



Jaward, M. H., Bull, D. R., & Canagarajah, C. N. (2006). Distributed tracking with sequential Monte Carlo methods for manoeuvrable sensors. In *IEEE Nonlinear Statistical Signal Processing Workshop 2006*, Cambridge, UK. (pp. 113 - 116). Institute of Electrical and Electronics Engineers (IEEE). 10.1109/NSSPW.2006.4378832

Link to published version (if available):
[10.1109/NSSPW.2006.4378832](http://dx.doi.org/10.1109/NSSPW.2006.4378832)

[Link to publication record in Explore Bristol Research](#)
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/pure/about/ebr-terms.html>

Take down policy

Explore Bristol Research is a digital archive and the intention is that deposited content should not be removed. However, if you believe that this version of the work breaches copyright law please contact open-access@bristol.ac.uk and include the following information in your message:

- Your contact details
- Bibliographic details for the item, including a URL
- An outline of the nature of the complaint

On receipt of your message the Open Access Team will immediately investigate your claim, make an initial judgement of the validity of the claim and, where appropriate, withdraw the item in question from public view.

DISTRIBUTED TRACKING WITH SEQUENTIAL MONTE CARLO METHODS FOR MANOEUVRABLE SENSORS

M.H. Jaward, D. Bull and N. Canagarajah

Department of Electrical & Electronic Engineering
University of Bristol, Bristol, UK
m.h.jaward@bristol.ac.uk

ABSTRACT

Nonlinear distributed tracking for a single target is addressed in this paper. This problem consists of tracking a target of interest while moving the sensors to 'best' positions according to an criteria appropriate for the problem. Both target tracking and manoeuvring of sensors are carried out jointly using a novel Sequential Monte Carlo technique. The proposed technique is illustrated using a bearing-only problem and simulations are used to compare the performance of the proposed technique with distributed tracking using fixed sensors.

1. INTRODUCTION

Sensors are valuable resources, making it imperative that they are deployed in a manner that maximises their effectiveness. This requires the determination of the optimal location and motion of the sensor swarm within the environment under consideration. As explained in [1], ideal sensor placement is a problem exhibiting non-linear combinatorial properties since the position of each individual sensor in the set of available sensors will affect the optimal placement of the other members of the set. Therefore, for a even a small number of sensors it is computationally onerous to exhaustively search all possible solutions for the optimum and the problem becomes one of search-based optimisation. Reference [1] reports that Genetic Algorithms (GA) are the most efficient optimisation techniques compared with other methods such as random search, dynamic hill climbing and simulated annealing. But the target tracking and the sensor location optimisation are done separately. Karan et al. [2] consider the problem of tracking with multiple asynchronous drifting sonobuoys using an Extended Kalman Filter (EKF). Marrs [3] solves the above problem using a sequential Monte Carlo method.

This paper is organised as follows: Section 2 presents the system model used in this paper and section 3 reviews briefly the Sequential Monte Carlo on which the proposed algorithm is based on. Section 4 introduces the proposed distributed

This work forms a part of the DIF-DTC project whose funding and support is gratefully acknowledged.

tracking algorithm and the performance of the proposed algorithm is illustrated in section 5.

2. SYSTEM MODEL

In this paper, we are concerned with the problem of performing on-line state estimation for multi-dimensional signals that can be modelled using Markovian state-space models that are nonlinear and non-Gaussian. The unobserved global state $\{\mathbf{x}_t; t \in N\}$ is modelled as a Markov process with initial distribution $p(\mathbf{x}_0)$ and transition probability $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. The observations $\{\mathbf{y}_t; t \in N\}$ are assumed to be conditionally independent (in time) given the process \mathbf{x}_t and of marginal distribution $p(\mathbf{y}_t|\mathbf{x}_t)$. We denote by $\mathbf{X}_t = \{\mathbf{x}_0, \dots, \mathbf{x}_t\}$ and by $\mathbf{Y}_t = \{\mathbf{y}_0, \dots, \mathbf{y}_t\}$, respectively, the system state and the observations up to time t . The measurements \mathbf{y}_t are recorded by K sensors, and we use \mathbf{y}_t^k to denote the subset of observations made by the k -th sensor.

