ESTIMATING TARGET STATE DISTRIBUTIONS IN A DISTRIBUTED SENSOR NETWORK USING A MONTE-CARLO APPROACH

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 address the issue of determining a 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 discrete set of weighted particles. The target state vector is the target positions and velocities in the 2D plane. Our approach can determine the state vector distribution even if the individual elements are not capable of observing it. The only condition is that the network as a whole can observe the state vector. A robust weighting strategy is formulated to account for mis-detections and clutter. To demonstate the effectiveness of the algorithm, we use direction-of-arrival nodes and range-doppler nodes.

1. INTRODUCTION

In sensor networks, distributed processing is becoming more popular than the centralized approaches [1]. This is because in centralized networks, since there is only one processing node in the network, if that particular node is incapacitated, the entire system fails. The communication overhead is also significant. Moreover, 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 in addition to sensing as a smart sensor. Distributed processing eliminates the need for a central processing node. Thus the system is not fully dependent on a single node for processing thus eliminating the computational bottleneck. Since a smart sensor can process its own data, it only transmits 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 in 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 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 information propagated between sensors is the cumulative state information, the amount of information passed between individual sensors does not increase. 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 different sensors as the 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.

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.

The organization of the paper is as follows. Section 2 briefly introduces the acoustic and radar trackers. Section 3 proposes a Monte-Carlo approach for the distributed estimation of the target state distribution. Section 4 demonstrates the effectiveness of the proposed algorithms on synthetic data. Conclusions and future work follow in Section 5.

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 [θ q ϕ]' space where θ is the direction towards the target, q is the ratio of the targets velocity to the targets 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 targets radial velocity. Detailed descriptions about these trackers can be found in [7–15].

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 targets 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 of the individual nodes and that the organic trackers operate in dif-

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.

ferent state spaces that have lower dimensions than the state space in which the targets distribution is desired. This means there is a many to one mapping from the states used by the organic trackers to the state space in which the final target distribution is generated. It is assumed that the organic trackers are available at the different 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.

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

To have an optimal particle distribution, one must sample from the true posterior distribution [16]. Using Bayes' rule, the posterior distribution can be expressed as

$$p(\mathbf{x}_t|\mathbf{z}_t) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{z}_t)} \tag{1}$$

where $\mathbf{x}_t$ is the target state at time t and $\mathbf{z}_t$ is the vector of measurements from all M sensors at time t. We assume that the measurements at the individual nodes are independent conditioned on the state. Hence, the combined data likelihood for all sensors can be factored into the product of the data likelihoods at the individual sensor nodes. Since no prior information is available, $p(x_t)$ is uniform and is dropped from the equation. $p(z_t)$ is simply a proportionality constant since it does not depend on the state. Therefore, (1) can be simplified as

$$p(\mathbf{x}_t|\mathbf{z}_t) \propto \prod_{m=1}^{M} p(\mathbf{z}_{m,t}|\mathbf{x}_t) \tag{2}$$

where $\mathbf{z}_{m,t}$ is the measurement from the m^{th} sensor at time t. We choose not to communicate raw data between nodes to limit communication bandwidth. Thus determining the posterior distribution analytically is impossible. Therefore, we chose as our proposal function

$$\pi(\mathbf{x}_t|\mathbf{z}_t) = \frac{1}{M}\sum_{m=1}^{M} p(\mathbf{x}_t|\mathbf{z}_{m,t}) \tag{3}$$

Our choice of the proposal function is an equally weighted mixture of the individual posterior distributions from the different nodes. For the different nodes, particles can be sampled from the individual posterior distributions as follows:

