CONTOUR TRACKING OF CONTAMINANT CLOUDS WITH SEQUENTIAL MONTE CARLO METHODS

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

Contour tracking for a single source emission is addressed in this paper. This problem is solved by estimating the contour boundary positions using a set of particle filters. The use of Sequential Monte Carlo techniques enables the tracking to performed when the measurements are noisy and the tracking results also includes the estimation uncertainty. The proposed technique is illustrated for a SCIPUFF generated single emission scenario and simulation experiments showed the successful tracking throughout the tracking period.

Index Terms— Tracking, Distributed estimation

1. INTRODUCTION

Recent international events have clearly demonstrated the need for a fast and accurate tracking and prediction capability for airborne contaminant emissions. Wherever possible, the collection of such data is done without putting personnel in harms way. In the event of airborne contamination, the use of UAVs to locate, detect and track such environmental data removes the operator from contact with potentially hazardous airborne contaminants. Existing contamination prediction systems rely on complex dispersion models which assume limited sensor input. They also suffer from problems of initialisation, especially with multiple sources and are based largely on Gaussian assumptions which break down in complex environments. In this paper, a novel algorithm for tracking airborne contaminants based on a non-linear/non-Gaussian technique is proposed.

Current developments in small air vehicle (SAV), and eventually micro air vehicle (MAV), show that sensing and tracking of airborne contaminants can be carried out distributively using sensor networks mounted on these airborne vehicles. Other military applications of sensor networks range from large-scale acoustic surveillance system for ocean surveillance to small networks of unattended ground sensors for ground target detection [1]. However, the availability of low-cost sensors and communication networks has resulted in the development of many other potential applications. One of such application is in infrastructure security where critical buildings and facilities such as power plants and communication centres have to be protected from potential terror attacks [2]. Environmental and habitat sensing is an another application of sensor networks. Environmental sensors are used to study vegetation response to climate trends and diseases, and acoustic and imaging sensors can identify, track, and measure the population of birds and other species. The System for the Vigilance of the Amazon (SIVAM) [3] provides environmental monitoring, drug trafficking monitoring, and air traffic control for the Amazon Basin. Sensor networks also have been used for vehicle traffic monitoring and control for a quite a while [4].

The proposed algorithm is based on Sequential Monte Carlo methods (Particle filters). This technique has been extensively used in tracking of moving objects. Although tracking of point sources has been clearly demonstrated, the tracking of spatially distributed object has not been widely reported. Gilholm et al. [5] study the problem of tracking of spatially distributed object using Sequential Monte Carlo methods but assumes the knowledge of the distribution of the source. But in airborne contaminant tracking, in general we do not have any such model. To address this problem, rather attempting to obtain a global model of the source, an algorithm based on a set of local particle filters working together to obtain local contours of the airborne contaminant, is proposed.

2. CONTOUR TRACKING ALGORITHM

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 $\{\boldsymbol{x}_t; t \in N\}$ is modelled as a Markov process with initial distribution $p(\boldsymbol{x}_0)$ and transition probability $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$. The observations $\{\boldsymbol{y}_t; t \in N\}$ are assumed to be conditionally independent (in time) given the process $\boldsymbol{x}_t$ and of marginal dis-

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

tribution $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$. We denote by $\boldsymbol{X}_t = \{\boldsymbol{x}_0, \dots \boldsymbol{x}_t\}$ and by $\boldsymbol{Y}_t = \{\boldsymbol{y}_0, \dots \boldsymbol{y}_t\}$, respectively, the system state and the observations up to time t. The measurements $\boldsymbol{y}_t$ are recorded by K sensors, and we use $\boldsymbol{y}_t^k$ to denote the subset of observations made by the k-th sensor. The tracking proposed in this paper is based on Sequential Monte Carlo techniques (also known as particle filters) [6]. The estimated contour points are obtained from the posterior distributions calculated at each sensor nodes.

The tracking is performed by estimating points on the contour boundary using several local particle filters. The sequence of these points generates the contaminant boundary of particular level of concentration. Following subsections explain the initialisation, prediction and update stages of the proposed algorithm and related issues.