This work jointly estimates the sensor locations while tracking. Although the algorithm developed is valid for wider range of applications, we concentrate our investigation on the problem of tracking a manoeuvring target in a 2-D plane. The manoeuvring target motion is modelled using a Markov jump system of the following form:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{v}_t) \quad (1)$$

Here $f(\cdot)$ is the system function and \mathbf{u}_t is a discrete time index indicating different modes of the target. The augmented state vector of 2-D positions of the target and sensors is \mathbf{x}_t and \mathbf{v}_t is the noise term. The proposed algorithm is illustrated for the bearing only tracking problem. This problem assumes that we have nodes which are capable of measuring the angle of the target's position relative to the sensor node ϕ_t .

3. PARTICLE FILTER ALGORITHM

Sequential Monte Carlo techniques also known as particle filtering and condensation algorithm have been described in length in the literature [4]. In what follows, we give a sum-

mary of the framework and conceptual ideas behind the sequential Monte Carlo techniques.

For tracking the target of interest, the posterior state distribution $p(\mathbf{x}_t|\mathbf{Z}_t)$, also known as filtering distribution has to be calculated at each time step. In Bayesian sequential estimation the filtering distribution can be computed according to the two step recursion:

prediction step

$$p(\mathbf{x}_t|\mathbf{Y}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{Y}_{t-1})d\mathbf{x}_{t-1}, \quad (2)$$

and

filtering step

$$p(\mathbf{x}_t|\mathbf{Y}_t) \propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_{t-1}), \quad (3)$$

where the prediction step follows from marginalisation, and the new filtering distribution is obtained through a direct application of the Bayes' rule. This recursion requires the specification of a dynamic model describing the state evolution, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and a model that gives the likelihood of any state in the light of the current observation, $p(\mathbf{y}_t|\mathbf{x}_t)$. The recursion is initialised with some distribution for the initial state $p(\mathbf{x}_0)$. Once the sequence of filtering distribution is known, point estimates of the state can be obtained according to any appropriate function, leading for example to the Maximum a Posteriori (MAP) estimate, $\arg \max_{\mathbf{x}_t} p(\mathbf{x}_t|\mathbf{Y}_t)$, and to the Minimum Mean Square Error (MMSE) estimate, $\int \mathbf{x}_t p(\mathbf{x}_t|\mathbf{Y}_t) d\mathbf{x}_t$.

The basic idea behind the particle filters is very simple. Starting with a weighted set of samples $\{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^N$ approximately distributed according to $p(\mathbf{x}_{t-1}|\mathbf{Y}_{t-1})$, new samples are generated from a suitably chosen proposal distribution, which may depend on the previous state and the new measurements, i.e., $\mathbf{x}_t^{(i)} \sim q_p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)$, $i = 1, \dots, N$. To maintain a consistent sample the new importance weights are set to

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q_p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)}, \quad (4)$$

with $\sum_{i=1}^N w_t^{(i)} = 1$. The new particle set $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^N$ is then approximately distributed according to $p(\mathbf{x}_t|\mathbf{Y}_k)$. The performance of the particle filter depends on the quality of the proposal distribution. In this paper, we use the state evolution model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ as proposal distribution and this makes the new importance weights in (4) become proportional to the corresponding particle likelihoods. This implementation of the sequential Monte Carlo method corresponds to the bootstrap filter as proposed in [5]. This leads to a very simple algorithm, requiring only the ability to simulate from the evolution model and to evaluate the likelihood.

4. DISTRIBUTED TRACKING ALGORITHM FOR MANEOURABLE SENSORS

In this section, we introduce a distributed tracking algorithm for manoeuvrable sensors. Specifically, this algorithm provides the Bayesian optimal target locations while manoeuvring the sensors to 'best' positions. The cost involved in manoeuvring is based on physical constraints as well as constraints on cost and resource consumption. This cost can be computed using information theoretic measures such as statistical entropy or using simpler measures such as Mahalanobis or Euclidean measures. The entropy-based definition, while mathematically precise, is difficult to compute in practice since we need to have the measurement before deciding how useful the measurement is. A more practical alternative is to estimate the usefulness of a measurement based only on characteristics of a sensor such as its location or sensing modality. Such a distance measure can be obtained using the Mahalanobis distance, a distance measure normalised by the uncertainty covariance. This gives a statistical distance measure between a sensor node and the target being tracked. In [6], similar distance measures are used to solve the selecting the useful set of sensors out of all available sensors.