For DOA nodes:

$$r^{(i)} \sim U(0, r_{max}) \tag{4}$$

$$\theta^{(i)} \sim N(\theta_t, \Sigma_{\theta_t}) \tag{5}$$

$$q^{(i)} \sim N(q_t, \Sigma_{q_t}) \tag{6}$$

$$\phi^{(i)} \sim N(\phi_t, \Sigma_{\phi_t}) \tag{7}$$

$$x_t^{(i)} = r^{(i)}\cos(\theta^{(i)}) \tag{8}$$

$$y_t^{(i)} = r^{(i)}\sin(\theta^{(i)}) \tag{9}$$

$$v_{x_t}^{(i)} = q^{(i)}r^{(i)}\cos(\phi^{(i)}) \tag{10}$$

$$v_{y_t}^{(i)} = q^{(i)}r^{(i)}\sin(\phi^{(i)}) \tag{11}$$

For Range-Doppler nodes:

$$r^{(i)} \sim N(r_t, \Sigma_{r_t}) \tag{12}$$

$$\theta^{(i)} \sim U(0, 2\pi) \tag{13}$$

$$v_r^{(i)} \sim N(v_{r_t}, \Sigma_{v_{r_t}}) \tag{14}$$

$$v_t^{(i)} \sim U(-(v_{max}^2 - (v_r^{(i)})^2)^{0.5}, (v_{max}^2 - (v_r^{(i)})^2)^{0.5}) \tag{15}$$

$$x_t^{(i)} = r^{(i)}\cos(\theta^{(i)}) \tag{16}$$

$$y_t^{(i)} = r^{(i)}\sin(\theta^{(i)}) \tag{17}$$

$$v_{x_t}^{(i)} = v_r^{(i)}\cos(\theta^{(i)}) + v_t^{(i)}\sin(\theta^{(i)}) \tag{18}$$

$$v_{y_t}^{(i)} = v_r^{(i)}\sin(\theta^{(i)}) - v_t^{(i)}\cos(\theta^{(i)}) \tag{19}$$

Estimates of $(\theta_t, \Sigma_{\theta_t})$, (q_t, Σ_{q_t}), and $(\phi_t, \Sigma_{\phi_t})$ are available from the organic trackers at the DOA nodes. Similarly, estimates of (r_t, Σ_{r_t}) and $(v_{r_t}, \Sigma_{v_{r_t}})$ are available from the organic trackers at the range-doppler nodes. There is a range ambiguity in the DOA node. Similarly there is a DOA ambiguity and a tangential velocity ambiguity in the range-doppler node. Therefore, these values are drawn from appropriate uniform distributions. Here, r_{max} is the assumed maximum range at which the target is visible to the DOA node, and v_{max} is the assumed maximum velocity of the target. Radial velocity is considered positive if the target is moving away from the node. Tangential velocity is considered positive if the tangential component points in the counterclockwise direction.

Using (4) through (19) one can sample particles from the individual posteriors. If the total number of nodes is M, then to sample D particles from the mixture given by (3), one can sample D/M particles from each individual posterior and combine these particles to generate the 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 uniformly sampled 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 more informative 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 following step 1 of the following algorithm, 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. Resampling does not require synchronization of the nodes.

Once the particles are sampled, they need to be weighted. Since the data from various nodes is not being shared, the components forming the weights must be computed at each node and the cumulative information should be transmitted. It is shown in [16] that the particle weights are given by

$$w_t^{(i)} = \frac{p(\mathbf{x}_t^{(i)}|\mathbf{z}_t)}{\pi(\mathbf{x}_t^{(i)}|\mathbf{z}_t)} \tag{20}$$

From (2) and (3), (20) can be simplified as

$$w_t^{(i)} \propto \frac{\prod_{m=1}^{M} p(\mathbf{z}_{m,t}|\mathbf{x}_t)}{\sum_{m=1}^{M} p(\mathbf{x}_t|\mathbf{z}_{m,t})} \tag{21}$$

From Bayes' rule, we get

$$p(\mathbf{x}_t^{(i)}|\mathbf{z}_{m,t}) = \frac{p(\mathbf{z}_{m,t}|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)})}{p(\mathbf{z}_{m,t})} \tag{22}$$

Since no pior information about the target state is available, $p(x_t)$ is uniform over the entire space and can be dropped from the equation. Thus (21) simplifies to

$$w_t^{(i)} \propto \frac{\prod_{m=1}^{M} p(\mathbf{z}_{m,t}|\mathbf{x}_t)}{\sum_{m=1}^{M} \frac{p(\mathbf{x}_t|\mathbf{z}_{m,t})}{p(\mathbf{z}_{m,t})}} \tag{23}$$

Thus, the weights for the particles can be calculated, within a proportionality constant, by evaluating a quotient in which the numerator is the product of the data likelihoods from the different nodes and the denominator is the weighted sum of the same likelihoods. This way, the weights are also communicated in a cumulative manner.

