# Estimation of Pollutant-Emitting Point-Sources using Resource-Constrained Sensor Networks

Michael Zoumboulakis and George Roussos

School of Computer Science and Information Systems
Birkbeck College, University of London
Malet Street, London WC1E 7HX, UK
{mz,gr}@dcs.bbk.ac.uk

**Abstract.** We present an algorithm that makes an appropriate use of a Kalman filter combined with a geometric computation with respect to the localisation of a pollutant-emitting point source. Assuming resource-constrained inexpensive nodes and no specific placement distance to the source, our approach has been shown to perform well in estimating the coordinates and intensity of a source. Using local gossip to directionally propagate estimates, our algorithm initiates a real-time exchange of information that has as an ultimate goal to lead a packet from a node that initially sensed the event to a destination that is as close to the source as possible. The coordinates and intensity measurement of the destination comprise the final estimate. In this paper, we assert that this low-overhead coarse localisation method can rival more sophisticated and computationally-hungry solutions to the source estimation problem.

**Key words:** Source Localisation, Spatial Event Detection, Estimation, Kalman Filters.

## 1 Introduction

We live in days of elevated risk that terrorists could acquire Chemical, Biological and Radiological (CBR) weapons to attack major cities. In a world of easy availability of CBR raw materials in hospitals of failed states, the greatest concern for citizens of urban centres is not over an attack by a nuclear warhead but with a "dirty-bomb" that would contaminate a wide area, trigger widespread panic and cause severe disruption [12]. Government agencies, such as the Home Office in the UK, are involved in "resilience" programmes, set to cost GBP 3.5bn per year by 2011, with the aim to *"ensure that, in the event of a terrorist incident the response from all concerned will be quick and effective, so that lives can be saved and the impact on property and the environment minimised"* [15].

The principal concern of such programmes is the detailed response strategy in the case of a CBR attack. For this, accurate assessment and modelling of hazards [5] are essential. Modelling can be split to the following two subproblems: forward and inverse. While the forward problem involves the dispersion prediction and hazard assessment given a known location of a point source emitting pollutants in space, the inverse problem involves locating this source given some measurements of the pollutants in the atmosphere. The quick localisation of an attack source can assist the specialist personnel in neutralising the threat as well as predicting the dispersion cloud and evacuating citizens away from it.

In this paper, we consider a scenario where a number of low-cost, resource-constrained sensor nodes, such as the TMote Sky [3], are deployed in an urban area with the task of detecting the presence of certain pollutants in the atmosphere. There exist real-world deployments, such as the U.S. Federal SensorNet's [16] testbeds at Washington DC and New York City, that aim to serve the aforementioned objective. Following the detection of the pollutant — a problem addressed by our earlier work on event detection [23] — the goal is to compute an estimate of the source's location and intensity.

We use an iterative in-network approach that does not rely on powerful nodes or network-wide collection and offline processing. Our method involves a pre-determined number of *initiating* nodes that independently sense the spatial event. Each of these nodes, begins a procedure that aims to route a packet intelligently towards the source of the spatial event. The ultimate aim is that after a small number of hops the packet should be at a node very close to the source of the spatial event — we will call this the *destination* node. By taking the location coordinates and measurement at the destination node, we have a coarse estimate of the source's location and intensity.

The method is designed to operate as close to real-time as possible and to be lightweight in terms of network communication. To address the latter point, we use directional local gossip to propagate event state information. This is somewhat akin to the Trickle [10] algorithm that uses polite gossip to bring network consistency with respect to a set of global shared parameters. Our algorithm starts with the initiating node making a guess of the spatial event state at single-hop neighbouring nodes and sending it to the local broadcast address. Nodes hearing the broadcast reply with their own measurements. The estimation error is calculated and the node that minimises the error is selected as the next hop and its measurement is used to correct the initial prediction made. This "predict-correct" cycle is in fact a straightforward Kalman filter. The node selected as next hop receives the filter parameters and repeats the steps. The procedure is continued iteratively, recording the network path along the way and it exits when a node runs out of unvisited neighbours that are likely to lead any closer to the source. The exact details of the algorithm will be described in section 2.2.

In the remainder of this paper, we will formalise the problem and we will discuss the requirements for a solution. We will then present our algorithm for decentralised in-network point-source estimation together with discussion of a test case and experimental simulation results. We will conclude by reviewing selected related work together with future plans.