2.1. Initialisation

The algorithm starts either with a set of known boundary points or estimated points using the technique described below. This technique works accurately if the boundary falls within the region of exploration. For simplicity, we attempt to estimate a fixed number of number of points (n). Each point is obtained by moving the airborne sensor normal to the contour boundary. In cases, where the initial contour boundary is not known accurately, the initialisation proceeds with the help of points known to fall close to boundary. It is assumed that the number of such points are m ($m << n$). First join these m points and then divide the circumference of the curve to obtain n equi-spaced points. Generate n number of normals associated with each point. The length of the normal is taken as d (depends on the contaminant scenario) and spatial angle is determined by its neighbouring points. By equi-sampling along each of these normals, we can obtain the prior state vector ($\boldsymbol{x_0} = [\boldsymbol{x}_0^1, \cdots, \boldsymbol{x}_0^k, \cdots, \boldsymbol{x}_0^n]$) associated with each n points. The state vector for the sensor, $\boldsymbol{x}_0^k$ is given by the N sample positions along the kth normal.

2.2. Prediction

Once the boundary locations at a previous time instance is known ($\hat{\boldsymbol{x}}_{t-1}^k$), we can calculate the probable regions of the current contaminant boundary. This is achieved by predicting the previous estimated boundary locations by velocity of each boundary points. Although, in contour tracking it is impossible to track a same point over a time span, the 'velocity' estimated using two consecutive times, can provide a initial region for further exploration. Initially, for the purpose of calculation of velocity, it is assumed that each boundary point is at the centre of the curve. If the velocity at point k is denoted as s^k, then the predicted locations are given by:

$$\boldsymbol{x_p}_t^{*k} = \hat{\boldsymbol{x}}_{t-1}^k + c\boldsymbol{s}^k \tag{1}$$

where c is a constant which weighs the velocity vector. Associated with each of the predicted locations, we generate normals at each point. As before, the angle of the normals are determined by two neighbouring points. Equi-spaced points along the normals provides the predicted state vector, $\boldsymbol{x}^{*k}$. Measurements taken at the predicted state vector ($\boldsymbol{y}_t^k$) provides the observations for the update step discussed below.

2.3. Update

Assume that at this stage we have the predicted state vector from the prediction step or initialisation. Starting from 1st sensor ($k = 1$), take measurements at positions denoted by the predicted state vector. Use these measurements in the particle filter update step to obtain boundary locations with higher likelihood. Effectively, these samples (particles) denote the posterior probability of the local contour boundary. These particles are used in the next time step as samples representing the prior probability. Expected values of these probabilities provides the contour locations. But in some cases, the normals may not traverse the contour boundary and this issue is discussed in section 2.4.1. In any case, we should determine whether the posteriori samples corresponds to the situation where the normals are intersecting the contour boundary. To check this we use the following simple criterion. We define a direction indicator, $s_d^i = \mathrm{sign}(y_t^i - y_{threshold})$ ($y_{threshold}$ is the contaminant level to be tracked and y_t^i is the measurement at position denoted by the ith element of $\boldsymbol{x}^{*k}$). This indicates on which side of contour boundary location, the measuring sensor is. A positive direction indicator denotes that the measurement point is inside the contour boundary and vice versa. Combining this with the particle filter weight, we can define the following the modified weight for each point in the the state vector,

$$w_{mod}^i = s_d^i w^i \quad i = 1, \cdots, N \tag{2}$$

If this modified weight is completely within the contour, the value w_{mod}^i equals to 1 and if it is completely outside the boundary, it is equal to -1. If this weight is $0 < w_{mod}^i < 1$ then the normals intersect the boundary and we should make sure that each of normals satisfy this condition. If this condition is not satisfied, we should extend the region of exploration as explained in the following subsection.

Thus, contour locations, $\hat{\boldsymbol{x}}^k$ at time t is given by:

$$\hat{\boldsymbol{x}}_t^k = E(p(\boldsymbol{x}_t^k)|\mathbf{Y}_t^k) \tag{3}$$

The curve made up from $\hat{\boldsymbol{x}}_t^k$ $k = 1, \cdots, n$ determine the estimated contour boundary.

2.4. Implementation Issues

To apply the above the algorithm successfully in different scenarios, a number of application related issues need to addressed. For example, in some cases, the normals generated may not traverse the contour and hence need to extended to the region where the contour boundary is. Another issue is that in some cases, the normals or extended normals may intersect each other. To obtain a clear unambiguous curve, we should address these problems.

