Target Tracking Approximation Algorithms with Particle Filter Optimization and Fault-Tolerant Analysis in Wireless Sensor Networks

Xiang Gao  
School of Microelectronics, Xidian University, Xi’an, China  
Email: xgao@mail.xidian.edu.cn

Yintang Yang¹, Duan Zhou², Jianxian Zhang¹,² and Changchun Chai¹  
¹ School of Microelectronics, Xidian University, Xi’an, China  
² School of Computer Science and Technology, Xidian University, Xi’an, China  
Email: ytyang@xidian.edu.cn, zhouduan3788@126.com, {jxzhang, ccchai}@mail.xidian.edu.cn

Abstract—In order to process target tracking approximation with unknown motion state models beforehand in a two-dimensional field of binary proximity sensors, the algorithms based on cost functions of particle filters and near-linear curve simple optimization are proposed in this paper. Through moving target across detecting intersecting fields of sensor nodes sequentially, cost functions are introduced to solve target tracking approximation and velocity estimation which is not similar to traditional particle filters that rely on probabilistic assumptions about the motion states. Then a near-linear curve geometric approach is used to simplify and easily describe target trajectories that are below a certain error measure. Because there may be some sensor nodes invalid in practice, so a fault-tolerant detection is applied to avoid the nodes’ reporting fault and also improve accuracy of tracking at the same time. The validity of our algorithms is demonstrated through simulation results.

Index Terms—particle filters, cost function, target tracking, fault tolerant

I. INTRODUCTION

In Wireless Sensor Networks (WSN), binary proximity sensors are able to be tasked to report outputs 1 when a motion target is within the proximity, and 0 otherwise. This simple sensing model is of both practical interest and information processing for several reasons. Firstly, simplified binary outputs based on the quantization rule [1] from sensors are able to save the large number of data transmission to fusion centre. Secondly, WSN fulfill the sensors’ location deployment based on the minimal sensor sensing capability [2] so that sensors ensure communication connection each other during the target tacking. Finally, this simple model permits the derivation of performance of abstraction, so tracking algorithms are designed under the assigned error and are achieved through auxiliary near-linear curve simplification approach.

However, the classical target tracking algorithm is often formulized as a Kalman filtering problem about gaussian models or a particle filtering issue about non-gaussian and non-linear models. The Kalman filter and the Extended Kalman filter (EKF) are presented and derived to solve the given systems dynamic models under linear and Gaussian state systems as well as non-linear and non-Gaussian state functions respectively [3]. The Kalman filter could be able to converge the systems to a steady state and convey the last system measurement to the novel state estimate. However, EKF using the frame of linear filter to solve non-linear systems results in the state linearization. Hence, EKF is not an optimal approach to obtain the target trace in this problem. The particle filters based on recursive Bayesian filters could be applied to any state transition and measurement model, so they are greatly superior to the EKF and currently applied to solve non-linear systems [4]. At the same time, many different modified particle filters are proposed. A new Gaussian sum particle filter is used to perform the sensors selection and the tracking is propagated among the fixed sensor nodes [5], while the multi model auxiliary particle filter [6] is presented to track the target among the mobile sensor nodes. When there are multiple targets in systems dynamics, a multi-modal particle filter with fast tracking capability is introduced to track multi-target trace [7].

Above these filtering algorithms, the authors adopt differently filters to handle sampled information obtained from the assumptions of given or known individual system dynamic mathematical models and probabilities, which are related to the system states and the observations. Meanwhile, in order to sample the information, sensors are able to process different sensing properties, such as radar, acoustic, magnetic, and so on [8]. While an assumption is proposed that the target states’ models are unknown in advance and sensors have the simplest function of reporting the target within their sensing regions, the common filtering algorithms are not suitable in this situation.
In this paper, we assume the tracking conditions without any sensing information about orientation, speed or other attributes of the target, and also, there are no probabilistic assumptions and system models used. However, cost reference particle filters [9, 10], which make use of user-defined cost functions measure the quality of state signal estimates according to the given system state-space mathematical models, could be introduced to solve the problems of no probabilistic assumptions. And then, our emphasis is on utilizing geometric features of neighbor sensors’ common sensing areas to establish limits of candidate paths and target moving boundaries according to sensors’ sensing disks partition [11] and the idea of minimalist approach [12] respectively. A particle filtering algorithm is used to clean up influences of roughly detecting information and limits of the boundaries. So Cost-Function Particle Filters based on Piece-wise Linear Curve (CPF-PLC) is designed for getting preferable target tracking and velocity estimation. In the meantime, a fault-tolerant analysis presented based on [13, 14] is added to target tracking considering the practical application.