### 1.1 Problem Statement & Requirements

We consider a single static point source that emits a pollutant of chemical or radiological nature. This source is located at the unknown coordinates $(x_s, y_s)$ in the two dimensional coordinate system. The presence and intensity of the pollutant in the atmosphere is sensed by $N$ sensor nodes located at $(x_i, y_i)$ coordinates with $i = 1, 2, \ldots, N$. The goal is to devise a computational method that estimates the coordinates $(\hat{x_s}, \hat{y_s})$ and intensity of the source, within some error $\epsilon$.

Formally, given $z_i$ sensor measurements collected at coordinates $(x_i, y_i)$ the goal is to estimate the vector $\hat{\boldsymbol{v}}$ :

$$\begin{bmatrix} \hat{x_s} & \hat{y_s} & \hat{I} \end{bmatrix}^T$$

where $\hat{I}$ is the intensity estimate as it would be measured 1 meter from the source [5].

The complexity of a proposed solution for this problem varies significantly depending on the assumptions made. In our case we assume a single static point source in $\mathbb{R}^2$ and a steady-state dispersion model. The latter point refers to a simplified gas concentration model, adapted from [7], where measurements are time-averaged and constant with respect to time. Furthermore, we assume that the nodes are aware of their own location coordinates.

The solution to the source localisation problem needs to be lightweight both in terms of computation and communication. It needs to operate in a decentralised manner without relying upon powerful nodes or offline processing and it needs to be capable of converging to an estimate rapidly — typically under two minutes.

Our approach tolerates both non-uniform distributions and, due to the fact that our estimation algorithm employs a Kalman Filter, a degree of measurement and process noise.

## 2 Iterative Source Location Estimation

### 2.1 The Kalman Filter

The Kalman Filter is an optimal, in the least squares sense, estimator of the true state of a dynamic linear system that its measurements are corrupted by white uncorrelated noise. In our context the Kalman filter is used to estimate the vector $\hat{\boldsymbol{v}}$. In its simplest form, a Kalman filter, is based on the following five equations:

$$\hat{x}_k^{\,-} = A\hat{x}_{k-1} + w_{k-1} \tag{1}$$

$$P_k^{\,-} = AP_{k-1}A^T + Q \tag{2}$$

$$K_k = P_k^{-}H^T(HP_k^{-}H^T + R)^{-1} \quad (3)$$

$$\hat{x}_k = \hat{x}_k^{-} + K_k(z_k - H\hat{x}_k^{-}) \quad (4)$$

$$P_k = (I - K_kH)P_k^{-} \quad (5)$$

Where $\hat{x}_k^{-}$ is the a priori state estimate, $\hat{x}_k$ is the a posteriori state estimate, $A$ is the state transition matrix, $w$ is the white, zero-mean, uncorrelated noise, $P_k^{-}$ is the a priori error covariance, $P_k$ is the a posteriori error covariance, $Q$ is the process error covariance, $R$ is the measurement noise covariance, $H$ is the measurement matrix, $z_k$ is the measurement taken at time $k$ and $K$ is the Kalman Gain. Equations 1 and 2 are the *Time Update* (Predict) equations. Equations 3 to 5 are the *Measurement Update* (Update or Correct) equations.

The goal of the Kalman filter is to formulate an *a posteriori* estimate $\hat{x}_k$ as a linear combination of an *a priori* estimate $\hat{x}_k^{-}$ and a weighted difference between an actual measurement $z_k$ and a measurement prediction $H\hat{x}_k^{-}$ as shown in equation 4 [20].

Due to space limitations we will not discuss the particulars of the Kalman Filter in more detail; the interested reader can refer to [17], [20], [22] for an extensive review.

### 2.2 In-network Estimation

Our localisation algorithm makes appropriate use of a Kalman filter on the basis of the assumptions of section 1.1. The process starts at an individual node — this can be a choice from a pre-determined set of nodes that sense the event independently. The desired outcome is to start a "walk" of the sensor field by visiting (i.e. sending a packet to) other nodes.