The algorithm developed in this paper assumes a semi-distributed architecture with feedback. For simplicity, in this paper we assume that the number of sensors is fixed (equal to K) and the distributed processing is performed according to known order. Communication is required only between neighbouring sensors and between the last and the first sensors. Processing at any arbitrary time t starts from sensor 1 and then sequentially repeated for each subsequent sensors. The sensor K contains the final results of distributed tracking. But as shown in the simulation section, due to the feedback of sensor estimates, the variation in the accuracy of target position estimates is quite small (except between first and remaining sensors). Therefore, final estimates can be obtained from any sensor other than the first one. The posterior distribution at sensor K at time $t - 1$ serves as the prior for the sensor 1 at time t .

Let the state vector at time t is defined as $\mathbf{x}_t = [(\mathbf{x}_t^O)' \mathbf{g}_t']'$, where \mathbf{g}_t and \mathbf{x}_t^O are the sensor locations and the target position respectively. It is assumed that the initial locations of sensors are distributed according to $\mathcal{N}(\mathbf{g}_0, \sigma_0^2)$. Here, \mathbf{g}_0 is the initial sensor locations and σ_0^2 is the associated variance. Then the algorithm is started with particle filtering in the node 1. The particles representing the predictive distribution at the node 1 is given by (Tilde is used to denote the predictive samples):

$$\begin{aligned} \tilde{\mathbf{x}}_t^{1(i)O} &= \mathbf{f}(\mathbf{x}_{t-1}^{K(i)O}, \mathbf{u}_t) \quad i = 1, \dots, N \\ \tilde{\mathbf{g}}_t^{1(i)} &= \mathbf{g}_{t-1}^{1(i)} + \mathbf{v}_t^{(i)}; \end{aligned} \quad (5)$$

where $\mathbf{x}_{t-1}^{K(i)O}$ are the posterior target samples of sensor K at previous time step, used as prior for this node and $\mathbf{v}_t^{(i)} \sim$

$\mathcal{U}[-m_S, m_S]$. The m_S is the maximum speed of sensor (or a constant proportional to the maximum speed).

Next, the update step of the particle filtering is performed at the node. The update step is performed in two steps: one for the target state vector and the second one for the state vector representing the sensor location. In the first update step, the posterior samples at time t is obtained from resampling from particles $(\tilde{\mathbf{x}}_t^{1(i)O})$ $i = 1, \dots, N$. The result of the resampling is the set of particles approximating the distribution of target trajectory at sensor 1 (denoted as $(\mathbf{x}_t^{1(i)O})$ $i = 1, \dots, N$):

$$p(\mathbf{x}_t^{1(i)O} | \mathbf{y}_t^1, \mathbf{Y}_{t-1}) = \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{x}_t^{1(i)O} - \mathbf{x}_t^{1(i)O}) \quad (6)$$

These posteriori particles representing the target trajectory is used to calculate the likelihood weights in the second update step for the sensor state vector. For example, if the sensors are to be positioned such that the distance between a sensor and the target and the distance between two sensors in the transmission link (This reduces the required transmission power and thereby providing a longer battery life) are simultaneously to be minimised, then the corresponding cost function can be evaluated as follows. If $S^1 = E[\mathbf{x}_t^{1(O)}]$ is taken as the target position (calculated from the first resampling step) and $\hat{\mathbf{g}}^2$ is the mean position of the sensor 2 (Sensor 1 only transmits to sensor 2), then the instantaneous cost corresponding to particle i at sensor 1 is given by,

$$\Phi(\mathbf{g}_t^1, \mathbf{x}_t^{1(i)O}, \hat{\Sigma}) = -\alpha(\mathbf{g}_t^1 - S^1)' \hat{\Sigma}^{-1} (\mathbf{g}_t^1 - S^1) - (1 - \alpha)(\mathbf{g}_t^1 - \hat{\mathbf{g}}^2)' \hat{\Sigma}^{-1} (\mathbf{g}_t^1 - \hat{\mathbf{g}}^2). \quad (7)$$

where $\hat{\Sigma}$ is the estimated covariance of the target position and α is a weight which gives importance to individual costs. This instantaneous cost is used as the likelihood weight (after suitable normalisation) in the resampling step for the particles corresponding to the sensor location. This provides the posterior particles, $\mathbf{g}_t^{1(i)}$ according to the likelihood weight given by (7). This completes the processing in the sensor 1.

