

A MONTE-CARLO METHOD FOR INITIALIZING DISTRIBUTED TRACKING ALGORITHMS

Milind Borkar, Volkan Cevher and James H. McClellan

Georgia Institute of Technology
Atlanta, GA 30332-0250

ABSTRACT

Distributed processing algorithms are attractive alternatives to centralized algorithms for target tracking applications in sensor networks. In this paper, we determine an initial probability distribution of multiple target states in a distributed manner to initialize distributed trackers. Our approach is based on Monte-Carlo methods, where the state distributions are represented as a weighted set of discrete state realizations. The filter state vector consists of the target positions and velocities on the 2D plane. Our approach can determine the state vector distribution even if the individual sensors alone are not capable of observing it. The only condition is that the network as a whole be able to observe the state vector. A robust weighting strategy is formulated to account for missed detections and clutter. To demonstrate the effectiveness of the algorithm, we simulate a network with direction-of-arrival nodes and range-doppler nodes.

1. INTRODUCTION

In sensor networks, distributed processing is becoming more popular than centralized approaches [1]. This is because centralized networks with only one processing node are vulnerable if that particular node is incapacitated. The communication overhead is also significant because if all the sensing nodes are trying to transmit raw data to the central processing node, the required bandwidth increases significantly with the number of nodes. To overcome these drawbacks, a distributed processing approach is attractive.

Distributed processing stipulates processing capabilities at individual sensors. We denote a sensor that has the ability to process data and communicate with neighboring sensors in addition to sensing the environment as a smart sensor. Distributed processing eliminates the need for a central processing node. Since a smart sensor can process its own data, it need only transmit sufficient statistics in the communication channel, minimizing the communication among sensors. Communication consumes more battery power than computation, hence smart sensor networks with distributed processing have additional advantages.

In this paper, a novel method for determining initial multiple target state distributions in a smart sensor network is proposed for a distributed framework. A Monte-Carlo method is used to generate a discretized approximation to the target state distribution. This distribution is represented using hypothesized target states called particles and their associated weights. The resulting distribution

Prepared through collaborative participation in the Advanced Sensors Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-02-0008.

can be used to initialize various distributed joint tracking (DJT) algorithms such as the ones in [2–6].

The algorithm satisfies the typical constraints of a distributed system. The communication between individual sensors has fixed bandwidth. Since the data propagated between sensors is the cumulative state information, the amount of data passed between individual sensors does not increase with the number of sensors. The sensor types focused on are Direction of Arrival (DOA) nodes (e.g., acoustic arrays with known microphone positions) and range-doppler nodes (e.g., a radar sensor). However, the results are general and can be extended to networks with different sensor modalities. Each sensor runs a tracking algorithm that operates in a different state space determined by the sensor modality. We shall refer to the tracking algorithms running at the individual sensors as organic trackers. The DJT operates in a state space which may be different from the state spaces of the organic trackers at the individual nodes. We assume that each tracker is capable of detecting a new target. When an organic tracker detects a new target in its limited subspace, it transmits information throughout the network to generate the target state distribution. We also have a robust weighting strategy that can accommodate clutter as well as missing data. Moreover, communication takes place between neighboring sensors only and there is a predefined path for the information flow through the network from the first sensor to the last sensor.

2. ACOUSTIC AND RADAR TRACKERS

The two types of sensor nodes used to demonstrate the initialization algorithm are DOA sensors and Range-Doppler sensors. The DOA tracker operates in the $[\theta \ q \ \phi]'$ space where θ is the direction towards the target, q is the ratio of the target's velocity to the target's range and ϕ is the heading direction of the target. The range-doppler tracker operates in the $[r \ v_r]'$ space where r is the range to the target and v_r is the target's radial velocity [7–10].

The focus of this paper is to generate a probability distribution for the target in the $[x \ y \ v_x \ v_y]'$ space where x and y are the Cartesian coordinates of the target's location and v_x and v_y are the velocity components along the x - y directions. Notice that the true location and velocity of the target is not observable at any single node and that the organic trackers operate in different state spaces that have lower dimensionality than the DJT state space. It is assumed that organic trackers are running at the various nodes and the outputs of the organic trackers are used to generate the desired probability distribution. The sensor network is assumed to be calibrated so each sensor is aware of its own location. However, sensors need not be aware of the locations of other sensors in the network.