Once the event is sensed, the initiating node makes an initial guess of the measurement at adjacent single-hop nodes. Since no other information is available this guess equals to a linear transformation of the local measurement. The node then tasks the one-hop unvisited neighbours to send their measurements. This is achieved by a local broadcast. We will see later how the local broadcast is avoided once nodes reach a consensus regarding the quadrant direction of the source. Once the replies with the measurements have been collected the innovations $Z_n = (z_k^{(i)} - H\hat{x}_k^{-})$ are calculated. Note that $\hat{x}^{-}$ is the initial estimate made by the initiating node and measurement $z_k^{(i)}$ has now a superscript $i$ to indicate which neighbouring node has reported it. Then the minimum innovation $\underset{z}{\operatorname{argmin}}(z_k^{(i)} - H\hat{x}_k^{-})$ is calculated and node $i$ is selected as the next hop.

When node selection takes place, the selector sends to the selectee a collection of filter parameters so the latter can continue the process. This allows each node in the network to run a lightweight application and only when necessary to be tasked to perform the estimation. We will refer to this collection of parameters as *particle*, since it resembles a travelling particle that facilitates a task.

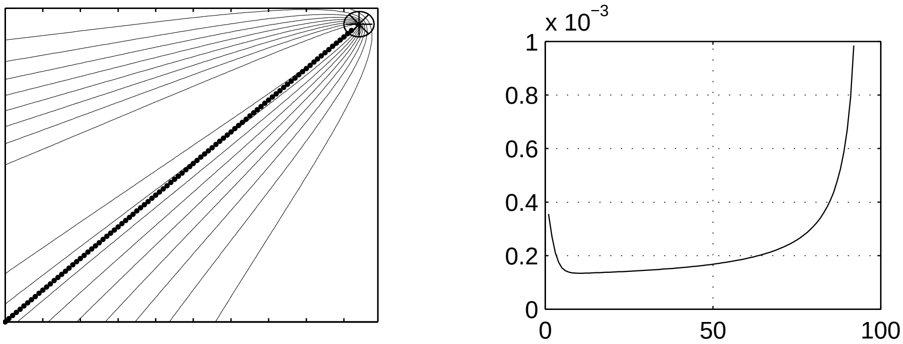
**Algorithm 1** Particle Localisation Algorithm

1: **variables** Estimate Error Covariance $P$, Measurement Noise Variance $R$, Process Variance $Q$, State Transition Matrix $A$, Measurement Matrix $H$, Initial Estimate $\hat{x}_k{}^-$, maxhopcount=1, netpath[], counter $c = 0$;
2: Project state estimate $\hat{x}_k{}^-$ ahead (Eq. 1).
3: Project error covariance $P_k{}^-$ ahead (Eq. 2).
4: Task *unvisited* neighbours within maxhopcount to report measurement.
5: **for** (each of replies received) **do**
6:     calculate innovations $(z_k{}^{(i)} - H\hat{x}_k{}^-)$
7: **end for**
8: Select as next hop the node that minimises the innovation.
9: Compute the Kalman gain (Eq. 3).
10: Correct (Update) estimate with measurement $z_k{}^{(i)}$ (Eq. 4).
11: Correct (Update) the error covariance $P_k$ (Eq. 5).
12: Compute relative error.
13: **if** abs(relative error) >=multiple$\cdot(\mathbb{E}[Rel\ Error])$ **then**
14:     **exit**
15: **else**
16:     Add local address to netpath[c] and increment c.
17:     Send particle to selected node (line 8) and task it to start at Line 1.
18: **end if**

The typical stopping condition is linked to the estimation relative error (line 13); for instance, when the estimation relative error exceeds a multiple of the mean relative error the process halts. A sharp rise in the relative error usually reveals that the particle has moved outside the plume or to an area where measurements differ significantly given the initial estimates.

Moreover, a variation of the algorithm has been developed that when the relative estimation error becomes high then a selector node considers candidate next-hop nodes that are more than one hop away. The same local broadcast is issued but the maximum hopcount is increased to a pre-determined value i.e. to <= 3. This effectively means that the selector will receive more measurement replies — to be precise it will receive $(2h+1)^2-1$ where $h$ is the number of hops. Since there are more measurements available it is more likely to find a candidate for next hop that will keep the error low.

