

Target Tracking Based Adaptive Particle Filter in Binary Wireless Sensor Networks

Su Ling-Dong, Zhai Ming-Yue

school of electrical and electronic engineering
North China Electric Power University
Beijing, China
e-mail: mingyue.zhai@gmail.com

Zhu Zhi-Yu

Institute of electronic and information engineering
Jiangsu University of Science and Technology
Beijing, China

Abstract—Since real-time and communication amount is crucial for the wireless sensor network target tracking, the performance of target tracking in the wireless sensor network is critically depended on real-time and communication amount reduction. This paper presents a target tracking method based on distributed adaptive particle filtering in binary wireless sensor network. Based on dynamic clustering, the adaptive particle filter receives the observations from children nodes and formulates the local estimate with the cluster head as the processing center. Simulation results show that the method can effectively improve the real-time tracking and reduce communication amount.

Keywords—adaptive particle filter; binary wireless sensor network; target tracking

I. INTRODUCTION

Recently, advances in wireless communication and electronics have allowed development of the wireless sensor network. They have been widely used both in commercial and military[1]. Target tracking is one of the most important applications in wireless sensor network. Considering particle filtering is very suitable for the situation of non-Gaussian and non-linear, many people use particle filtering for target tracking in wireless sensor network[2]-[4]. However, the wireless sensor network has the characteristics of the limited energy, the limited bandwidth and the limited energy of calculation. All the characteristics bring problems in the application of traditional particle filtering.

Traditional particle filtering cannot be directly applied in wireless sensor network, because of its large of the calculation. In order to save energy of the WSN, some people propose and implement the tracking by the distributed particle filtering(DPF)[5]-[8].Based on dynamic clustering, the particle filter receives the observations from children nodes and formulates the local estimate with the cluster head as the processing center. But in this algorithm, large amounts of data need to be transmitted between cluster head nodes. Inevitably, it will consume a large amount of energy. Currently, target tracking algorithms based on binary sensor network have been proposed[9]-[11]. In binary sensor network, the sensor only provides one bit of information about target: whether the target is present within the sensing range or not. The target tracking based particle filtering in

binary sensor network has been mentioned in [11]. But it is a centralized particle filtering algorithm(CPF), not suitable for the characteristics of WSN.

In this paper, we presents a target tracking method based on distributed adaptive particle filtering(DAPF) in binary wireless sensor network. The binary sensor network has more advantages than traditional sensor network. Because the sensor only transmit one bit of information, it will reduce communication amount. Furthermore, because of the particularity of the adaptive algorithm, the cluster nodes are not need to exchange large amount of data. Simulation results show that the method can effectively improve the real-time tracking and reduce communication amount.

The paper is organized as follows. First, in section II we provide a brief overview of the binary wireless sensor network. In section III we introduce the distributed adaptive particle filtering. In section IV we present the distributed adaptive particle filtering algorithm for tracking a single target in binary wireless sensor network. In the follow section, we give the simulation and demonstrate the performance of the proposed method in section IV.

II. BINARY SENSOR NETWORK DESCRIPTION

In the binary sensor network, sensors are deployed randomly. When the target is in the detection range of sensor network, the sensors measure a signal of the target and if the signal is above the threshold, the sensors transmit one signal to the fusion center; otherwise, they keep silent. In this paper, we assume the fusion center knows the location of the sensor nodes and also the nodes are fixed.

In fact, the binary wireless sensor networks have already been used in target tracking. In [9]-[10], they locate the target by geometric knowledge based on network structure, the sensor detection radius and combined with the time spent by the target through the detection range of the sensor. However, in this paper we use the model as in [11], that is, the received signal strength of the n th sensor is $y_{n,t}$, processes locally, and sending binary information to the fusion center according to the following steps:

1) The node compare with the received signal strength $y_{n,t}$ and the threshold γ , if the $y_{n,t}$ is less than γ , it keeps silent and doses nothing;

2) If the $y_{n,t}$ is above the γ , it sends the one bit of information to the fusion center.

Hence, only the node that received signal strength is above threshold send the information to the fusion center.

The received signal from the n th sensor in the fusion center is :

$$z_{n,t} = \beta_n s_{n,t} + \varepsilon_{n,t} \quad (1)$$

Where

$$s_{n,t} = \begin{cases} 1 & \text{if } y_{n,t} > \gamma \\ 0 & \text{if } y_{n,t} \leq \gamma \end{cases} \quad (2)$$

$\varepsilon_{n,t}$ is the observation noise, β_n is associated with the n th sensor, and $\varepsilon_{n,t} = N(0, \sigma_\varepsilon^2)$. In this paper, we set $\beta_n = 20$ and $\sigma_\varepsilon^2 = 0.01$.