3. A MONTE-CARLO APPROACH FOR THE DISTRIBUTED ESTIMATION OF THE TARGETS PROBABILITY DISTRIBUTION

We choose as our proposal function

$$\pi(\mathbf{x}_t|\mathbf{z}_t) = \frac{1}{M} \sum_{m=1}^M p(\mathbf{x}_t|\mathbf{z}_{m,t}), \quad (1)$$

where \mathbf{x}_t is the target state vector, \mathbf{z}_t is the set of measurements from all nodes in the network, $\mathbf{z}_{m,t}$ is the measurement vector at the m^{th} node and M is the total number of nodes in the network. Equation (1) represents an equally weighted mixture of the individual posterior distributions from the individual nodes. It is desired that the target should be initialized in the $[x \ y \ v_x \ v_y]^T$ space. Assume the state vector is n -dimensional in general. Also, assume that at sensor m , target measurements are represented by s dimensional feature vectors, where $0 < s \leq n$. If any of these measurements are not functions of the state vector, they will not contribute any useful information for tracking and, therefore, they can be discarded. Thus, without loss of generality, we can assume each measurement is a function of the state vector.

$$\mathbf{z}_{m,t} = f_m(\mathbf{x}_t) = \begin{bmatrix} f_{m,1}(\mathbf{x}_t) \\ f_{m,2}(\mathbf{x}_t) \\ \vdots \\ f_{m,s}(\mathbf{x}_t) \end{bmatrix}. \quad (2)$$

We shall now determine the conditions $f_m(\cdot)$ must satisfy in order for one to be able to sample (1). Let $f_m(\cdot)$ be a continuously differentiable vector valued function. If and only if all measurements at a particular sensor provide complimentary information without redundancy, then

$$\det(\nabla f_m(\mathbf{x}_t)) \neq 0. \quad (3)$$

If any of the measurements provide redundant information, those particular measurements can be discarded to give a measurement space of reduced dimension and no redundancy. Therefore, we can assume all measurements provide complementary information and (3) is satisfied.

First, consider the case, when $s < n$. Given the measurements $\mathbf{z}_{m,t}$, the system in (2) is underdetermined. Therefore, there exist infinitely many solutions for \mathbf{x}_t . These solutions form a level set in the state space. In some cases, the level set can be represented by explicit equations relating the state variables. However, this may not be possible in most cases even though the level sets do exist. Let α be any solution of (2). By the Implicit Function Theorem, in the neighborhood of α , the level set $L_f(\mathbf{z}_{m,t})$ is an $n - s$ dimensional manifold. Let Λ represent the set of all such manifolds. Particles can be generated by sampling uniformly from points in Λ and adding appropriate noise determined by the measurement model.

Now consider the case, when $s = n$. By the Inverse Function Theorem, given measurements $\mathbf{z}_{m,t}$, a unique inverse function $f_m^{-1}(\cdot)$ exists in the neighborhood of $\mathbf{z}_{m,t}$ and therefore there exists a unique solution to (2) given by

$$\mathbf{x}_t = f_m^{-1}(\mathbf{z}_{m,t}). \quad (4)$$

Hence, particles can be sampled from an appropriate distribution centered at \mathbf{x}_t with variance determined by the measurement model. This implies that the particles can be sampled from the individual

posterior distributions without sharing raw data.

If the total number of nodes is M , then to sample D particles from the mixture given by (1), one can sample D/M particles from each individual posterior and combine these particles to generate a final set of D particles. However, this method has an inherent disadvantage. If one of the nodes does not detect the new target, D/M particles are spread uniformly over the entire state space for that node and these particles do not add any information to the system. Instead of sampling these particles uniformly, it is desirable to sample only from the posteriors for the nodes that have detections. Hence, more particles cover the state space of interest. These disadvantages can be eliminated by implementing the algorithm given in this paper, where a weighted resampling operation ensures that the various individual posteriors for nodes with detections are equally weighted irrespective of the total number of nodes. In the algorithm, resampling does not require synchronization of the nodes.