The typical use of this localisation algorithm is to employ many particles for robustness. With $n$ particles we add a geometric computation — not shown in the above listing for the sake of simplicity — to establish a *consensus* regarding the quadrant in which the source is located. This operates as follows: after a small number of hops (i.e. $\leq 10$), a convex hull is evaluated for the particles' coordinates. Recall that the convex hull is the boundary of the minimal convex set containing a finite set of points (i.e. the $N$ particle coordinates). Provided that particles commence at nearby locations (i.e. $\leq 4$ hops away from each other) the convex hull can be evaluated without a significant communication overhead. The mean direction of movement is given by calculating the centroid of the convex hull (Figure 3(a)). Only a coarse *quadrant direction* is necessary




(a) Particle path (b) Relative Estimate Error

**Fig. 1.** An example of the particle path (a) shown by the darker line. Starting at $(1, 1)$ it moves towards the source at $(95, 95)$. It reaches a distance of 2.83 away after 93 hops. On (b) the estimation error is shown. While, initially it is reducing as the particle approaches the source it increases gradually and then sharply — the sharp rise indicates that the measurements are not in line with expectations and it is the stopping condition.

at this point. This is the quadrant in which $N$ particles estimate the source location and it constitutes the consensus. Once the consensus becomes known, there is no need for local broadcasts. Instead, the estimates are sent directly to the neighbours in the consensus direction. This achieves an improvement both in terms of communication — $\frac{3}{4}$ less messages at each hop — and estimate accuracy.

## 3 Evaluation

Given that empirical evaluation using realistic conditions is problematic due to the hazardous application scenarios, we have evaluated the correctness of the algorithm via simulations in MATLAB. We have considered a specific test case of a time-averaged gas plume over a 100-by-100 square grid according to the model described in [7] and the assumptions of section 1.1. The performance criteria used were: (a). the Euclidean distance to the true source and (b). the length of the path taken from the start to the end point that comprised the final estimate. The former criterion is associated with estimation accuracy while the latter is a measure of speed, communication cost and efficiency. Moreover we have employed a naive algorithm that selects the maximum reading as the next hop as a baseline benchmark.

Figure 1 exhibits the simplest case involving a single particle starting at an arbitrary location and iteratively moving towards the source region of the spatial event until it reaches at $(93, 93)$, a distance of 2.83 away from the true source — a useful estimate of the true coordinates and an intensity estimate within $\epsilon = .0121$ of the true intensity. Taking into account that network-wide broadcasts were not

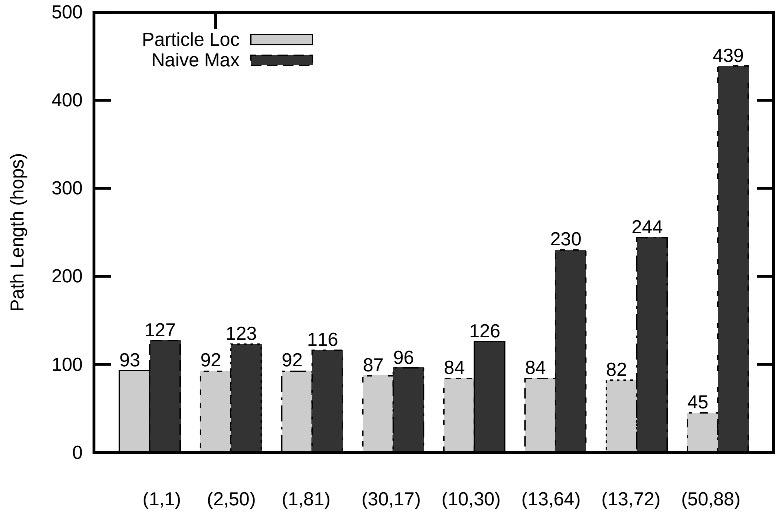
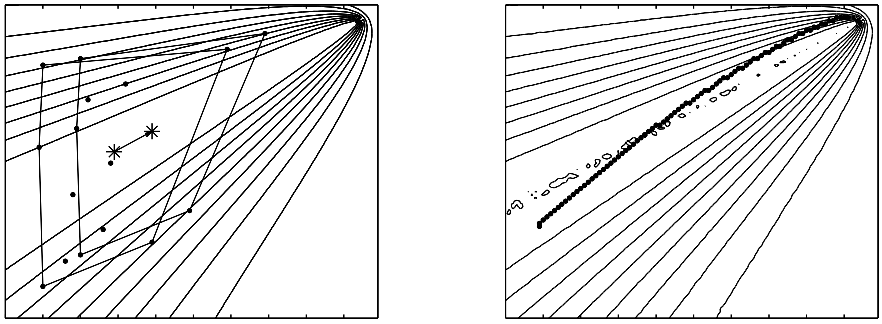