From above we can see, we also need to know the received signal strength of the n th sensor $y_{n,t}$. In this paper, we use the model as follow:

$$y_{n,t} = g_n(x_t) + v_{n,t} = \frac{\Psi d_0^\alpha}{\|r_n - I_t\|^\alpha} + v_{n,t} \quad n=1,2,\dots,N \quad (3)$$

Where

$g_n(x_t)$ is the function of the received signal strength of the n th sensor, $v_{n,t}$ is the noise which independent from $\varepsilon_{n,t}$, r_n is the location of the n th sensor, I_t is the location of the target at time t , $\|r_n - I_t\|^\alpha$ is the Euclidean distance between I_t and r_n , Ψ is emitted power of the target at the distance d_0 , α is associated with the transmission medium, and $v_{n,t} = N(\mu_v, \sigma_v^2)$.

III. INTRODUCTION OF THE DISTRIBUTED ADAPTIVE PARTICLE FILTERING

A. brief introduction of particle filtering

Particle filtering is an emerging and powerful methodology for sequential signal processing with a wide range of applications in science and engineering. Particle filtering is a sequential Monte Carlo methodology. The basic idea is the recursive computation of relevant probability distributions using the concept of important sampling approximation of probability distributions with discrete random measures.

Consider the following DSS model[12]:

$$x_t = f(x_{t-1}, v_{t-1}) \quad (4)$$

$$z_t = h(x_t, n_t) \quad (5)$$

Where $z_t \in R^{n_y}$ is the observations of the model, $x_t \in R^{n_x}$ is the state vector of the system. $v_t \in R^{n_v}$ and $n_t \in R^{n_n}$ represent the state and observed noises, respectively, $f(\bullet)$ and $h(\bullet)$ are the state and observation functions, respectively. The objective is to estimate $x_{1:t}$ sequentially based on the observation $y_{1:t}$. All the information about the

state $x_{1:t}$ is contained in the posterior density $p(x_{1:t} | y_{1:t})$. Thus, our interest is the posterior density $p(x_{1:t} | y_{1:t})$ and the corresponding expectations $E(x_{1:t} | y_{1:t})$ which is the MMSE of the state $x_{1:t}$. The basic concept of Monte Carlo filtering is to approximately represent the posterior density with properly weighted samples drawn from a proposal distribution, $q(x_{1:t} | y_{1:t})$, also called importance function. If we draw N samples $X_t = \{x_{1,t}^1 \dots x_{1,t}^N\}$ from the importance function, the posterior density can be approximated as

$$p(x_{1:t} | y_{1:t}) = \sum_{i=1}^N \tilde{\omega}_t^i \delta(x_{1:t} - x_{1,t}^i) \quad (6)$$

Where

$$\omega_t^i = \frac{p(x_{1:t}^i | y_{1:t})}{q(x_{1:t}^i | y_{1:t})} \quad (7)$$

The $\tilde{\omega}_t^i$ are the normalized weight, and $\sum_{i=1}^N \tilde{\omega}_t^i = 1$. Thus, the expectations can be estimated by:

$$E(x_{1:t} | y_{1:t}) = \sum_{i=1}^N x_{1,t}^i \tilde{\omega}_t^i \quad (8)$$

The procedure has the following four steps:

1 generating particles from

$$x_{1,t}^i \sim q(x_{1,t}^i | x_{1,t-1}^i, y_{1,t-1}^i), \quad i=1,2,\dots,N$$

2 updating the weights for $i=1,2,\dots,N$ as

$$\omega_{t+1}^i = \omega_t^i \frac{p(y_t | x_t^i) p(x_{t+1}^i | x_t^i)}{q(x_{t+1}^i | x_t^i, y_{t+1}^i)}$$

3 normalizing the weights

$$\tilde{\omega}_{t+1}^i = \frac{\omega_{t+1}^i}{\sum_{j=1}^N \omega_{t+1}^j}$$

4 resample

In practice, after a short time run, only a few of the particles have a significant weight while weights of the other particles are approximately equal to zero. Hence, the procedure needs resample.

B. adaptive particle filtering

However, the traditional particle filtering is not suitable for the situation of the real time, because of its large amount of computation. For the system which has the exigent request for real time, the choice of sampling numbers is extremely important for particle filtering and it needs to make a compromise between filtering precision and compute complexity. This paper we propose an adaptive particle filtering that the number of used particles during filtering can be changed according to the real filtering situation[13].

