end{bmatrix} \tag{2}$$

$$Gk=\begin{bmatrix} T^2/2 & 0 & 0 \\ T & 0 & 0 \\ 0 & T^2/2 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix} \tag{3}$$

2.2 Target Motion Model

In this paper, measurements of range and bearing are given by [5]:

$$z_k = h(x_k) + w_k^i, \quad i = 1,2,...,N \tag{4}$$

with

$$h(x_k)=\begin{bmatrix} ri \\ bi \end{bmatrix}=\begin{bmatrix} \sqrt{x_k^2 + y_k^2} \\ \tan^{-1}(y_k / x_k) \end{bmatrix} \tag{5}$$

And white measurement noise $w_k^i \sim N(0, R_k^i)$.

3. Basic PF and DPF in WSN

3.1 Basic Particle Filter

In the bayes filtering framework, the posterior distribution is updated recursively over the current state x_t given all observations $Z_t = \{z_i\}_{i=1}^t$ up to and including time *t* as follows[6]:

$$p(x_t \mid Y_{t-1}) = \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid Y_{t-1}) dx_k \tag{6}$$

$$p(x_t \mid Y_t) = \frac{p(y_t \mid x_t) p(x_t \mid Y_{t-1})}{p(y_t \mid Y_{t-1})} \tag{7}$$

$$p(y_t \mid Y_{t-1}) = \int p(y_t \mid x_t) p(x_t \mid Y_{t-1}) dx_k \tag{8}$$

Using Monte Carlo sampling points, particle filter executes the filtering process by generating weighted sampling points of state variances recursively. Generic particle filter algorithm can be found in [4].

3.2 Distributed Particle Filter in WSN

Particle filter has been widely used in target tracking. In WSN, the information transferred between nodes and the computing process in nodes are limited due to the restriction of power, computing capacity and bandwidth, so some changes must be conducted in particle filter in order to use it in WSN target tracking. The main idea of distributed particle filter is to deal with the computation at different sensor rather than at a central unit.

One typical DPF is the forward-backward type of distributed particle filter [6], which may be the first DPF for WSN. Supposing K nodes in WSN, and N particles for each node, in this algorithm, firstly, it is assumed that measurements at sensor are independent with each others, and in the particle filtering process, the likelihood $p(y_t \mid x_t)$ can be approximated by a parametric model $p(y_t \mid x_t) = L(x_t; \theta_t^k |_{k=1}^K)$. Secondly, only a single communication chain exists from node 1 to node k, with any node i in the interior of the chain communicating only with nodes i-1 and i+1. In the initialization step, each node samples N particles from $p(x_0)$. At time t, Node i sample from its importance distribution to generate N particles. Node i calculates the value of its likelihood for each one of these particles for the current observation and then trains the model $\{(\tilde{x}_t^{(j)}, L(x_t; \{\theta_t^k\}_{k=1}^i),$ $p(y_t^k \mid \tilde{x}_t^{(j)}))\}_{j=1}^N$. The parameters $\{\theta_t^k\}_{k=1}^i$ are then appropriately quantized and transmitted to node i+1 in the chain. In the next phase of this algorithm, the estimated parameters $\{\theta_t^k\}_{k=1}^K$ are propagated back along the communication chain. And each node uses the parameters to calculate estimates of likelihood for its samples $\{\tilde{x}_t^{(j)}\}_{j=1}^N$, which will be used to calculate the importance weights of particles. In the mentioned process above, it is assumed that the sensors act synchronously and record their measurements at the same time. So it can be regarded as synchronously particle filter.

There is complicated training process in this algorithm, and all nodes compute synchronously during target tracking. One simple idea is to transfers value and weight of particles directly between nodes and represents the posterior distribution. But there are N particles in each node and the transferred bits will be very large, so the posterior distribution of particle filter is assumed to be a GMM with C mixture probabilities. [2] is such type of DPF. In this algorithm, firstly, the WSN is divided into a series of group misrelated, and a single particle filter runs in each group. Through the head of current group, parameters of filter are transferred to the next head and update the posterior distribution. On the last group of sensors target tracking was estimated. As need transfer number of value and weight of particle, in this algorithm, low dimension GMM is used to approximate the likelihood distribution of DPF. To implement a distributed particle filter (DPF), particles and weights are distributed over entire network. Each sensor should maintain N particles $x_{m,k}^{(n)}$ and weights $w_{m,k}^{(n)}$. The posterior distribution of particle filter is assumed to be a GMM with C mixture probabilities. For each unobserved state y_c, observation $z_{m,k}$ follows a Gaussian distribution with mean μ_c and variance Σ_c :

$$p(z_{m,k} \mid \mu_c, \Sigma_c) = \frac{1}{\sqrt{2\pi\|\Sigma_c\|}} e^{-\frac{1}{2}(z_{m,k}-\mu_c)^T \Sigma_c^{-1} (z_{m,k}-\mu_c)} \tag{9}$$

The Gaussian mixture distribution for observation $z_{m,k}$ is:

$$p(z_{m,k} \mid \theta) = \sum \alpha_{m,c} p(z_{m,k} \mid \mu_c, \Sigma_c) \tag{10}$$

where θ is the set of the distribution parameters to be estimated, $\theta = \{\alpha_{m,c}, \mu_c, \Sigma_c; c = 1,...,C, m = 1,...,M\}$.