**Fig. 2.** Comparison of our particle localisation algorithm with a maximum selection naive algorithm. The $x$-axis is labelled with starting coordinates. All paths ended at a distance less than 4 (Euclidean) to the real source, but our approach resulted in shorter (in the case of $(50, 88)$, up to a factor of 9) paths.

required and only local communication between a very small percentage of the nodes was involved, we assert that the method is lightweight and respects the constraints of the devices. In addition, although the starting coordinates can be any point in two-dimensional space, for the sake of demonstrating one of the strengths of the algorithm in this discussion we will assume starting points located fairly far from the source.

Figure 2 shows the benchmark comparison to the naive algorithm; this small set of random starting points suggests correct and efficient behaviour. The fact that all paths taken by the 8 particles achieved a distance less than 4 to the true source in less than 100 hops seems to suggest quick convergence to appropriate estimates. Furthermore, in various test runs our algorithm did not seem to be affected by the local maxima that caused the naive maximum selection algorithm to move in circular fashion and result in long paths.

Figure 3(a)) shows the typical operation of the algorithm with $n$ particles and the geometric addition that computes the consensus quadrant direction of movement shown by the arrow. Using the consensus, individual particles only send estimates towards that quadrant direction reducing communication and achieving more robust estimates. By adding this geometric aspect to the stochastic estimation nature of the algorithm we factor a level of error tolerance in the approach: as long as the majority of the particles move in the right direction, we guarantee that energy will not be wasted in erroneous propagation decisions that can not possibly lead to an accurate estimate.



(a) Mean Direction of Movement

(b) Corruption of measurements by Noise

**Fig. 3.** (a) Two convex hulls evaluated for 8 particles after 10 and 20 hops respectively. Marked with an asterisk are the centroids of the polygons (consensus). (b) Particle starting at $(10, 30)$ and heading towards the source — measurements are corrupted by random noise.

Lastly, our approach shows tolerance to small measurement noise. This was tested by corrupting the gas concentration measurements by random noise of small magnitude (i.e. $N(0, .0001)$) (Figure 3(b)) — this had no impact to the estimation performance of the filter. However, increasing the noise magnitude results in inconsistent behaviour and requires adaptation of the filter parameters which we will not discuss here any further due to space limitations.

## 4 Related Work

Coarse grained localisation techniques can be as simple as the *Centroid Calculation* or Point-in-Triangle (PIT) methods [9]. A refinement of the PIT technique is the Approximate Point in Triangle (APIT) described in [6]. A geometric approach based on the circles of Apollonius is described in [4]. A robust to noise and measurement errors data fusion algorithm that extends the latter approach is presented in [2].

A different family of methods is based on the well-understood Time Difference of Arrival (TDoA) localisation. There is a geometric and a numerical solution to this problem — more information on the details of TDoA solutions can be found in [11], [13], and [21].

Another solution is based on Maximum Likelihood Estimation (MLE): the least squares minimisation of the estimate of vector $\hat{\boldsymbol{v}}$ can be solved by gradient descent or numerically according to the methodologies of [5] and [14].

Lastly, the Kalman filter has been employed in radioactive source localisation in [5]. The difference with our approach is that we perform the estimation in-the-network while [5] assumes offline processing. A combination of a Kalman filter with Time of Arrival (ToA) is presented in [8] while a distributed Kalman filter for wireless sensor networks is presented in [18].

## 5 Future Work & Discussion

To validate the correctness of the simulation results, at the time of this writing, we are in the process of implementing the algorithm in a network-specific simulator [1]. This simulator accepts TinyOS code and allows evaluating the impact of many real-world network-specific parameters such as density, distribution and so on.

Concluding, we have presented an iterative point-source coarse localisation algorithm that operates in-the-network and does not require powerful nodes or network-wide collection and offline processing. A straightforward Kalman filter is at the heart of the algorithm that iteratively computes the source estimate. Using this approach has the following advantages:

- *Efficient paths* in terms of length and distance to the source, when compared to a naive maximum selection algorithm.
- *Robustness* which is introduced by using multiple particles and a geometric approach to establish mean direction of movement. This direction consensus facilitates both reduction in communication and improved accuracy of estimates.
- *Lightweight*, real-time properties that neither involve the entire network nor make any assumptions about individual placement of nodes.
- *Noise resilience* due to intrinsic characteristics of the Kalman filter make our approach robust to errors introduced by the inexpensive circuitry and sensory equipment.