Firstly, define a comprehensive evaluation index $IEI = \frac{1}{t \times RMSE}$. Where t is the simulation time and $RMSE$ is the error of the filtering. In order to balance number of

particles and IEI , define a integrate performance cost function (IPC):

$$C(\zeta_t, N_t) = m \int_{-\infty}^{+\infty} \zeta_t^2 f(\zeta_t) d\varepsilon_t + \bar{c}_t \times N(t) \quad (10)$$

Where the error of the estimation is $\zeta_t = \frac{1}{N(t)} \sum_{i=1}^{N(t)} x_t^i - \hat{\mu}_{t-1}$, m is a coefficient, \bar{c}_t is a particle's average time consuming, $f(\zeta_t)$ is probability density of the error, and $N(t)$ is the number we need. Hence, in order to make $C(\zeta_t, N_t)$ minimum, $N(t)$ can be expressed as in (11):

$$N(t) = \sqrt{\frac{mP_{t-1}}{\bar{c}_t}} \quad (11)$$

The adaptive particle filtering has the following tree steps:

1). initialize $t=0$

Draw $x_0^i \sim p(x_0)$ from prior density for $i = 1, 2, \dots, N_0$

2). $t = t + 1$

Calculate the number of particles $N(t)$ according to

$$N(t) = \sqrt{\frac{mP_{t-1}}{\bar{c}_t}}$$

Draw $N(t)$ particles $\{x_{0,t-1}^i, \omega_{t-1}^i\}_{i=1}^{N(t)}$ from $N(\hat{\mu}_{t-1}, P_{t-1})$

Update $\{x_{0,t}^i, \omega_t^i\}_{i=1}^{N(t)}$

State estimation $\hat{\mu}_t = \sum_{i=1}^{N(t)} x_t^i \omega_t^i$

Variance estimation

$$P_t = \sum_{i=1}^{N(t)} \omega_t^i (x_t^i - \hat{\mu}_t)(x_t^i - \hat{\mu}_t)^T$$

3). quit the filter if the filter is end, otherwise back to the step 2.

In order to ensure the accuracy of the filter, set the minimum number of the particles. When $N_{bottom} > N(t)$, then set the number of particles are N_{bottom} . In this paper, we set $N_{bottom} = 100$.

C. Distributed algorithm

In order to overcome these shortcomings of the centralized tracking, the tracking strategy should use distributed dynamic clustering.

1). when the target is moving in their detection range, wake up the sensors and choose the largest $y_{n,t}$ as the head node;

2). all the nodes which are within 10m away from head node group cluster with the head node;

3). when the target moves out of the detection range of a node in the cluster, reselect the head node to form a new cluster;

4). measurement data and status information of the original cluster head is transferred to the new cluster head for target tracking;

5). repeating the clustering process until the target move out the wireless sensor network.

D. Distributed adaptive particle filtering algorithm in binary wireless sensor network

As we can be seen from above, particles draw from the important density function. The number of particles is M , weight is the $\tilde{\omega}_t^i$, and $\tilde{\omega}_t^i \propto p(z_t | x_t^i) \tilde{\omega}_{t-1}^i$. Since the noise samples $\varepsilon_{n,t}$ are assumed independent, we have:

$$p(z_t | x_t^i) = \prod_{n=1}^N p(z_{n,t} | x_t^i) \quad (11)$$

Hence, $p(z_{n,t} | x_t^i)$ can be written as

$$\begin{aligned} p(z_{n,t} | x_t^i) &= p(z_{n,t} | s_{n,t} = 0, x_t^i) p(s_{n,t} = 0 | x_t^i) \\ &\quad + p(z_{n,t} | s_{n,t} = 1, x_t^i) p(s_{n,t} = 1 | x_t^i) \\ &= p(z_{n,t} | s_{n,t} = 0) p(s_{n,t} = 0 | x_t^i) \\ &\quad + p(z_{n,t} | s_{n,t} = 1) p(s_{n,t} = 1 | x_t^i) \end{aligned} \quad (12)$$

Where

$$p(z_{n,t} | s_{n,t}) = N(\beta_n s_{n,t}, \sigma_\varepsilon^2) \quad (13)$$

And

$$p(s_{n,t} = 1 | x_t^i) = Q\left(\frac{\gamma - g_n(x_t^i) - \mu_v}{\sigma_v}\right) \quad (14)$$

$$p(s_{n,t} = 0 | x_t^i) = 1 - Q\left(\frac{\gamma - g_n(x_t^i) - \mu_v}{\sigma_v}\right) \quad (15)$$

Where $Q(\cdot)$ denotes the complementary of the standard normal cumulative distribution function.