Once the particles are sampled, they need to be weighted. Since the data from individual nodes is not shared, the components forming the weights must be computed at each node and the cumulative weights should be transmitted. It is shown in [11] that the particle weights are given by

$$w_t^{(i)} = \frac{p(\mathbf{x}_t^{(i)}|\mathbf{z}_t)}{\pi(\mathbf{x}_t^{(i)}|\mathbf{z}_t)}. \quad (5)$$

From (1), the Bayes' rule and the fact that the measurements at the various sensors are independent given the target state, (5) can be simplified to

$$w_t^{(i)} \propto \frac{\prod_{m=1}^M p(\mathbf{z}_{m,t}|\mathbf{x}_t^{(i)})}{\sum_{m=1}^M \frac{p(\mathbf{z}_{m,t}|\mathbf{x}_t^{(i)})}{p(\mathbf{z}_{m,t})}}. \quad (6)$$

Thus, the weights for the particles can be calculated, up to a proportionality constant, by evaluating a quotient in which the numerator is the product of the data likelihoods from the individual nodes and the denominator is a weighted sum of the same likelihoods. Hence, the weights can be updated cumulatively if the numerators and denominators are communicated between nodes.

When the final particles are proposed, there is an ambiguity as to which sensor proposed a particular particle. If a particular sensor detects multiple targets, then this brings in additional complexity, since the particles can not be associated with their detectors. If a simple Gaussian likelihood function is used and the likelihood for a particle is zero at one of the sensors, then based on (6) its overall weight will also be zero. This situation occurs if even one sensor does not detect a target. In such a situation, one would not want the overall weight of the particle to be zero since the target is present with high probability if all the other sensors detect the target. To avoid this degeneracy, it is important that a robust likelihood function that accounts for missed detections be used.

The approach used here is similar to the approach used in [12]. Assume that there are M sensors in the network and consider the weighting at sensor m . Assume that sensor m detects K targets. Then, given a particle, or a hypothesized target state $\mathbf{x}_t^{(i)}$, the measurements $\mathbf{z}_{m,k,t}$, $k = 1, \dots, K$, could have been generated either by a target or by clutter. The clutter distribution is assumed to be Poisson with spatial density λ . The probability of miss is given by q . It is assumed that there is an equal probability for each of the K measurement vectors to be the true measurement and the true target measurement is Gaussian distributed about the true target state.

ALGORITHM:

- **Sequentially Sampling the Proposal Function**
 - D = Number of particles used for initialization.
 - $S(i)$ = Sensor i , where $i = 1, \dots, M$
 - $\tilde{w} = 0$
 - If $S(1)$ has a detection,
 - sample D particles based on the posterior distribution at $S(1)$,
 - each particle will have equal weight,
 - $\tilde{w} = \tilde{w} + 1$,
 - Else,
 - set all particles equal to 0.
- Send particles and \tilde{w} to $S(2)$.
- For $i = 2, \dots, M$
- current sensor is $S(i)$,
 - accept D particles and \tilde{w} from $S(i - 1)$,
 - give each received particle a weight of \tilde{w} .
 - If $S(i)$ has a detection,
 - * sample D new particles based on the posterior distribution at $S(i)$,
 - * each new particle will have equal weight,
 - give each new particle a weight of 1
 - * from the $2D$ particles, obtain D particles by using a weighted sampling with replacement,
 - * each particle will now have equal weight,
 - * $\tilde{w} = \tilde{w} + 1$.
 - Send particles and \tilde{w} to $S(i + 1)$.
-

Thus, as shown in [12], the likelihood function can be simplified as: $p(\mathbf{z}_{m,t} | \mathbf{x}_t^{(i)}) \propto$

$$1 + \frac{1 - q}{\sqrt{(2\pi)^s |\Sigma|} q \lambda} \sum_{k=1}^K e^{-\frac{(\mathbf{z}_{m,k,t} - g(\mathbf{x}_t^{(i)}))^T \Sigma^{-1} (\mathbf{z}_{m,k,t} - g(\mathbf{x}_t^{(i)}))}{2}}, \quad (7)$$

where s is the dimensionality of the measurement vector at sensor m , Σ is the covariance of the measurement model, and $g(\cdot)$ is the mapping from the target state to the measurement state. Putting all this together, we formulate the given algorithm.

