# Source localization of Contaminants using Non-linear Approximation Filtering

Channa Navaratna, Ph.D. †
Menaka Navaratna, Ph.D. ‡

**Abstract**

Advancements in chemical and biological agents allow human life to be elevated to a whole new higher level, yet in wrong hands their use had produced devastating results in past. Unarguably it is vital to develop methods to track and trace the origin of hazardous contaminant particles in order to minimize the damages they are capable of causing, and contain their ill effects. Here we consider the problem of devising filtering methods in order to estimate the origin of contaminant particles using information gathered from a finite number of sensors



## Introduction

The Kalman filter is the optimal solution to the Bayesian estimation problem for a given linear, stochastic, state-space system with additive Gaussian noise. Closed form solution have been derived for the aforementioned problem and have been extremely popular in the past [2], [3]. However, if the actual dynamical system drifts from a linear dynamical system or assumptions on characteristics of noise are incorrect, the filter tends to diverge. A variety of algorithmic modifications were invented in an attempt to compensate for the model errors that caused the misbehavior of the filter. This issue had been addressed to some extent by *Extended Kalman Filter* where an approximated linear system is derived for every calculation step. The practicality of this approach has limited application to large complicated dynamical systems. Other techniques such as Uncented Kalman filter, Gaussian sum filter also suffer from the similar shortcomings as they either directly or indirectly use Kalman estimation techniques.

In recent years, computational power has reached to an extraordinary peak that one can implement algorithms once discarded due to their extensive computational cost. One type of powerful algorithm that resurfaced in recent years is the particle filter (PF) algorithm (Sequential Monte-Carlo algorithms) [6]. Recently particle based sampling filters have been proposed and used successfully to recursively update the posterior distribution of $p\left(x_k / \{y_1 \ldots y_k\}\right)$ using *sequential importance sampling and resampling*. In contrast to Kalman filters particle filters in general can be used with non-linear, non-Gaussian dynamical systems. However, it needs to use a large amount of samples (particles) for a robust operation and accurate estimation which in many cases can be computationally expensive.

In our problem, we need to estimate the initial state $x_0$ by processing the observations $\{y_1 \ldots y_k\}$ available up to $t = k$. This class of problems is tends to be more difficult and computationally more expensive. The beauty of particle filters is that it provides a solution to the localization problem and it exhibits excellent results.

Since particle filters inherently computationally expensive modification to the algorithm as well as to implementation have been done to dramatically reduce computational time. It is shown in this paper that convergence of the solution well within practical limits of containment or evacuation.



**Particle Filtering**

Consider the following discrete time non-linear system

$$x_{k+1} = f(x_k) + \omega_k$$
$$y_k = g(x_k) + \theta_k$$

where $x_k \in R^n, y_k \in R^d$ and $\omega_k, \theta_k$ are independent noise processes of appropriate dimensions. It is assumed that the initial distribution $x_0$ is independent of $\omega_k$ and $\theta_k$. Mean and variance of $\omega_k, \theta_k$ are assumed to be known. Here we consider the Markovian state space models where state of the system $x_k$ depend only on the previous state $x_{k-1}$ in a probabilistic sense.

It is assumed that the probability distribution of $x_0$ is $p(x_0)$ and the distribution for the transition is $p(x_k \mid x_{k-1})$. It is also assumed that the conditional distribution of the outputs is $p(y_k \mid x_k)$.