When the final particles are proposed, there is an ambiguity as to which sensor proposed a particular particle. If a particular sensor has multiple detections, 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 (23) 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. To avoid this degeneracy, it is important that a robust likelihood function that accounts for target misses is used.

The approach used here is similar to the approach used in [17, 18] Assume that there are M sensors. The focus here will be on the weighting at sensor m where $m = 1, ..., M$. Assume that sensor m has K measurements. Then, given a particle or a hypothesized target state $\mathbf{x}_t$, measurements $\mathbf{z}_{m,k,t}$, $k = 1, ..., K$, could have been generated either by the target or by clutter. The clutter distribution is assumed to be Poisson with spatial density λ. The probability of miss is set equal to q. It is assumed that there is an equal probability for each of the K measurements to be the true measurement and the true target measurement is Gaussian distributed about the true target state. Thus, as shown in [18] the likelihood function can be simplified as:

$$p(\mathbf{z}_{m,t}|\mathbf{x}_t^{(i)}) \propto 1 + \frac{1}{\sqrt{(2\pi)^n|\Sigma|}q\lambda} \cdot \sum_{k=1}^{K} exp(-0.5(\mathbf{z}_{m,k,t} - g(\mathbf{x}_t^{(i)}))^T\Sigma^{-1}(\mathbf{z}_{m,k,t} - g(\mathbf{x}_t^{(i)}))) \tag{24}$$

where n is the dimensionality of the measurement at sensor k, Σ is the covariance of the Gaussian distribution and $g(.)$ is the mapping from the target state to the measurement state.

Steps 2 and 3 of the following algorithm explain the weighting step. The set of particles along with their associated weights give a discrete representation of the probability distribution of the target in the desired state space.

- **ALGORITHM:**
  D = Number of particles used for initialization.
  $S(i)$ = Sensor i, where $i = 1, ..., M$
  $K(i)$ = Target i, where $i = 1, ..., K$
  $wNum$ = Numerator of weights.
  $wDen$ = Denominator of weights.
  w = particle weights.
  $x_t^{(i)}$ = particle i at time t.

- **STEP 1: Sequentially Sampling the Proposal Function**
  w_t = 0
  If $S(1)$ has a detection,
  - Spread D particles uniformly along the detection geometry in X-Y space
  - Each particle will have equal weight
  - $w_t = w_t + 1$