In the subsequent sensors (i.e., $k = 2, \dots, K$), the priors for the state vector is obtained from the previous sensors ($k = 1, \dots, K - 1$) as shown below:

$$\tilde{\mathbf{x}}_t^{k(i)O} = (\mathbf{x}_t^{k-1(i)O}) \quad (8)$$

$$\tilde{\mathbf{g}}_t^{k(i)} = \mathbf{g}_{t-1}^{k(i)} + \mathbf{v}_t^{(i)}; \quad (9)$$

These particles are used in the update step and it is similar to the one in the sensor 1 for both target and sensor components of the state vector. It is to be noted that the ordering of sensors is arbitrary and the communication links are required only between the adjacent sensors and between the last one and the first one. As the communications of samples representing any distribution is bandwidth-intensive, any low-cost implementation of distributed fusion as proposed in [7], can be used to reduce the huge communication overhead.

The following table explains the complete algorithm for the sensor k :

Table 1. PROPOSED DISTRIBUTED PARTICLE FILTER

-
- Draw $\tilde{\mathbf{x}}_t^{k(i)}$ = $[(\mathbf{x}_t^{k(i)O})' \mathbf{g}_t^{k(i)O}]'$ from the equation (5) if $k = 1$ or else from the equation (8) for $i = 1 : N$.
 - Assign the particle, $(\mathbf{x}_t^{k(i)O})$ a weight according to the section 3 and resample for $i = 1 : N$.
 - Assign the particle, $\mathbf{g}_t^{k(i)}$ a weight according to the (7) and obtain the resampled particles for $i = 1 : N$.
 - Transmit the resampled particles to the sensor 1 if $k = K$ or else transmit to the sensor $k + 1$.
-

5. SIMULATION RESULTS

The proposed algorithm is illustrated with simulations for the bearing-only tracking problem. Sensor manoeuvring is based on the cost criteria (7). Simulations are also conducted for Sequential Monte Carlo based tracking for fixed sensor systems. The manoeuvring system is simulated as explained in [8]. All simulations assume the use of four sensors (K). Initially, these four sensors are located at positions, (20,20), (20,-20), (-20,-20) and (-20, 20). For the fixed sensor case, the sensors are permanently at these locations, denoted as node 1, 2, 3 and 4 respectively. Simulations are repeated for ten different target trajectories for a time duration of 1500 units. The performance criterion used is the average RMSE (Root Mean Square Error). This is calculated at each sensor by averaging the RMSE for the ten target trajectories.

Figure 1 shows the simulation results for the average RMSE at four sensors ($\alpha = 0.9$). As seen here, because of feedback of information from previous sensors, the RMSE performance improves in the second node. But the improvements in subsequent nodes are minimal (Note that RMSE curves for sensors 2, 3 and 4 are closely spaced). Figure 2 shows the simulation results for the average RMSE for both fixed and manoeuvrable cases ($\alpha = 1$). As seen here, in the fixed case, the tracking is lost after the initial period while in the manoeuvrable case, the distributed tracker closely follows the trajectory of the target. The superior performance of the proposed algorithm is further illustrated by considering the value of the RMSE which is very small (less than 2m). It should be noted that the RMSE study explained above is for ten trajectories. Figure 3 shows the tracking performance for one such trajectory. This figure shows the estimated trajectories from manoeuvrable and fixed distributed trackers and the actual trajectories. As seen here, the manoeuvrable tracker closely follows the path of the target while the tracker with the fixed nodes lose the track after the target makes a sudden turn. One reason for the superior estimation performance of the proposed tracker is that as the sensor is very close to the target, the

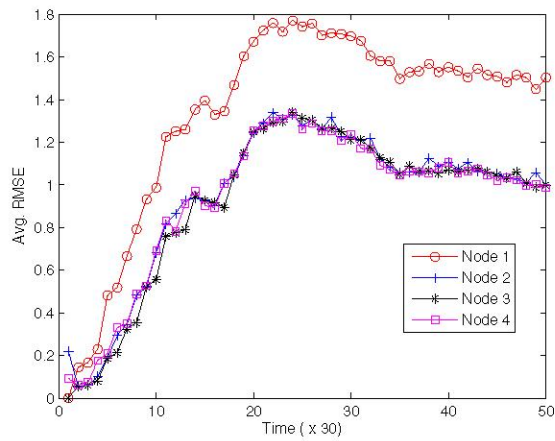


Fig. 1. Average RMSE at four sensor nodes

bearing-only measurements have less uncertainty. The more accurate measurements thus improve the estimation accuracy. Sensor trajectories shown in figure 4 shows that four sensors closely follow the target trajectory as constrained by the cost function.