II. The Geometric Description of Sensing Disks

For simplicity, the authors assume that the sensing region of each sensor is disk, and sensors are deployed uniformly which could keep communication coverage of the whole networks. Sensing binary information is transmitted by dynamic hierarchy routing protocol [15]. All sensors are assumed to detect an object correctly in their effect sensing regions and incorrect detections are avoided by fault-tolerant algorithms proposed [16].

An example is proposed to express the model of target’s movement across these sensing disks and some geometric features are also defined as follows.

![Figure 1. The geometry of target trajectories across three sensors’ sensing disks](image)

The shaded areas $S_j$ are target’s location over time intervals $t_i$ with constant sensing information, and the sensing binary boundary signals of sensor a, b and c are defined by $d_{ja}$ in Fig. 1. The located boundary curves $B_j (j = 1, 2, \ldots)$ of the moving target in each shaded area are stamped by $d_{ja}$ in time $t_i$ order and target tracking algorithms in this paper will use these geometric features to restrain particle sampling limits and tendency of tracking estimation.

III. TRACKING WITH GEOMETRIC FILTERING

In this section, the algorithm of CPF-PLC is described in detail. When target tracking obtained by Cost-Function Particle Filtering (CPF), a posterior near-linear curve simplified algorithm is proposed to improve the target’s velocity estimation and also clean up special errors between true target trajectories and estimate paths.

A. Definition of Cost Function

In order to estimate target position $X_{0,t}$ which is two-dimensional vector including x and y coordinates and couldn’t be able to makes use of any Probability Distribution Functions(PDF), a cost function is defined to deduce a posteriori state PDF $P(X_{t_0}, Y_{t_0})$ according to sampling quality observations in $S_j$.

The idea of this cost function is derived from [9].The candidate paths (called particles) are created in terms of samples at each time instant $t$ by the previous candidate paths at time instant $t-1$. At any time $t$, there are $M$ particles within the current $S_j$ for the $l$th particle denoted by $X^l_{t}$. At the next time instant $t+1$, $N$ particles are chosen uniformly at random from the current $S_j$ for $l$th particle denoted by $X^l_{t+1}$, and so there are $MN$ increments between the position $X^l_{t}$ and $X^l_{t+1}$, and also $MN$ particles. The increment $Y_{t, i, i}$ is defined as:

$$Y_{t, i, l} = X_{t, i+1}^l - X_{t, i}^l$$  \hspace{1cm} (1)

We assume that target trajectories are low-frequency variation with respect to the target moving trend in each $S_j$. Therefore, when sampling time difference is shorter between $t$ and $t+1$, the value of $Y_{t, i}$ is able to be regarded as the approximate vector velocity at time $t+1$. Sampling weights are abstracted from the cost function that penalizes changes in the vector velocity. These weights will depend on the former observation $Y_t$ at time instant $t$ to estimate this potential punishment and void bad samples. Based on a prediction of the changes vector velocity from $Y_t$, a cost function is given by

$$R(X_t | Y_t) = \left| Y_{t, i} - Y_t \right|.$$  \hspace{1cm} (2)

B. Cost-Function particle filtering algorithm

This section describes CPF on the basis of above the definitions.

First step: selection of the most promise particles, namely resampling process.

The proposed resampling technique proceeds sequentially in a manner similar to the Sample Important Resampling (SIR) method [17] and initial costs are assigned zero. Given a set of $MN$ increments at time
instant $t$, where $M$ particles $X^m_{i,t}$ are from time instant $t-1$ and $N$ particles $X^n_{i,t}$ are chosen from $M$ particles uniformly at time instant $t$ respectively. According to (2), let the particles $\{X^m_{i,t}, Y^{(m)}_{i,t}\}_{m=1}^{MN}$ sorted based on their predicted costs $R^{(m)}_t$ in descending order and the first $M$ of $MN$ particles are replicated $N$ times which is to avoid sample impoverishment. The weights are defined as:

$$\omega^m_t \propto \lambda(R^{(m)}_t), \quad (3)$$

where $\lambda$ is a monotonically decreasing function which is defined as $\lambda(R^{(m)}_t) = \frac{1}{(R^{(m)}_t)^{\beta}}$. The new particle streams $\{X^m_{i,t}, R^{(m)}_t\}_{m=1}^{MN}$ at time instant $t$.