The particle filter is to estimate the distribution $p(x_k \mid Y_k)$ using posterior probability distribution $p(Y_k \mid X_k)$ with $X_k = \{x_0, x_1, \ldots, x_k\}$ and $Y_k = \{y_1, y_2, \ldots, y_k\}$. Then it allows us to calculate any optimal estimate of the state, such as the conditional mean

$$\hat{x} = E[x_k \mid Y_k] = \int x_k \, p(x_k \mid Y_k) \, dx_k$$

Bayes' rule can be used to rearrange the posterior distribution,

$$p(X_k \mid Y_k) = \frac{p(Y_k \mid X_k) \, p(X_k)}{\int p(Y_k \mid X_k) \, p(X_k) \, dX_k}$$

A recursive formula for the aforementioned can be obtained as follows [6]:

$$p\left(X_{k+1} \mid Y_{k+1}\right) = p\left(X_k \mid Y_k\right) \frac{p\left(y_{k+1} \mid x_{k+1}\right) p\left(x_{k+1} \mid x_k\right)}{p\left(y_{k+1} \mid Y_k\right)}$$ Marginal distribution of $p(x_k \mid Y_k)$ can be calculated as follows:

$$p\left(x_k \mid Y_k\right) = \frac{p\left(y_k \mid x_k\right) p\left(x_k \mid Y_{k-1}\right)}{\int p\left(y_k \mid x_k\right) p\left(x_k \mid Y_{k-1}\right) dx_x}$$

Particle filter (PF) is an approximation that uses sequential Monte Carlo methods to reach a solution with a finite number of calculations.

When the initial state is unknown, and needs to be found, the initial distribution is approximated by a uniform distribution over an appropriate region of the state space. Following steps describe the algorithm in detail.

**Algorithm** Particle Filtering

Step 1

- Draw N samples $x_0$ from the state space with importance weights=1/N and set t = 1

Step 2

- Draw N samples $\tilde{x}_t^{(i)}$ from $p\left(x_t \mid x_{t-1}^{(i)}\right)$ $i = 1, \ldots, N$
- Evaluate the importance weights $w_k^{(i)} = p\left(y_k \mid \tilde{x}_k^{(i)}\right)$
- Normalize the weights

Step 3

- Resample with replacement N particles from $\tilde{x}_t^{(i)}$ according to the weights
- Set $t \to t+1$ and go to step 2

Conditional probability of each particle ($\tilde{x}_t^{(i)}$) at $t=k$ is changed at step 2. Resampling in step 3 is based on the weights associated with the particles that could result in small, average and large values according to the conditional probability. Resampling draws N samples from $\tilde{x}_t^{(i)}$ $i = 1, \ldots, N$ by repeating the particles with larger weight and removing the ones with smaller weights. Even though this step improves the resolution of the area with higher probability, it does not improve the accuracy of the initial state $x_0$.

## Brownian Motion

Dispersion of small solid and gaseous particles plays an important role in many natural processes and environments and lead to the formation of complex structures. As such processes are very hard to model detailed empirical work on the physical conditions and the parameter space for a variety of

different dispersion scenarios is needed.

In the past few years, particle dispersion in buildings and urban areas has received increasing attention from the scientific community. Knowledge and technology of chemical and biological particle dispersion are still far from mature and much can be done by increasingly understanding the underlying physics of the different practical applications. A number of recent successful theories of particle transport are based on the ideas of Brownian motion [1].

Consider a foreign particle immersed in a flow of dense fluid. The trajectory of such particle follows an irregular and random path. The force on such a particle is regarded to be the result of two components. First one is the frictional force due to the drag extended on the particle and the other being the fluctuating force, $A'(t)$. If $u$ represents the velocity of the particle then the frictional force is assumed to be proportional to $u$. Using the Stoke's law it is calculated to be $-\gamma' u$ where $\gamma'$ is the frictional constant. $\gamma'$ is given by $6\pi a\eta$, where $\eta$ is the viscosity of the medium and $a$ is the radius of the particle. The random force $A'(t)$ represents the continuous collision the particle with the particles in immersed media. Using Langevin equation:

$$m\frac{du}{dt} = -\gamma' u + A'(t)$$

where $m$ is the mass of the particle. The above equation can be expressed as,

$$\frac{du}{dt} = -\varsigma u + A(t) \qquad (1)$$

where $\varsigma = \gamma'/m$ and $A(t) = A'(t)/m$. The following assumptions are crucial for the solution of (1).