In summary, we can perform the distributed adaptive particle filtering in binary wireless sensor networks as following steps:

1). initialize $t = 0$

Format cluster, and draw $x_0^i \sim p(x_0)$ from prior density for $i = 1, 2, \dots, N_0$

2). $t = t + 1$

Format cluster

Calculate the number of particles $N(t)$ according to

$$N(t) = \sqrt{\frac{mP_{t-1}}{\bar{c}_t}}$$

Draw $N(t)$ particles $\{x_{0,t-1}^i, \omega_{t-1}^i\}_{i=1}^{N(t)}$ from $N(\hat{\mu}_{t-1}, P_{t-1})$

Update $\{x_{0,t}^i, \omega_t^i\}_{i=1}^{N(t)}$ according to Eq.(11)~(15)

State estimation $\hat{\mu}_t = \sum_{i=1}^{N(t)} \omega_t^i x_t^i$

Variance estimation

$$P_t = \sum_{i=1}^{N(t)} \omega_t^i (x_t^i - \hat{\mu}_t)(x_t^i - \hat{\mu}_t)^T$$

3). select the next fusion node according to the node selection algorithm. Transmit $(\hat{\mu}_t, P_t)$ to the next fusion node and estimate the next state.

IV. SIMULATIONS

We present some computer simulations that illustrate the performances of the proposed algorithm. We considered a scenario where the sensors are deployed randomly in a field with dimensions 100×120 . Assume all the sensors have the same property, and each node knows the location of the remaining nodes. The attenuation parameter was set to $\alpha = 2.5$ and the reference power parameter to $\Psi = 5000$ at $d_0 = 1m$. The threshold was set to 2.5. We set $\mu_v = 1$, $\sigma_v^2 = 0.01$, $\sigma_e^2 = 0.01$, and $\beta_n = 20$. In (11), we set $m = 0.85$. According to trials, we get $\bar{c}_t = 0.000289s$.

Motion model of the target was $x_t = G_x x_{t-1} + G_u u_t$.

Where $x_t = [x_{1,t} \ \dot{x}_{1,t} \ x_{2,t} \ \dot{x}_{2,t}]$ is the state vector, x represent the location of the target, \dot{x} represent the velocity of the target, T_s is the sampling interval, u_t is the noise related to the acceleration of the target.

Besides

$$G_u = \begin{bmatrix} 1 & \sin mT_s & 0 & \frac{1 - \cos mT_s}{m} \\ 0 & \cos mT_s & 0 & -\sin mT_s \\ 0 & \frac{\cos mT_s}{m} & 1 & \frac{\sin mT_s}{m} \\ 0 & \sin mT_s & 0 & \cos mT_s \end{bmatrix} \quad G_u = \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ 1 & 0 \\ 0 & \frac{T_s^2}{2} \\ 0 & 1 \end{bmatrix}$$

Where $m = \frac{a}{\sqrt{\dot{x}^2 + \dot{y}^2}}$, $a = 2$, \dot{x} and \dot{y} represent component of the velocity at the current moment, respectively.

In the implement of distributed adaptive particle filtering, we use $N = 200$ particles and 200 sensors. In Fig.1, we can see the sensors are displayed with small circles and the head nodes are displayed with small star. In the addition, the true trajectory and the track trajectory is displayed with the solid line and dotted line, respectively. It can be seen that the algorithm track the target's trajectory closely, so the algorithm we proposed is effective.

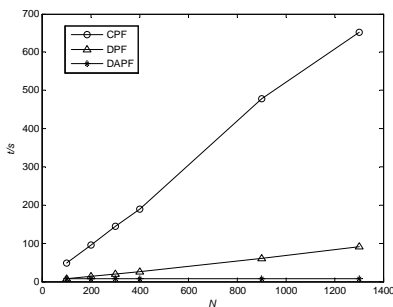


Figure 1. Target tracking with DPAF

In Fig.2, we can see the simulation time-consuming comparison of three algorithms (centralized particle filtering, distributed particle filtering and distributed adaptive particle filtering). It is clear that distributed adaptive particle filtering's time-consuming is significantly less than the centralized particle filtering's and distributed particle filtering's in the case of the same number of particles. At the same time, with the increase of the number of particles, distributed adaptive particle filtering has the same time-consuming. It is because the number of used particles during filtering can be changed according to the real filtering situation.