4. SIMULATIONS

Figure 1 demonstrates the initialization strategy for one target that appears at (50, 50)m with a velocity vector of (14, 14)m/s. For the purpose of this simulation, we assume that only 4 sensors in the sensor network detected the target: two acoustic sensor nodes, located at (200, -200)m and (1200, 200)m, and two radar nodes, located at (500, 400)m and (800, -300)m. Organic trackers at these four nodes detect this target and produce estimates in their own state spaces. $D = 2000$ particles were used to adequately sample the state space of interest. Figures 1(a) to 1(d) represent the sequential particle proposal stages of the algorithm. Although all particles are four dimensional, most of the subfigures in Fig. 1

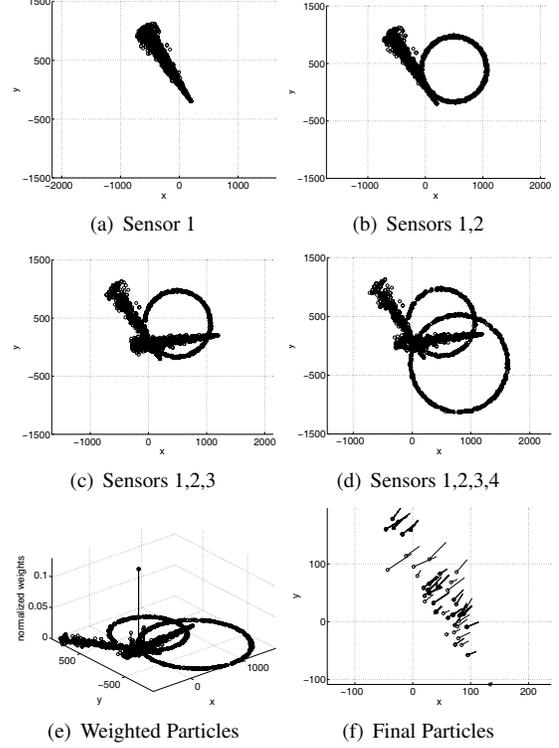


Fig. 1. Simulation for single target

show only the x - y locations of the particles. The velocity information is then shown in Fig. 1(f).

Weights are calculated for the final particles shown in Fig. 1(d). Particles along with their weights are shown in Fig. 1(e) and this represents the probability distribution of the target in the x - y space. As expected, the distribution is highly peaked around the true target state. Estimates of the true target state can be made based on this weighted set of particles. These estimates can be used to initialize any distributed tracking algorithm.

It is observed that the majority of particles have extremely low weights and do not contribute any useful information. To eliminate these particles and replicate those with high weights, the particles are sampled with replacement according to their weights to give the set of particles in Fig. 1(f). Here the circles represent the particle positions and the lines extending from the circles represent the magnitude and direction of the velocities. It can be seen that the final set of particles is concentrated around the true target state at $[50\text{m}, 50\text{m}, 14\text{m/s}, 14\text{m/s}]^T$. The mean of these particles is $[46.5\text{m}, 69.6\text{m}, 13.7\text{m/s}, 15.3\text{m/s}]^T$ and the variance for each state is $[(35\text{m})^2, (55\text{m})^2, (4\text{m/s})^2, (4.4\text{m/s})^2]^T$. These variances could be reduced by increasing the number of particles. However, the main determinant of the final estimation variances is the distributed nature of the problem.

The final set of particles were used to initialize a distributed particle filter tracker that the authors are developing. In Fig. 2, the true track is given by the solid line, whereas the estimated track is given by the dashed line. For comparison, tracking is also performed when the tracker is initialized using a uniform grid based approach. In this approach, 6000 particles are proposed uniformly over the entire state space of interest and the measure-

ments at the individual sensors are simply used for weighting the proposed particles. In this case, the track is represented by the dotted line. The ∇ symbols represent the radar nodes and the \times symbols represent the acoustic nodes. Gaussian noise was added to the measurements at the various nodes. The noise had a covariance of $\text{diag}([3^\circ, 0.03\text{sec}^{-1}, 12^\circ]^2)$ for the acoustic sensor nodes and $\text{diag}([9\text{m}, 0.6\text{m/s}]^2)$ for the radar nodes. As observed, the tracking algorithm is rather accurate when initialized using the proposed Monte-Carlo approach. Tracking results are very poor when the uniform grid based approach is used for initialization even though three times as many particles are used to initialize the target. Because the initial target distribution does not provide effective support, the range measurements of the radar sensors do not contribute to the DJT performance.