Assume all observed data from all nodes are sent to a centralized unit where a standard EM algorithm is used to estimate the parameter set θ. The log-likelihood for the observed data satisfies:

$$L(\theta \mid z) = \log \prod_{m=1}^{M} \prod_{j=1}^{k} p(z_{m,j} \mid \theta) = \sum_{m=1}^{M} \sum_{j=1}^{k} p(z_{m,j} \mid \theta)$$
$$= \sum_{m=1}^{M} \sum_{j=1}^{k} \alpha_{m,c} p(z_{m,k} \mid \mu_c, \Sigma_c) \tag{11}$$

4. Asynchronous Distributed Particle Filter

4.1 Dynamic Cluster Structure

In some former DPF, the cluster structure is fixed. In the ADPF, dynamic cluster structure is used. Supposing all nodes have the same detecting capacity, choose one cluster head, forming all nodes in the range of sing-hop of cluster head into a cluster, and this cluster head is used to deal with the sampling data and get local estimation. Supposing C is the max distance of sing-hop communication, R is the max range of detecting, and D is the distance of cluster head and other node. When target entering into the detecting region of WSN, and the number of node already detecting the target is over a predefined threshold, choose the nearest node as the cluster head. Forming the cluster and recall all nodes in this cluster. Following the movement of target, some nodes in the cluster will be out of the detecting region. If the number of nodes out of the detecting region is over a predefined threshold or *D*+*C*>*R*, choose a new cluster head and construct a new cluster. Otherwise predict the new location of target using filtering algorithms.

4.2 Gaussian Mixture Model Using EM

Using EM algorithm, the parameters of Equation 7) can be calculated. Given observation z and current parameter set θ^t where t is the time step between two consecutive sensor observations at k and k+1, the conditional expectation of joint distribution $p(z, y \mid \theta)$ is defined as:

$$Q(\theta, \theta^t) = \sum_{c=1}^{C} \sum_{m=1}^{M} \sum_{j=1}^{k} \log[p(z_{m,k}, y_c \mid \theta))] p(y_c \mid z_{m,j}, \theta^t)$$
$$\sum_{c=1}^{C} \sum_{m=1}^{M} \sum_{j=1}^{k} \log[\alpha_{m,k} p(z_{m,k} \mid \mu_c, \Sigma_c)] p(y_c \mid z_{m,j}, \theta^t)$$

Detail information about the Gaussian mixture model using EM algorithm can be found in [6].

4.3 Asynchronous Updating GMM Parameters

When running DPF in WSN asynchronously using GMM approximate posterior distribution, after getting into the new cluster region, former sampling information will be lost. Here propose a new GMM incrementally updating algorithm using PCA concept.

Supposing m0 nodes in the first cluster, each node send its parameters $\theta = [\alpha_{m,c}, \mu_c, \Sigma_c]^T$ to the first cluster head, and compose the parameters matrix $P_0 = [\theta_1, ...\theta_{n0}]$. It will be used to approximate the posterior distribution and transfer to the next cluster head.

Supposing the former cluster is the (i-1)th cluster with n_{i-1} sensors, parameters matrix is P_{i-1}, the current cluster is the ith cluster with n_i sensors, parameters matrix is P_i . $\overline{P}_{i-1}$ and $\overline{P}_i$ are the raw average vectors of P_{i-1} and P_i .

decomposing P_i with SVD:

$$P_i = U \Sigma V^T \tag{12}$$

and denote $\overline{P}_i$ as the row average vector of P_i .

Using the P_i and P_{i-1} to compose a new matrix $P_i^* = [P_i, P_{i-1}]$, decomposing it with SVD:

$$P_i^* = [P_i, P_{i-1}] = \tilde{U} \tilde{\Sigma} \tilde{V}^T \tag{13}$$

where $\tilde{U} = [U, U^*]$, $\tilde{\Sigma} = \begin{bmatrix} \Sigma & U^T P_{i-1} \\ 0 & U^{*T} P_{i-1} \end{bmatrix}$ and $\tilde{V}^T = \begin{bmatrix} V^T & 0 \\ 0 & I \end{bmatrix}$;

$$P_{ix} = [P_i - \overline{P}_i \mid sqrt(\frac{n_i * n_{i-1}}{n_i + n_{i-1}}) * (\overline{P}_i - \overline{P}_{i-1})] \tag{14}$$

$P_{iy} = P_{ix} * (I - U * U^T)$, calculate the QR decomposing of $P_{iy} : P_{iz} = QR(P_{iy})$. Using V^T, U, P_{ix} and P_{iz} to compose the matrix $\tilde{P}_i$:

$$\tilde{P}_i = \begin{bmatrix} ff * V & UT * P_{ix} \\ 0 & P_{iy}^T * P_{ix}(I - U * U^T) \end{bmatrix} \tag{15}$$

where ff is the forgotten factor with value in the range of [0, 1]. It indicates the weights of last parameters in the current computing time. In this experiment, the ff is set to 0.8.