1) The mean of the fluctuating force $A(t)$ over the ensemble of particles starting with the same initial velocity $u_0$ at $t = 0$ is zero. i.e. $E\{A(t)\} = 0$

2) It is assumed that $A'(t)$ is independent of $u$. The values $A(t)$ at two different times $t_1$ and $t_2$ are not correlated except for small intervals $(t_1 - t_2)$

$$E\{A(t_1)A(t_2)\} = \phi(|t_1 - T_2|)$$

where $\phi(x)$ is a function with a very sharp maximum at $x = 0$, $\phi(x)$ being very small for $x \neq 0$

3) The correlation of $A(t)$ obey the following:

$$E\{A(t_1)A(t_2)\ldots A(t_{2n+1})\} = 0$$

$$E\{A(t_1)A(t_2)\ldots A(t_{2n})\} = \sum_{all\ pairs} E\{A(t_i)A(t_j)\}E\{A(t_k)A(t_l)\}\ldots$$

Assumption (2) describes the sampling interval of time $\Delta t$ during which, rapid changes to $A(t)$ is expected where as changes in $u(t)$ is expected to be very small. To solve the above equation, we must solve a stochastic differential equation. That is the probability of the solution $W(u,t;u_0)$ is $u$ at the time $t$, given $u-u_0$ at $t=0$. It can be shown that probability distribution of $W$ is Gaussian.

Using the knowledge of linear first order differential equations we can solve equation (1)

$$u = u_0 e^{-\varsigma t} + e^{-\varsigma t}\int e^{-\varsigma \tau} A(\tau) d\tau \,.$$

Since an analytical solution involves rigorous calculations, a numerical solution is adopted in many situations.

Rewrite (1),

$$u_k - u_{k-1} = -\varsigma u_{k-1}\Delta t_k + \sigma A_k(\Delta t_k)$$

where $\Delta t_k = t_k - t_{k-1}$. Then

$$u_k = u_{k-1}(1-\varsigma \Delta t_k) + \sigma A_k(\Delta t_k)$$

and this can be solved iteratively.



## Simulations & Results

Flow inside a building is defined by the Navier-Stokes equations which can be found in any standard text book in fluid mechanics [4], [5]. Three dimensional (3D) fluid flow is analyzed using software Airpak/Fluent. Models are created and solved to extract the velocity information of the fluid flow within the framework. Data is extracted for node points in the space of fluid flow that are not necessarily placed in an equally spaced grid. Data is exported into Matlab where it is processed and placed in a multi-dimensional array that represents an equally spaced grid. This is done by approximating the velocities in and in $z$ direction using spline curves. It is assumed that the origin of contaminant is within the scope of one or more sensors. A fair estimate of the number of particles released is also assumed to be known.

For simulation purposes we consider a three dimensional room with length 20 meters, width 20 meters and height 10 meters. Figure 1 shows the fluid flow inside the room while table 1 shows the system parameters. Contaminant particles are introduced at coordinates $x = 4.0$, $y = 6.0$, $z = 14.0$. Figure 2 shows the contaminant transport for 100 particles. Six sensors are placed to record the contaminant concentration at locations given by (6,4,0.0), (0.0,7.5,10.0), (6.5,5.0,20.0), (20.0,2.5,10.0), (10.0,0.0,10.0), (3.5,7.0,15.0).