Second step: choose $M$ particles from $X^m_{i,t}$ with the best cost $R^{(m)}_t$. For $m = 1, \ldots, MN$, let

$$Y^{(m)}_{i,t} = \left\| X^m_{i,t} - X^n_{i,t} \right\|,$$  

where $X^m_{i,t} (m = 1, \ldots, N)$, and $m$ are the samples at time $t+1$ for the first step. So the costs at time $t+1$ are expressed as:

$$R^{(m)}(X^m_{i,t} \mid Y^{(m)}_{i,t}) = \left\| Y^{(m)}_{i,t} - Y^n_{i,t} \right\| \quad (5)$$

and $\omega^n_{i,t}$ is obtained by (3).

Third step: estimation of the position $X_{i,t}$. There are many ways to estimate $X_{i,t}$ at time instant $t+1$, such as:

$$\overline{X}_{i,t} = \sum_{m=1}^{MN} X^m_{i,t} \omega^m_{i,t}. \quad (6)$$

### C. Near-Linear Curve Simplification Algorithm

In this paper, there is no any probabilistic assumption or target motion state information used. The target tracking estimated by CPF may omit and exceed the boundary limits about each $S_i$, where the estimate trajectories stab across $B_j$. So a posterior near-linear curve simplification algorithm named Piece-wise Linear Curve (PLC) from [18] is proposed to solve the problems. Through PLC, the velocity estimate error $\varepsilon$ is able to be bounded in a reasonable range and then it is helpful to improve velocity re-estimation at the same time.

The description of PLC in detail is seen from [18]. Hence, the estimate trajectories of the target obtained from CPF are viewed as a polygonal curve $C$ which is composed of line segments of vertices $\overline{X}_{i,t}$. The aim is to minimize the maximum number of vertices $\overline{X}_{i,t}$ of curve $C$ to get a simplified curve $C'$ so that the error between $C$ and $C'$ is under an assigned error measure $\phi$. Then, average diameter $d$ of each $S_i$ is able to be computed by [11,19]. This value of $d$ means that the segment length of estimate trajectories in each $S_i$ is lower than $d$ at least. Therefore, the following conclusions are obtained:

Conclusion 1: when $\phi$ is equal to $d$, the velocity estimate error $\varepsilon$ could be optimized and be lower than an upper bound at least with $d$-simplification of curve $C$.

Proof. Conclusion is given that $\varepsilon \leq \frac{\sqrt{2}d}{X_{i,t}X_{i,t+1}}$, and the detailed proof process is seen from [19]. At the same time, it is known that the size of $C$ with $d$-simplification of curve $C$ is at most the size of $C'$ with $d/2$-simplification of curve $C$ proved in [18]. So the error of $\varepsilon$ will become larger with the increment of $\lambda$, which leads to increment of the length of $\overline{X}_{i,t}X_{i,t+1}$ when $C$ is a monotone curve. In section III-A, target trajectories have been assumed as the low-frequency variation. Consequently, the curve $C$ here is able to be viewed as the monotone curve approximatively. Then, the differences of true trajectories and estimate paths are at least, and therefore, when $C$ is determined as a $d$-simplification of $C$, $\varepsilon$ is lower than a certain upper bound at least.

Conclusion 2: the intervals of sampling time in CPF must be shorter than that of the target across the two adjacent $B_j$.

Proof. Because PLC is to clean up overlength segments during the estimation of CPF and conclusion 1 clarifies that $C$ is a $d$-simplification of $C$ at least, there is one particle at least in each $S_i$ so that sampling time is must shorter the differences between $t_i$ and $t_{i+1}$. The shorter sampling time, the more accurately target trajectories are estimated because of the more sampling particles.