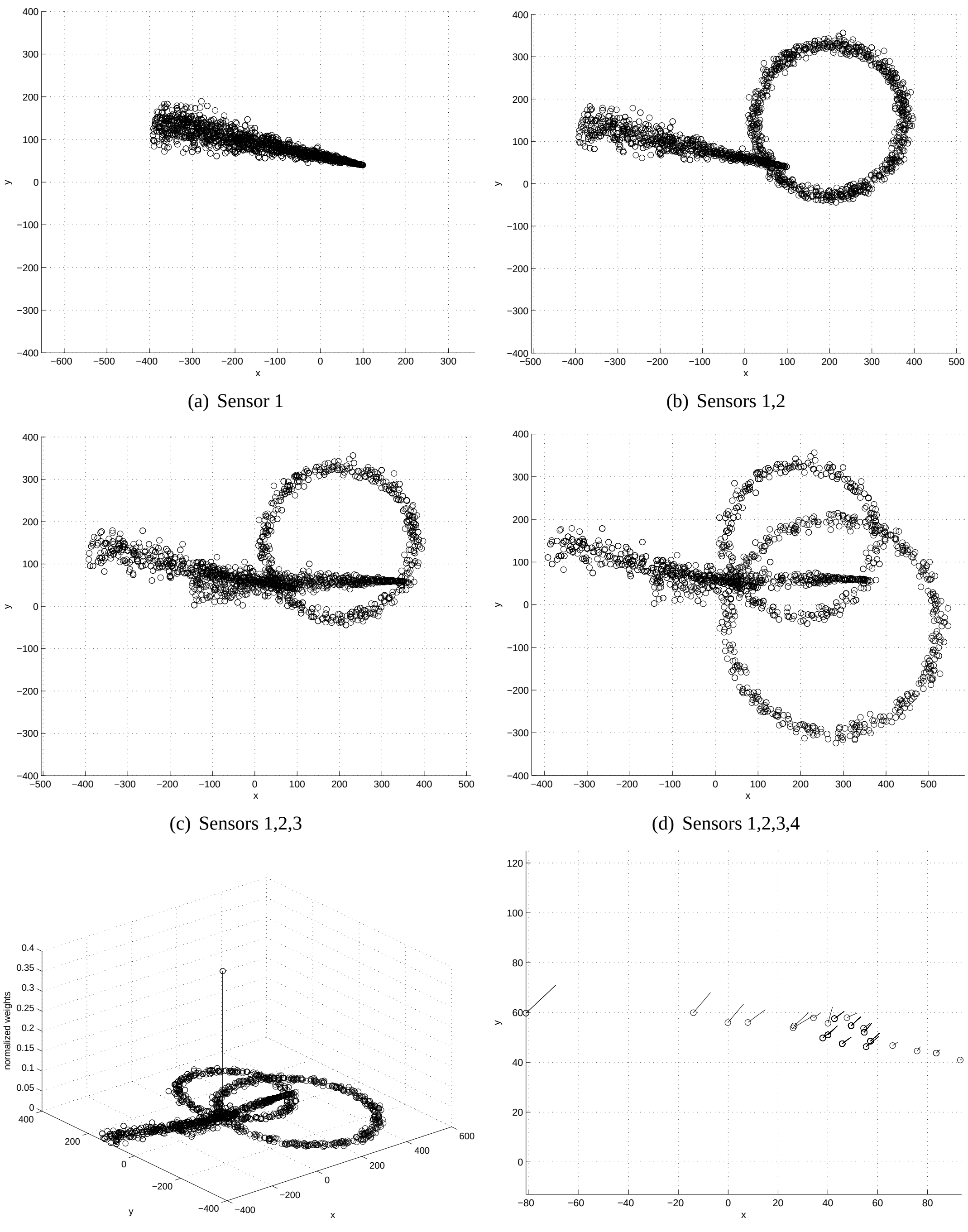
  Else
  - set all particles equal to 0

  Send particles and w_t to $S(2)$
  For $i = 2, ..., M$
  - Current sensor is $S(i)$
  - Accept D particles and w_t from $S(i-1)$
  - Give each received particle a weight of w_t
  - If $S(i)$ has a detection
    * Spread D new particles uniformly along the detection geometry
    * 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
    * $w_t = w_t + 1$
  - Send particles and w_t to $S(i+1)$

- **STEP 2: Weighting the Particles and Back Propagating Final Particles**
  For $i = 0, ..., M-1$
  - Current sensor is $S(M-i)$
  - If $i > 0$
    * Accept particles, $wNum$ and $wDen$ from $S(M-i+1)$
  - Else
    * $wNum = 1$
    * $wDen = 0$
  - For $i = 1, ..., D$
    * $wNum^{(i)} = wNum^{(i)} \cdot p(\mathbf{z}_{t,M-i}|x_t^{(i)})$
    * $wden^{(i)} = wden^{(i)} + \frac{p(\mathbf{z}_{t,M-i}|x_t^{(i)})}{p(\mathbf{z}_{t,M-i})}$
  - Send particles, $wNum$ and $wDen$ to $S(M-i-1)$

- **STEP 3: Propagate Final Weights**
  Current sensor is $S(1)$
  For $i = 1, ..., D$
  - $w^{(i)} = \frac{wNum^{(i)}}{wDen^{(i)}}$