**Figure 1**
**Fluid flow in a 3D room**

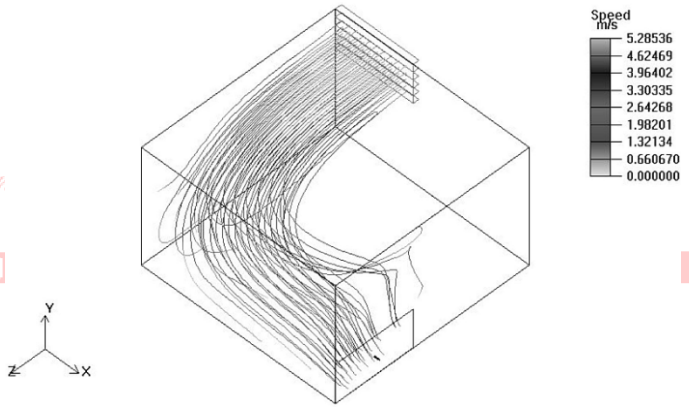




**Figure 2**
**Contaminant dispersion inside a room**

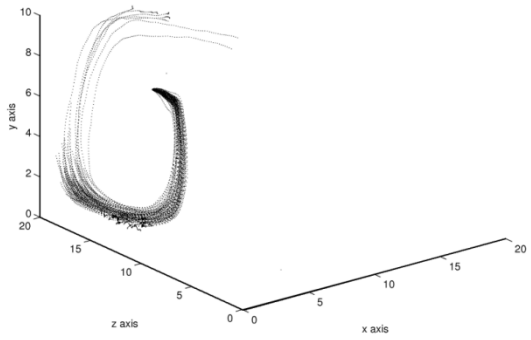


**Table 1**
**System Data**

| Name of the parameter | Value |
|---|---|
| time step | 0.15 sec |
| Process noise variance | 0.05 |
| System noise variance | 0.05 |
| velocity coefficient | 0.25 |
| random coefficient | 1.00 |

The sensor characteristics for these examples are described by

$$y = \sum_{i=1}^{N} e^{a\sqrt{(x_s - x)^2 + (y_s - y)^2 + (z_s - z)^2}}$$

where N is the number of particles that falls within the scope of the particular sensor, $a < 0$ is a constant specific to the sensor, $x_s, y_s, z_s$ are the coordinate of a given sensor, and $x, y, z$ represent the coordinates of a contaminant particle 324 initial points are generated from a grid spaced with 2.0 m in $x, y, z$ directions. Table 2 shows the convergence of the solution. It should be noted that the origin of the contaminant is covered by the initial points generated.

**Table 2**
**Convergence of Estimated ($x_0$; $y_0$; $z_0$)**

| Step | Time (sec) | $x_0$ | $y_0$ | $z_0$ |
|---|---|---|---|---|
| 1 | 0.8440 | 4.000 | 6.000 | 14.000 |
| 2 | 1.6410 | 4.000 | 6.000 | 14.000 |

*A. Modified Importance Weights Formula*

In order to test the stability and reliability of the importance weight formula, let us introduce the contaminant at (3.80, 6.20, 13.80). Let us use the same initial points as before and we expect the solution to converge to the closest point (4,6,14) on the grid. Instead, Table 3 shows that solution converge to the point (4,8,14).

**Table 3**
**Without Modified Importance Weights**

| Step | Time (sec) | $x_0$ | $y_0$ | $z_0$ |
|---|---|---|---|---|
| 1 | 0.8280 | 4.000 | 7.9505 | 14.000 |
| 2 | 1.6100 | 4.000 | 8.0000 | 14.000 |
| 3 | 2.3910 | 4.000 | 8.0000 | 14.000 |

It was noted that accuracy of the solution can be improved by multiplying each importance weight by a factor determined by considering all the observations at sensors.

Let

$$a_{ij} = \frac{\text{output at sensor } i \text{ for initial point } j}{\sum_{j=1}^{N} \text{output at sensor } i \text{ for initial point } j}$$

Let

$$v = \begin{bmatrix} \sum_{i=1}^{6} a_{i1} \\ \vdots \\ \sum_{i=1}^{6} a_{iN} \end{bmatrix}$$

where N is the number of initial points. Now importance weights are multiplied by *v* and normalized to obtain modified importance weights.

$$q(i) = \frac{v(i)q(i)}{\sum_{i=1}^{N} v(i)q(i)}$$

Table 4 shows the solution after modifying the importance weights formula and it converges to the anticipated point.