2.4.1. *Extending the region of exploration*

From the particle filter weight, w, we obtain the sample position which gives the maximum likelihood and generate a new set of normals around these sample positions. Either we can have the normal length to be d or depending on the variance of unnormalised particle weight, we can increase the length of normals. A very low variance indicates that all samples are quite similar and therefore it is difficult to find the direction where the likelihood improves. By using a longer length ($2d$) for normals in this situation, we can quickly reach the contour boundary.

3. SIMULATION RESULTS

In this section, simulation results obtained with our proposed algorithm are presented. The vapour contamination data is simulated using the SCIPUFF software [7]. The algorithm written in MATLAB uses these data as measurements for tracking. As the tracking is based on a set of particle filters, the measurements at each sample locations are obtained as required (no off-line processing is required). Proposed algorithm processes these measurements after adding noise with a variance equal to 0.002. A simple scenario of single emission is considered with just one cloud during the period of tracking. Aim of the tracking is to estimate the contours points with a level of $-12dB$ ($y_{threshold} = -12dB$).

Each particle filter uses $N = 50$ number of samples (they correspond to N number of locations along the line of exploration). Initially, sensors are located at positions, $(-85.21, 36.06)$,$(-85.175, 36.01)$, $(-85.2, 35.97)$, $(-85.25, 35.96)$, $(-85.3, 35.98)$, $(-85.32, 36.04)$ and $(-85.27, 36.07)$. The number of points on the contour to be estimated is $n = 35$ (We can employ either n number of sensors or use fewer sensors to estimate the n points sequentially). Emission is assumed to occur at time zero and the tracking is carried out at time steps of 30 minutes for a duration of 15 hours.

Figure 1 shows the initial state vectors along normals at initial positions (Initial known location are marked with a circle). Figures in diagram 2 illustrates the successful tracking of contaminant throughout the simulation period. A Monte Carlo simulation with ten different random seeds showed that the proposed algorithm is robust to different simulation conditions and the algorithm was able to track the contaminant at all iterations. Estimation error is quantified using two quantities: Root Mean Square Error (RMSE) and Kullback-Liebler (KL) distance. The RMSE (RMSE in this paper is calculated by considering n estimation errors at a particular time) quan-

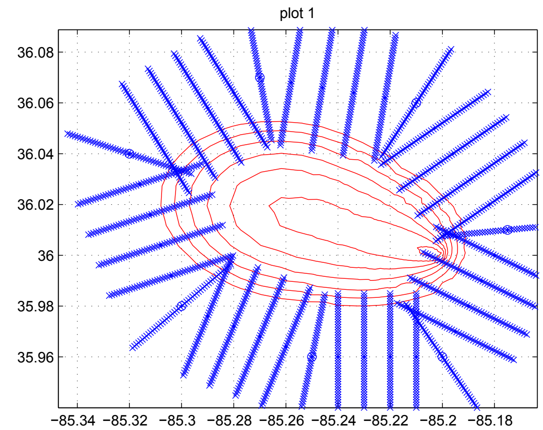


Fig. 1. Initial state vector positions

tifies how far our estimated contour positions varies from the actual contour in terms of spatial distance. But this fails to take into consideration the variation in contaminant intensity. A small value of RMSE does not indicate that contaminant is safe but it tells that that estimated contour is close in terms of distance to the actual one. The approximation of the estimated contaminant level to the actual contaminant level can be calculated using the KL distance.

Figure 3 shows the RMSE and KL distance for the proposed algorithm. As can be seen from figure 3a, although the RMSE slightly increases with time, it is low throughout the tracking duration. The increase of RMSE is caused by the rapid dispersal of contaminant with time. The KL distance shown for five different runs in figure 3b shows that the KL distance is very close to zero. A KL value of zero means that the both estimated and actual levels coincide. A jump in KL distance at time step 25 is caused by one of n points being slightly away the contour boundary (At this point, the different contour levels are very closely spaced). As the operating SNR (Signal to Noise Ratio) varies very widely, this phenomenon (i.e., few points losing track) is expected. This study shows that our proposed algorithm successfully track the airborne contaminant for the simple case presented. Tracking for complex environments where contaminant clouds can merge or split is currently under study.