  Send w to $S(2)$
  For $i = 2, ..., M$
  - Accept w from $S(i-1)$
  - Send w to $S(i+1)$



(a) Sensor 1

(b) Sensors 1,2

(c) Sensors 1,2,3

(d) Sensors 1,2,3,4

(e) Weighted Particles

(f) Final Particles

Fig. 1. Simulations for Single Target Case

4. SIMULATIONS

Assume a target appears at an X-Y location (50m,50m) with velocity of 4 m/s in the x-direction and 4 m/s in the y-direction. There are a total of four sensors in the field. 2 acoustic sensor nodes are located at (100m,40m) and (350m,60m) whereas 2 radar nodes are located at (200m,150m) and (275m,-50m). Organic trackers at the four nodes detect this target and produce estimates in their own state spaces.

For the purpose of this simulation, $D = 2000$ particles were used in order to adequately sample the state space of interest. Fig. 1(a) to 1(d) represent the sequential particle proposal stage of the algorithm. Although all particles are four dimensional, all subfigures in Fig. 1 only show the X-Y locations of the particles. The velocity information is only shown in Fig. 1(f). In figure 1(a), sensor 1, which is a DOA sensor, detects the target at a particular angle and distributes 2000 particles along that angle up to an assumed maximum range. These particles are propagated to sensor 2, a range-doppler sensor. Sensor 2 receives the particles from sensor 1 and gives these particles a weight of 1 since they represent information from a single sensor. Sensor 2 detects the target at a particular range. Since angle information is not available, sensor 2 distributes another 2000 particles about a circle with radius equal to the detected range and center at the sensor position. Out of the 4000 particles at sensor 2, 2000 particles are sampled uniformly with replacement. These particles are shown in Fig. 1(b) and are propagated to sensor 3, another DOA sensor. Sensor 3 receives the particles from sensor 2 and gives these particles a weight of 2 since these particles represent the combined information from two sensors. Sensor 3 detects the target at a particular angle and distributes another 2000 particles along that angle. These new particles have a weight of 1. From the 4000 particles at sensor 3, a weighted sampling with replacement is used to generate 2000 equally weighted particles. These particles are shown in Fig. 1(c) and are propagated to sensor 4, another range-doppler sensor. Sensor 4 receives the particles from sensor 3 and gives them a weight of 3 since they represent the combined information from 3 sensors. Then sensor 4 detects the target at a particular range and distributes another 2000 particles along a circle with radius equal to the detection range and center at the sensor location. These new particles are given a weight of 1. From the 4000 particles at sensor 4, 2000 particles are obtained using a weighted sampling with replacement. These final particles are plotted in Fig. 1(d) and are propagated back to all the sensors.

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 about 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 multiply 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 [50,50,4,4]'.

The final set of particles were used to initialize a distributed joint tracker we have been developing. The track estimate can be seen in Fig. 2. The true track is given by the solid line and the estimated track is given by the dashed line. As observed, the tracking algorithm is very accurate when initialized using our Monte-Carlo approach.

Simulations using two targets are shown in Fig. 3. Here, the sensor locations are unchanged and the true target states are given by [50,50,4,4]' and [50,150,4,-4]'. The weighted particle set is shown in Figure 3a. The distribution of the target state is clearly seen in Figure 3b which represents the set of particles that survive a weighted resampling operation. As expected, the particle distribution is concentrated about the true target states.

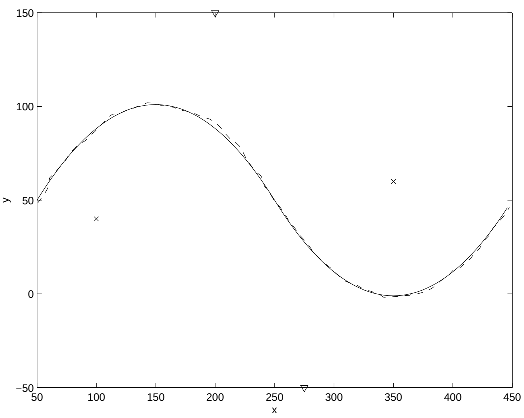


Fig. 2. Tracking Simulation

5. CONCLUSIONS AND FUTURE WORK

A method for generating the probability distribution that models missing targets 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 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. This set of weighted particles can be used to make various inferences about the target state and also to initialize various distributed tracking algorithms.