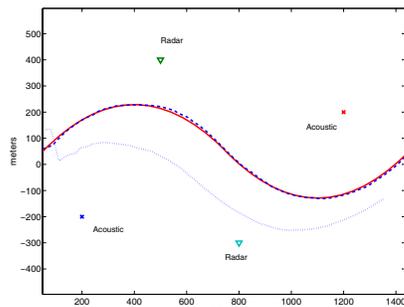


Fig. 2. Tracking Simulation

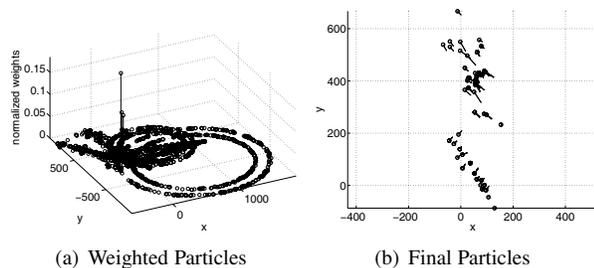


Fig. 3. Simulation for two targets using 2000 particles

Figure 3 simulates the initialization for two targets. Here, the sensor locations are the same as earlier, and the true target states are given by $[50\text{m}, 50\text{m}, 14\text{m/s}, 14\text{m/s}]^T$ and $[50\text{m}, 400\text{m}, 14\text{m/s}, -14\text{m/s}]^T$. The weighted particle set is shown in Fig. 3(a). The distribution of the target state is clearly seen in Fig. 3(b), which represents the set of particles that survive the weighted resampling operation. As expected, the particle distribution is concentrated about the true target states.

5. CONCLUSIONS AND FUTURE WORK

A method for generating the probability distribution that models missed detections and clutter for multiple targets in a distributed smart sensor network is proposed. A Monte-Carlo method is used to sequentially sample the state space of interest to generate particles, and then a robust weighting function is used to represent

the degree of belief in each particle. This weighting function can accommodate multiple targets, clutter and missing data. The final target state distribution is represented as a weighted set of particles that can be used to make various inferences about the target state and also to initialize various distributed tracking algorithms.

For the purpose of this paper, the prior target state distribution was assumed to be uninformative and chosen to be uniform. Future work will also consider the case of informative priors to generate distributions reflecting prior knowledge of the true target state.

6. REFERENCES

- [1] D. Estrin, R. Govindan, and J. Heidemann, "Scalable coordination in sensor networks," Tech. Rep. USC Tech Report 99-692, USC/ISI, 1999.
- [2] Y. Wong, J. Wu, L. Ngoh, and W. Wong, "Collaborative data fusion tracking in sensor networks using monte carlo methods," in *Proceedings. 29th Annual IEEE International Conference on Local Computer Networks*, 2004.
- [3] M. Liggins II, C. Chong, I. Kadar, M. Alford, V. Vannicola, and S. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," in *Proceedings of the IEEE*, 1997.
- [4] P. Storms, J van Veelen, and E. Boasson, "A process distribution approach for multisensor data fusion systems based on geographical dataspace partitioning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, pp. 14–23, Jan. 2005.
- [5] S. Balasubramanian, I. Elangovan, S. Jayaweera, and K. Namuduri, "Distributed and collaborative tracking for energy-constrained ad-hoc wireless sensor networks," *IEEE Wireless Communications and Networking Conference*, vol. 3, pp. 1732–7, 2004.
- [6] J. Liu, M. Chu, J. Liu, J. Reich, and F. Zhao, "Distributed state representation for tracking problems in sensor networks," *Third International Symposium on Information Processing in Sensor Networks*, pp. 234–42, 2004.
- [7] V. Cevher and J. H. McClellan, "General direction-of-arrival tracking with acoustic nodes," *IEEE Trans. on Signal Processing*, vol. 53, pp. 1–12, Jan. 2005.
- [8] Y. Zhou, P.C. Yip, and H. Leung, "Tracking the direction-of-arrival of multiple moving targets by passive arrays: Algorithm," *IEEE Trans. on Signal Processing*, vol. 47, no. 10, pp. 2655–2666, October 1999.
- [9] S. Hong, R. Evans, and H. Shin, "Optimization of waveform and detection threshold for range and range-rate tracking in clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, pp. 17–33, 2005.
- [10] E. Hughes and M. Lewis, "Intelligent agents for radar systems," *Electronics Systems and Software*, vol. 3, pp. 39–43, Feb.-March 2005.
- [11] A. Doucet, "On sequential simulation-based methods for Bayesian filtering," Tech. Rep. CUED/F-INFENG/TR.310, Department of Engineering, University of Cambridge, 2001.
- [12] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*, Academic-Press, 1988.