**Table 4**
**With Modified Importance Weights**

| Step | Time (sec) | $x_0$ | $y_0$ | $z_0$ |
|---|---|---|---|---|
| 1 | 0.844 | 4.000 | 7.0526 | 14.000 |
| 2 | 1.625 | 4.000 | 6.0062 | 14.000 |
| 3 | 2.406 | 4.000 | 6.0000 | 14.000 |

*B. Expanding initial points in multi-stages*

Contaminants are introduced at (8.32, 7.40, 6.76) and initial points (14,079 points) are generated at the nodes of a grid with intervals of 1.0, 0.5, 0.5 meters in x, y and z direction respectively. Table 5 indicates less accurate estimate with high computational time when large number of initial points are used.

**Table 5**
**Using a Dense set of Initial Points in a Single Stage**

| Step | Time (sec) | $x_0$ | $y_0$ | $z_0$ |
|---|---|---|---|---|
| 1 | 122.8900 | 8.0566 | 7.4800 | 6.8041 |
| 2 | 245.0930 | 8.0018 | 7.5045 | 6.8015 |

Let us introduce initial points at different resolutions in multi-stages. First initial points are generated to cover the whole space and PF is executed for a sufficient number of iterations to obtain a subset of initial points. These initial points are further expanded into their surrounding in order to obtain improved accuracy. In the first stage, 729 points generated from a grid with (in x, y, z order) 2, 1, 2 meter intervals are used. Four steps are calculated using particle filter algorithm and the solution is expanded to its surrounding creating 343 points in the second stage. The solution reached in second stage is expanded as before to obtain 637 points after five steps of calculation. Table 6 shows that the solution reaches an accurate estimate while dramatically reducing the computational time (clearly more than ten times faster compared to single stage result).

**Table 6**
**Solution in Multi-stages**

| Stage 1 with 729 points | | | | |
|---|---|---|---|---|
| Step | Time(sec) | $x_0$ | $y_0$ | $z_0$ |
| 1 | 1.8750 | 7.9314 | 7.8573 | 6.0027 |
| 2 | 3.6720 | 8.0027 | 8.0000 | 6.0000 |
| 3 | 5.4690 | 8.0000 | 8.0000 | 6.0000 |
| 4 | 7.2820 | 8.0000 | 8.0000 | 6.0000 |
| Stage 2 with 343 points | | | | |
| 5 | 8.1720 | 8.2895 | 7.6082 | 6.7968 |
| 6 | 9.0000 | 8.3450 | 7.5570 | 6.7295 |
| 7 | 9.8280 | 8.3372 | 7.5440 | 6.5938 |
| 8 | 10.6410 | 8.3971 | 7.5037 | 6.7000 |
| 9 | 11.4530 | 8.4985 | 7.5007 | 6.5426 |
| Stage 3 with 637 points | | | | |
| 10 | 13.1250 | 8.2807 | 7.5831 | 6.8125 |
| 11 | 14.7030 | 8.2870 | 7.5185 | 6.7476 |
| 12 | 16.2500 | 8.2512 | 7.4998 | 6.7500 |
| 13 | 17.7970 | 8.2504 | 7.4923 | 6.7500 |
| 14 | 19.3440 | 8.2504 | 7.4962 | 6.7500 |
| 15 | 20.8910 | 8.2500 | 7.4988 | 6.7500 |
| 16 | 22.4530 | 8.2500 | 7.4994 | 6.7500 |
| 17 | 24.0160 | 8.2500 | 7.4998 | 6.7500 |

*C. Use of observations to generate initial states*

Limiting the number of initial points in particle filters can dramatically improve the computational time. One way of doing this is to generate initial points that falls into the scope of sensors that reports a concentration level. This idea can be further extended by considering the intersection of the scopes of sensors whose output records contaminant particles at time $t = 0$. Contaminants are introduced at (8.32, 7.40, 6.76). Consider the scopes of two sensors and generate initial points with intervals 0.25, 0.125, 0.25 in x, y and z direction respectively. For this example we generate 1086 points to compute the solution (Table 7).