4. CONCLUSIONS

A novel algorithm for tracking airborne contaminant was proposed and simulations with SCIPUFF was used to illustrate the performance of the proposed algorithm. The proposed algorithm based on Sequential Monte Carlo methods estimates the local points of contour boundary and can be operated in a noisy measuring environment. Performance metrics of KL distance and RMSE were used to assess the performance of

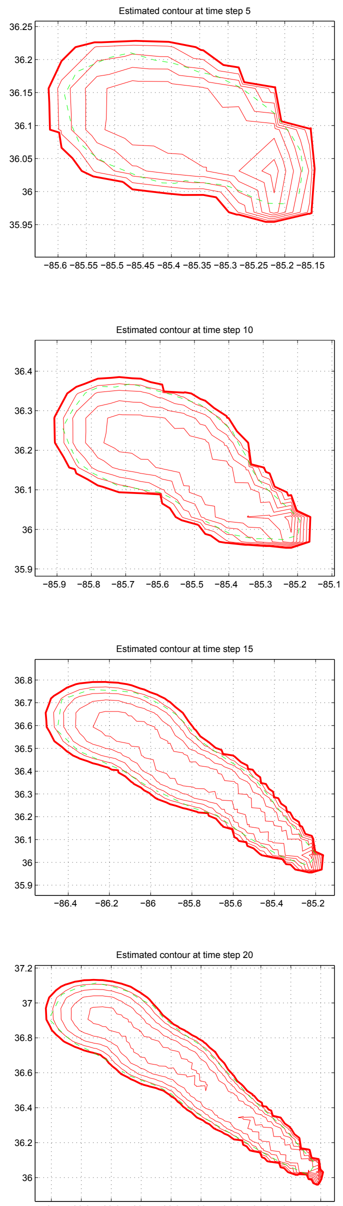


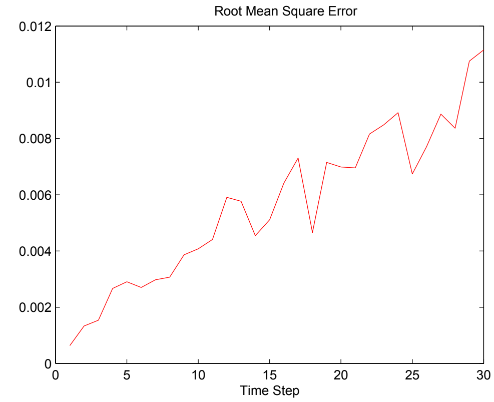
Fig. 2. Contour tracking results at time steps of 5, 10, 15 and 20

the algorithm and these performance measures showed that the tracking was successful throughout the emission period.

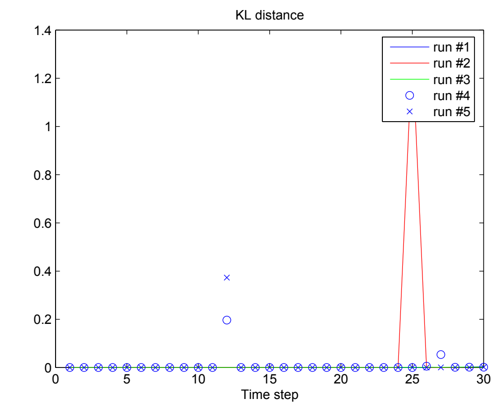
5. REFERENCES

[1] C.-Y. Chong and S.P. Kumar, "Sensor networks: evolution, opportunities and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, August 2003.

[2] R. Hills, "Sensing for danger," [Online] Available: http://www.llnl.gov/str/JulAug01/Hills.html, July/Aug. 2001.



(a) RMSE



(b) KL distance

Fig. 3. RMSE and KL distance

[3] D. Jensen, "Communication, navigation and surveillance for the amazon," [Online] Available:http://www.aviationtoday.com/reports/avionics/previous /0602/06002sivam.htm, 2002.

[4] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proc. Int. Conf. Mobile Computing and Networking (MOBICOM)*, 1999, pp. 263–270.

[5] K. Gilholm and D. Salmond, "Spatial distribution model for tracking extended objects," *IEE Proceedings- Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 364–371, October 2005.

[6] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York, 2001.

[7] R. I. Sykes, S. F. Parker, D. S. Henn, C. P. Cerasoli, and L. P. Santos, "PC-SCIPUFF version 1.1PD.1 Technical Documentation. ARAP report no. 718," Tech. Rep., Titan Corporation, Titan Research & Technology Division, ARAP Group, P.O. Box 2229, Princeton, NJ, 1998.