Finally, the recursive nature of the Kalman filter makes it a good match for operation in resource-constrained devices — approaches such as [19] use optimised implementations — and a valid generic solution to the CBR source localisation problem.

## References

1. Avrora - The AVR Simulation and Analysis Framework. `http://compilers.cs.ucla.edu/avrora`.
2. Jren-Chit Chin, David K. Y. Yau, Nageswara S. V. Rao, Yong Yang, Chris Y. T. Ma, and Mallikarjun Shankar. Accurate localization of low-level radioactive source under noise and measurement errors. In *SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 183–196, New York, NY, USA, (2008).
3. Sentilla Corp. TMote Sky Datasheet. `http://www.sentilla.com/pdf/eol/tmote-sky-datasheet.pdf`.
4. Joseph Cox and Michael Partensky. Spatial Localization Problem and the Circle of Apollonius, January (2007).
5. Ajith Gunatilaka, Branko Ristic, and Ralph Gailis. Radiological Source Localisation (DSTO-TR-1988), (1988).
6. Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek Abdelzaher. Range-free localization schemes for large scale sensor networks. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 81–95, New York, NY, USA, (2003).

7. Hiroshi Ishida, Takamichi Nakamoto, and Toyosaka Moriizumi. Remote sensing of gas/odor source location and concentration distribution using mobile system. *Solid State Sensors and Actuators*, 1(16):559–562, (1997).
8. Ulrich Klee, Tobias Gehrig, and John McDonough. Kalman filters for time delay of arrival-based source localization. *EURASIP J. Appl. Signal Process.*, (1):167–167, (2006).
9. Bhaskar Krishnamachari. *Networking Wireless Sensors*. Cambridge University Press, 2005.
10. Philip Levis, Neil Patel, David Culler, and Scott Shenker. Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *NSDI'04: Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation*, Berkeley, CA, USA, (2004).
11. G. Mellen, M. Pachter, and J. Raquet. Closed-form solution for determining emitter location using time difference of arrival measurements. *IEEE Transactions on Aerospace and Electronic Systems*, 39(3):1056–1058, (2003).
12. BBC News Online. Threat of dirty bombs 'increased' . `http://news.bbc.co.uk/1/hi/uk/7960466.stm`, March (2009).
13. Nageswara Rao. Identification of simple product-form plumes using networks of sensors with random errors. In *9th International Conference on Information Fusion*, pages 1–8, (2006).
14. Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. Dynamic fine-grained localization in Ad-Hoc networks of sensors. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, New York, NY, USA, (2001).
15. Home Office Office for Security and Counter Terrorism. Chemical, Biological, Radiological, Nuclear Resilience. `http://security.homeoffice.gov.uk/cbrn-resillience/`.
16. SensorNet. Nationwide Detection and Assessment of Chemical, Biological, Nuclear and Explosive (CBRNE) Threats. `http://www.sensornet.gov/sn_overview.html`.
17. Dan Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. WileyBlackwell, (2006).
18. Demetri P. Spanos, Reza Olfati-Saber, and Richard M. Murray. Approximate distributed Kalman filtering in sensor networks with quantifiable performance. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 18, Piscataway, NJ, USA, (2005).
19. Jimfron Tan and Nicholas Kyfuakopoulos. Implementation of a Tracking Kalman Filter on a Digital Signal Processor. *IEEE Transactions on Industrial Electronics*, 35(1):126–134, (1988).
20. Greg Welch and Gary Bishop. An Introduction to the Kalman Filter. Technical Report 95-041, Chapel Hill, NC, USA, (1995).
21. Xiaochun Xu, Nageswara Rao, and Sartaj Sahni. A computational geometry method for DTOA triangulation. In *10th International Conference on Information Fusion*, pages 1–7, (2007).
22. Paul Zarchan. *Fundamentals of Kalman Filtering: A Practical Approach*. Progress in Astronautics & Aeronautics Series. AIAA, 2 edition, (2005).
23. Michael Zoumboulakis and George Roussos. Escalation: Complex event detection in wireless sensor networks. In *EuroSSC*, volume 4793, pages 270–285, (2007).