Future work will focus on a fully automated distributed tracking algorithm which will use the method outlined in this paper as an initialization strategy.

6. REFERENCES

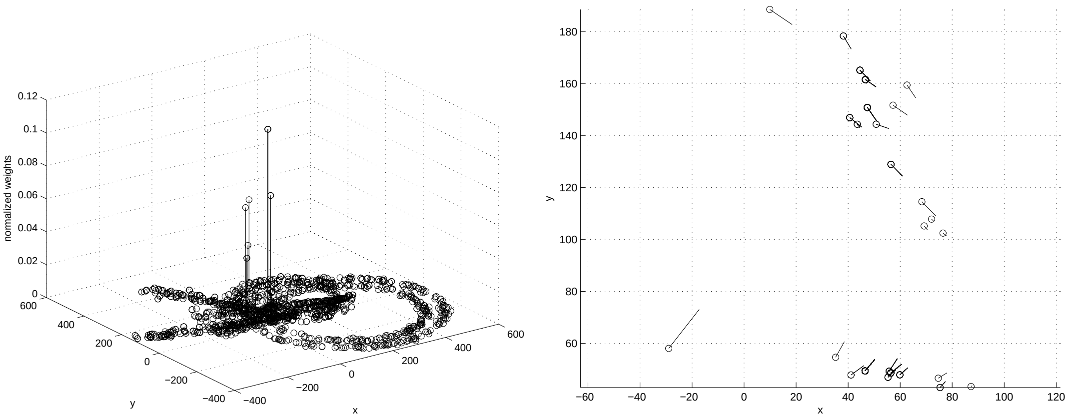
[1] R. Govindan D. Estrin and J. Heidemann, "Scalable coordination in sensor networks," Tech. Rep. USC Tech Report 99-692, USC/ISI, 1999.

[2] L. Ngoh Y. Wong, J. Wu 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] I. Kadar M. Alford V. Vannicola M. Liggins II, C. Chong and S. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," in *Proceedings of the IEEE*, 1997.

[4] J van Veelen P. Storms 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. Jayaweera S. Balasubramanian, I. Elangovan 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.



(a) Weighted Particles (b) Final Particles

Fig. 3. Simulations for Two Target Case

[6] J. Liu J. Reich J. Liu, M. Chu 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," To appear in *IEEE Trans. on Signal Processing*.

[8] M. Orton and W. Fitzgerald, "A Bayesian approach to tracking multiple targets using sensor arrays and particle filters," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 216–223, February 2002.

[9] R.R. Allen and S.S. Blackman, "Implementation of an angle-only tracking filter," in *SPIE Proc.*, 1991, vol. 1481, pp. 292–303.

[10] A. Farina, "Target tracking with bearings-only measurements," *Elsevier Signal Processing*, vol. 78, pp. 61–78, 1999.

[11] 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.

[12] J. Sanchez-Araujo and S. Marcos, "An efficient PASTd-algorithm implementation for multiple direction of arrival tracking," *IEEE Trans. on Signal Processing*, vol. 47, pp. 2321–2324, August 1999.

[13] V.J. Aidala, "Kalman filter behavior in bearings-only tracking applications," *IEEE Trans. on Aerospace and Electronic Systems*, vol. AES-15, pp. 29–39, January 1979.

[14] R. Evans S. Hong 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.

[15] E. Hughes and M. Lewis, "Intelligent agents for radar systems," *Electronics Systems and Software*, vol. 3, pp. 39–43, Feb.-March 2005.

[16] A. Doucet, "On sequential simulation-based methods for Bayesian filtering," Tech. Rep. CUED/F-INFENG/TR.310, Department of Engineering, University of Cambridge, 2001.

[17] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*, Academic-Press, 1988.

[18] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, pp. 5–28, 1998.