The algorithm of PLC is described as follows:

<table>
<thead>
<tr>
<th>Algorithm : PLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $C = \phi$;</td>
</tr>
<tr>
<td>2. for all $\overline{X}_{i,t}(t = 1,2,\ldots)$ do $k = t+1$</td>
</tr>
<tr>
<td>3. if $\delta(\overline{X}_{i,t}, C) &gt; \phi$ then $k = t+1$</td>
</tr>
<tr>
<td>4. $C' \leftarrow C' \cup \overline{X}_{i,t}$;</td>
</tr>
<tr>
<td>5. else</td>
</tr>
<tr>
<td>6. $t = t+1$ and $k = t$;</td>
</tr>
<tr>
<td>7. end if</td>
</tr>
<tr>
<td>8. end for</td>
</tr>
</tbody>
</table>

The function of $\delta(\cdot)$ denotes the error of the segment $\overline{X}_{i,t}$ with respect to $C$ under the error measure $\phi$.

### D. Sensor Nodes Fault-tolerant Models

Considering some sensor nodes invalid about their reporting of the tracking information, a fault-tolerant detection algorithm is introduced to modify their fault reporting and improve the tracking accuracy according to [13]. The fault sensor nodes are corrected by Bayesian probabilities estimation of their neighbor nodes, which could report correct binary proximity when the target moves in or out of the nodes’ sensing disks. Define $E_i(a,K)$ as the output result of $K$ neighbor nodes which make the same decision “$a$” around senor node $i$. If sensor node $i$ could be corrected as the decision “$a$” by $K$ neighbor nodes, PDF is expressed:


\[ P(D_i = a | E_i(a, K)) = \frac{K}{N}, \]

where \( N \) is the number of the total neighbor nodes around sensor node \( i \).

According to Bayesian estimation, the probability that the sensor node can make a decision about whether to disregard its own sensor reading \( S_i \) in the face of the evidence \( E_i(a, K) \) from its neighbor nodes is:

\[ P_{\text{act}} = P(D_i = a | S_i = a, E_i(a, K)) = \frac{(1-p)K}{(1-p)K + p(N-K)}, \]

where \( p \) is the sensor node fault probability is uncorrected.

Let \( g_K \) be the probability that exactly \( K \) of node \( i \)'s \( N \) neighbor nodes are not faulty,

\[ g_K = \binom{N}{K} (1-p)^k p^{(N-k)}. \]

The average number of errors after the fault-tolerant analysis:

\[ \alpha = \sum_{k=0}^{N} P_{\text{act}} g_{N-k} p + (1 - \sum_{k=0}^{N} P_{\text{act}} g_{k})(1-p) \]

where \( n \) is total number of sensor nodes in WSN.

To make the value \( \alpha \) minimum, the threshold of probabilities \( \Theta \) is defined as:

\[ \Theta = \frac{(1-p)(K-1)}{2p + (K-1)}. \]

The detailed proof is seen from [13, 14].

IV. SIMULATION RESULTS

The simulation tests are carried out to evaluate the performance of our algorithm of CPF-PLC. The whole algorithm is written in matlab. Our general experimental setup simulated a 150 x 150 unit field, including 25 sensors in a regular 5 x 5 grid with the distance of 24 units. The sensing radius for each sensor is set to 45 units. The number of sampling particles is 100. A similar negative spin curve is created to simulate the target trajectories. We also introduce the algorithm of Occam track [19] to compare with our algorithm. Occam track is to get the longest segments as estimate trajectories which cross the most \( B_j \) and is a pure geometric algorithm. At the same time, the probability \( p \) is set to 0.2, and each node can only communicate with its immediate neighbor node in each cardinal direction. So each node has four neighbor nodes at most when this node is located in central area of WSN.

Fig. 2(a) shows that original sensor nodes aren’t modified by fault tolerant detection. The star points presented in Fig. 2(a) stand for the error nodes uncorrected from original sensor faults. However, the corrected sensor faults are shown in Fig. 2(b), where the star points with circle are not corrected nodes which are errors before detection and the dot points with circle are new errors introduced and the rest points denote correct sensor nodes. So through the detection, the sensor faults reduce by three sensor faults from five sensor faults from Fig. 2(a), which can improve performance of the tracking.