**Table 7**
**Using Observations of Two Sensors**

| Step | Time (sec) | $x_0$ | $y_0$ | $z_0$ |
|---|---|---|---|---|
| 1 | 6.6560 | 8.2000 | 7.4281 | 6.8105 |
| 2 | 9.4530 | 8.2917 | 7.5061 | 6.7618 |
| 3 | 12.2190 | 8.2493 | 7.4818 | 6.7500 |
| 4 | 15.0160 | 8.2500 | 7.4859 | 6.7500 |
| 5 | 17.7970 | 8.2500 | 7.4963 | 6.7500 |
| 6 | 20.5630 | 8.2500 | 7.4988 | 6.7500 |
| 7 | 23.3280 | 8.2500 | 7.4998 | 6.7500 |
| 8 | 26.0940 | 8.2500 | 7.4998 | 6.7500 |
| 9 | 28.8750 | 8.2500 | 7.5000 | 6.7500 |

## Conclusions

Use of Particle filters in estimating the origin of contaminants is explored and we can anticipate sequential Monte Carlo methods would result in better convergence for similar types of problems. This paper considers a simple form of particle filter algorithm along with modifications to importance weights formula. In addition, initial points were introduced at multi stages and generated based on sensor observations. This results in an increased performance with a dramatic reduction in computational cost.

† *Channa Navaratna, Ph.D.*, Indiana University of PA, Pennsylvania, USA
‡ *Menaka Navaratna, Ph.D.*, Florida Gulf Coast University, Florida, USA

Acknowledgement: We express our greatest appreciation to late Dr. W.P. Dayawansa for his guidance and support

## References

[1] Donald A. McQuarrie, *Statistical Mechanics*, University Science Books, Sausalito, CA 2000
[2] B. D. O. Anderson and J. B. More, *Optimal Filtering*, Prentice Hall, Englewood Cliffs, NJ 1979.
[3] Ian B. Rhodes, "A tutorial Introduction to Estimation and Filtering", *IEEE transcation on Automatic Control*, Vol AC-16, No 6, December 1971, pp 688-706.
[4] Paul A. Longwell, *Mechanics of Fluid Flow*, McGraw- Hill, Inc., New York, 1966
[5] Arthur G. Hansen, *Fluid Mechanics*, John wiley and sons Inc., New York, 1997
[6] Arnaud Doucet, Nando de Freitas, Neil Gordan, *Se- quential Monte Carlo Methods in Practice*, Springer- Verlag, New York, 2001
[7] Dan Crisan and Arnaud Doucet, "A Survey of Con- vergence Results on Particle Filtering Methods for Practitioners", *IEEE transactions on signal processing*, Vol 50, No 3, March 2002
[8] Rudolph van der Merwe, Eric Wan, "Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models", *Proceedings of (ICASSP '03) IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003, Vol- ume: 6, pp: 701- 704.
[9] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, Robust Monte Carlo Localization for Mobile Robots, *Artificial Intelligence Journal*, 2001.
[10] Javier Nicolas Sanchez, Adam Milstein, and Evan Williamson, "Robust Global Localization Using Clus- tered Particle Filtering", *Proceedings of AAAI Robotics, Artificial Intelligence* , 2002
[11] S.T. Pfister, K.L. Kriechbaum, S.I. Roumeliotis, J.W. Burdick, "Weighted range sensor matching algorithms for mobile robot displacement estimation", *Proceedings of ICRA '02 IEEE International Conference on*

*Robotics and Automation*, 2002., Volume: 2 , May 2002 –pp. 1667 -1674
[12] Kenneth R. Muske, James W. Howse, "Comparison of recursive estimation techniques for position tracking radioactive sources", *Proceedings of the American Con- trol Conference*, Volume: 2 , June 2001 pp.: 1656 -1660