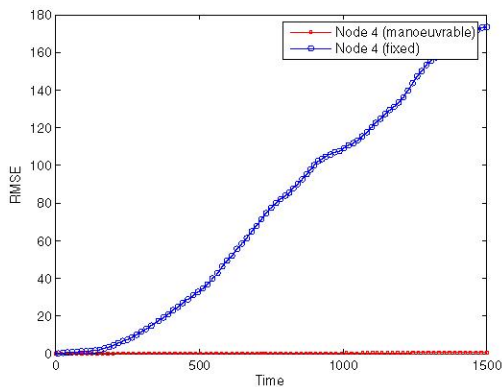


Fig. 2. RMSE at the sensor node 4 for both manoeuvrable and fixed cases

6. REFERENCES

- [1] P. Thomas, M.D. Bull, and G.R. Ravenscroft, "WMD hazard prediction- blending three AI techniques to produce a superior defence capability," in *Proc. 23rd SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, 2003.
- [2] M. Karan, J. Wang, and S.E. Hammel, "Bearings-only tracking with sea trail sonar data from multiple asynchronous sonobuoys," in *Underwater Defence Technology Conference*, 1998.
- [3] A.D. Marrs, "Asynchronous multi-sensor tracking in clutter with uncertain sensor locations using Bayesian sequential

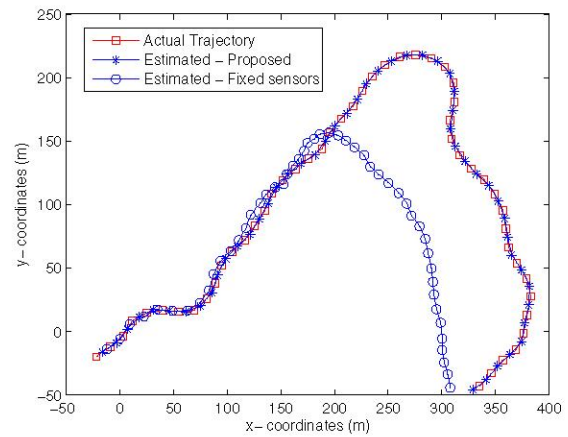


Fig. 3. Actual trajectory and estimated trajectories

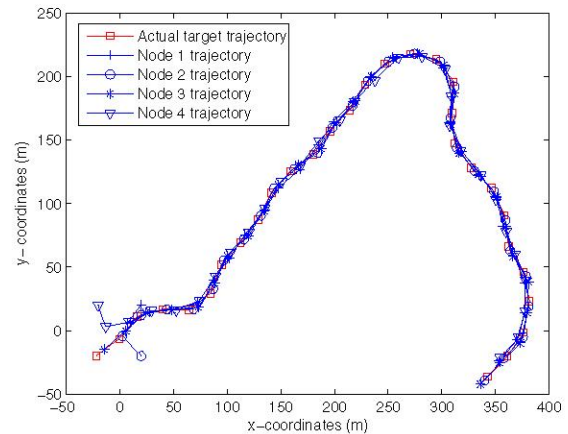


Fig. 4. Sensor trajectories

Monte Carlo methods," in *Proc. of IEEE Aerospace Conference*, Big Sky, MT, 2001, pp. 2171–2178.

- [4] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York, 2001.
- [5] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/ non-Gaussian Bayesian state estimation," *IEE Proc.-F*, vol. 140, no. 2, pp. 107–113, 1993.
- [6] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, March 2002.
- [7] X. Sheng, Y.-H. Hu, and P. Ramanathan, "Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network," in *Fourth International Symposium on Information Processing in Sensor Networks*, 2005, pp. 181–188.
- [8] M. Coates, "Distributed particle filters for sensor networks," in *Int. Symp. Information Processing in Sensor Networks*, Berkeley, California, 2004.