Real lines stand for the target trajectories and dotted lines present the estimate trajectories using different tracking algorithms in Fig. 3. The deployment of the sensor nodes with fault tolerant detection in Fig. 3(a) and (b) is the same as that in Fig. 2(b). The cross points with circle stand for failure nodes in Fig. 3(a) and (b). However, the sensor nodes are correct sensor nodes in Fig. 3(c) and (d). Because CPF-PLC track makes use of the particle filtering, the accuracy of estimation is higher than that in Occam track essentially. At the same time, CPF-PLC track also adopts a posterior clean up process which can reduce estimation errors in inflexion of the trajectories from the results. Average segment is about 3d through simulations. So the results from Fig. 3 shows that the performance of CPF-PLC is well on solving some special trajectories and could obtain a better approximation on target tracking. At the same, the tracking accuracy of Fig. 3(a) and (b) with fault tolerant are less than that of Fig. 3(c) and (d) without fault tolerant (namely without sensor faults) respectively. The reason is obviously that the error sensor nodes affect the performance of the tracking. Hence, if we want to improve the accuracy of the tracking, the measure to be taken is that more correct and effective sensor nodes are
deployed in around of the error nodes so that the applications about WSN can’t be affected by sensor faults.

Figure 3. Compare between two tracking algorithms with or without fault tolerant

Figure 4. Velocity estimation by CPF-PLC

In Fig. 4, the velocity is estimated by CPF-PLC track without fault tolerant. The results shows that the velocity is kept approximate value 1.1 units to wing fluctuation and the relative velocity is lower than the value 0.2 units, which means that the velocity estimate error $\varepsilon$ is about 20%. Then the results from figure 3 show that average segment is about $3d$. These data represent that our algorithm keeps a good performance according to collusion 1 and this algorithm presents the robust estimation.

V. CONCLUSION

In this paper, we present two sub-algorithms for target tracking so that they become an effective and accurate estimation about the trajectories and velocity. Hence, CPF-PLC is designed to achieve this aim and avoids that estimate trajectories across $B_j$ is out of pre-ordained order, while this situation might appear in occam track of [19]. Particle filters can present large number of robust samples, so it is able to estimate the better trajectories. Then a near-linear curve approach is used to simplify and easily describe target trajectories under a certain error so that the velocity estimate error $\varepsilon$ is given the upper bound at least. The simulation results prove our conclusions.

ACKNOWLEDGMENT

This paper is supported by the Major State Basic Research Program of China (B1420080204), National Science Fund for Distinguished Young Scholars (60725415).

REFERENCES


Xiang Gao was born in Jiangsu province in 1982. He received B.S. degree from Nanjing Normal University in 2004, and received Ph.D. degree in Microelectronics and Solid State Electronics at Xidian University, Xi’an, China, in 2011. His current research interests include Wireless Sensor Networks, Target Tracking and Network on Chip System.

Yintang Yang received his B.S. and M.S. degrees in Microelectronics and Solid State Electronics at Xidian University, Xi’an, China in 1982 and 1984, respectively, and received the Ph.D. degree at Xi’an Jiaotong University in 1999.

He is currently a Professor and Ph. D supervisor with School of Microelectronics. His research interests include VLSI technology, new semiconductor materials and devices, microelectronics reliability technology.

Duan Zhou received her Ph.D. degree in Computer Application Technology from School of Computer Science and Technology, Xidian University, Xi’an, China in 2009.

She is currently a Processor with School of Computer Science and Technology of Xidian University. Her research focuses on embedded system, computer control, design of integrate circuit and globally asynchronous locally synchronous (GASL) circuits.

Jianxian Zhang received his B.S. and M.S. degrees in computer science and technology at Xidian University, Xi’an, China in 2005 and 2008, respectively.

He is currently working toward the Ph.D. degree in Microelectronics and Solid State Electronics in School of Microelectronics, Xidian University, Xi’an, China.

His research focuses on communication-centric design methodologies for large-scale system on chip, with emphasis on modeling, analysis, and energy optimization for network-on-chip architectures.

Changechun Chai received his his B.S. degree in Semiconductor Device and Physics at Tsinghua University and M.S. degree in Semiconductor Physics and Device at Xidian University in 1983 and 1990, respectively, and received the Ph.D. degree in Microelectronics and Solid State Electronics at Xidian University in 1999.

He is currently a Professor and Ph. D supervisor with School of Microelectronics. His research interests include VLSI technology, new semiconductor materials and devices, microelectronics reliability technology.