Calculating SVD of $\tilde{P}_i$:

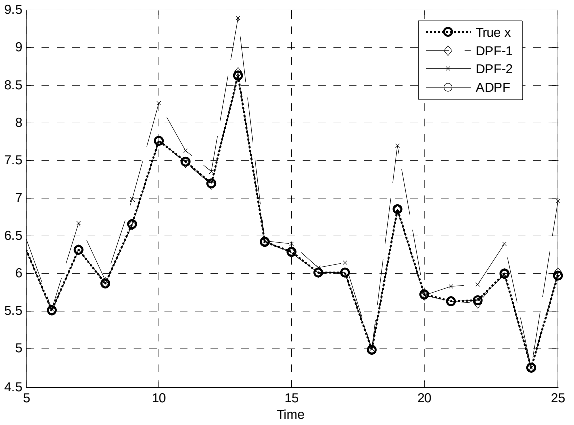

Figure 1. True states and observations

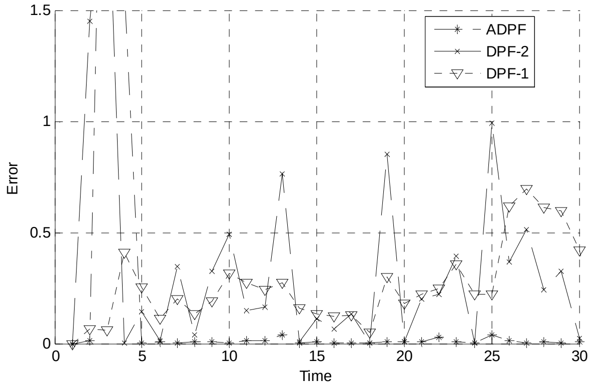

Figure 2. Results of mean errors

$$\tilde{P}_i = \tilde{U}\tilde{\Sigma}\tilde{V}^T \tag{16}$$

5. The Simulation Experiments

In order to test the proposed algorithm, the ADPF is compared to the DPF algorithms in [1] and [2]. Experimental results are shown in Figure 1–Figure 2 and Table 1. All results are the means of 100 runs.

As shown in the experimental results, it is clear that, the proposed ADPF has better performance than other two typical DPF algorithms. It can be explained as the proposed algorithms can use the sampling information as incremental updating GMM parameters more effectively.

6. Conclusions

Synchronous DPF and GMM parameters transferred DPF have their own disadvantages which limit their using range. In this paper, a novel filtering method – asynchronous DPF (ADPF) for target tracking in WSN is proposed. With incremental updating GMM parameters, ADPF can use the sampling information effectively. And ADPF can also deal with the situation of different number of nodes in different cluster when using the dynamic cluster structure. Simulation result shows that ADPF has better performance than other two typical DPF algorithms.

REFERENCES

[1] D. Guo and X. Wang, "Quasi-monte carlo filtering in nonlinear dynamic systems," IEEE Trans. Signal Process, Vol. 54, No. 6, pp. 2087–2098, 2006.

[2] M. S. Arulampalam, S. Maskell, N. Gordon, *et al*. "A tutorial on particle filters for online nonlinear/non- gaussian bayesian tracking [J]," IEEE Trans on Signal Proceeding, Vol. 20(2), pp. 174–188, 2002.

[3] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners [J]," IEEE Trans on Signal Proceeding, Vol. 50(3), pp. 736–746, 2002.

[4] B. D. Anderson and J. B. Moore, "Optimal filtering," Prentice-Hall, New Jersey, 1979.

[5] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tarcking part I: dynamci models," in IEEE Trans. Aerospace and Electronic System, Vol. 39, 2003.

[6] X. R. Li and V. P. Jilkov, "A survey of maneuvering target tracking-part III: measurement models," in SPIE Conf. on Signal and Data Proceeding of Small Target, 2001.

[7] M. Coates, "Distributed particle filters for sensor networks," in Proceeding of 3rd Intl sysmosium on Information Proceeding in sensor networks, Berkely, CA, USA.

[8] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," Proceeding of AeroSense: The 11th International Sysmpsium on Aerospace/ Defence Sensing, Simulation and Controls, Orlando, Florida, 1997. Vol. Muti Sensor Fusion, Tracking and Resource Mangement II pp.182–193.

[9] Y. Shi and R. C. Eberhart "A modified particle swarm optimizer," In Proceedings of the IEEE International Conference on Evolutionary Computation. Piscataway, NJ: IEEE Press, pp. 69–73, 1998.

[10] J. Riget, "A diversity-guided particle swarm optimizer," the ARPSO. EVALife Technical Report 2002–02, Dept. of Computer Science, University of Arhus, 2002.

[11] D. Guo, X. Wang, and R. Chen, "New sequential monte carlo methods for nonlinear dynamic systems," Statistics and Computing, Vol. 15, No. 2, pp. 135–147, 2005.

[12] Y. Bar-Shalom and X. R. Li, "Kirubarajan T. Estimation with applications to tracking and navigation: theory, algorithm and software [M]," New York: Wiley, 2001.