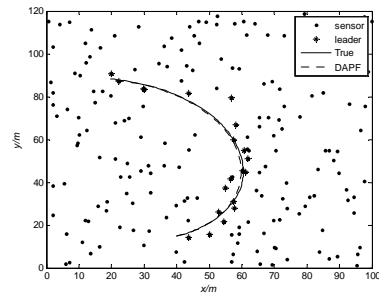


Figure 2. Comparison of time-consuming of three algorithms

Considering the specificity of wireless sensor network, communication amount is the important indicator to measure the effectiveness of the algorithm. We use the same method as in [8] to estimate communication amount but change some parameters.

$$N_e N_f + 3c N_c N_s N_f + N_n N_i$$

Where $N_f = 16$, $N_i = 8$, $N_c = 1$, $N_n = 1$, N_e represent the number of nodes within the cluster, N_s represent the dimension of the state variables. In this paper, we set $N_s = 4$.

TABLE I COMPARISON OF COMMUNICATION AMOUNT OF THREE ALGORITHMS(BITS)

Number	DPF	GMM-DPF	DAPF
100	141618	9266	3634
200	256789	8469	3349
400	512783	8463	3343
900	1095191	8087	3223

In the table 1, we can see the traffic comparison of DPF, DAPF, and DPF in Gaussian mixture model (GMM-DPF). We can see from the table that distributed adaptive particle filtering's traffic is the least in these algorithms. From above all, we can see our algorithm yields good performance and outperforms other algorithms.

V. CONCLUSION

Target tracking is the important application of wireless sensor network. But the characteristics of wireless sensor network limit the traditional tracking algorithm. In order to track target accurately in wireless sensor network, in this paper, we focus on the use of distributed adaptive particle filtering for target tracking in binary wireless sensor network. Extensive simulations of this algorithm performed under

different configurations. From the simulations, we can see our algorithm can track the target accurately, and reduce the communication amount and the energy consumption. Therefore, distributed adaptive particle filtering is an effective algorithm and is more fit for wireless sensor network.

ACKNOWLEDGEMENTS

The paper is supported by the Natural Science Foundation of China (NSFC) with Grant No. 60972004, and Nature Science Foundation of Beijing with Grant No. 4122073.

REFERENCES

- [1] H.Yang, B.Sikdar, "A protocol for tracking mobile targets using sensor networks", in proceedings of IEEE SNPA, pp.71-81, 2003
- [2] Hongliang Ren, Max Q.-H. Meng, "Power Adaptive Localization Algorithm for Wireless Sensor Networks Using Particle Filter", IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, vol.58, NO.5, pp.2498-2507, 2009
- [3] Onur Ozdemir, Ruixin Niu, Pramod K. Varshney, "tracking in wireless sensor networks using particle filtering: physical layer considerations", IEEE transactions on signal processing, vol.57, NO.5, pp.1987-1999, 2009
- [4] O. Ozdemir, R. Niu, P. K. Varshney, "Channel aware particle filtering for tracking in sensor networks", in Proc. Asilomar Conf. Signals, Systems, Computers, Pacific Grove, CA, Oct. pp.290-294, 2006.
- [5] Wei Gao ,Hai Zhao, Chunhe Song, "A new distributed particle filtering for WSN target tracking", in 2009 International Conference on Signal Processing Systems, pp.334-337, 2009
- [6] Mark Coates, "Distributed particle filters for sensor networks, Information Processing in Sensor Networks", IPSN2003, Springer ,pp:99-107, 2004
- [7] L. Zuo, K. Mehrotra, P. Varshney, "Bandwidth-efficient target tracking in distributed sensor networks using particle filters", in Proc. of 14th European Signal Processing Conference EURASIP2006, 2006.
- [8] Sheng X H , Hu Y H , "Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network", in Proc of 4th Int Symposium on Information Processing in Sensor Networks, pp.181-188, 2005
- [9] R. Niu, P. Varshney, "Target location estimation in wireless sensor networks using binary data", In Conference on Information Sciences and Systems, 2004.
- [10] Zijian Wang, Eyuphan Bulut, "Distributed Target Tracking with Imperfect Binary Sensor Networks", Proceedings of the Second Annual Conference of the International Technology Alliance, pp:306-307, 2008
- [11] Petar M. Djuric, Mahesh Vemula, "Target Tracking by Particle Filtering in Binary Sensor Networks", IEEE Transactions on signal processing, vol.56, NO. 6, pp:2229-2238, 2008
- [12] Tadesse Ghirmai, Jayesh H.Kotecha, "Blind equalization for time-varying channels and multiple samples processing using particle filtering", Digital Signal Processing, vol.14, pp.312-331, 2004
- [13] Zhang Miao, HU Jian-wang, "A new adaptive particle filtering algorithm", in command control and simulation, Vol.31, pp.39-44, 2009