Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

Non-linear Approximation Filtering in Estimating the Origin of Contaminant Particles in a Fluid Flow Using Fixed Sensors

Channa Navaratna
Department of Mathematics and Statistics
Texas Tech University
Lubbock, TX 79409-1042, USA
channa@math.ttu.edu

W. P. Dayawansa
Department of Mathematics and Statistics
Texas Tech University
Lubbock, TX 79409-1042, USA
daya@math.ttu.edu

Abstract— **The problem of accurately locating the origin of contaminant particles from noisy measurements obtained from a finite number of fixed sensors is treated here. The main methodology used here is particle filtering techniques in conjunction with model reduction. The physical system considered here consists of contaminant particles originating from an unknown location in a confined space which are being carried away by the airflow inside a room. The path of the particles is influenced by both air flow inside the room as well as the inherent random movement of the particles. Concentration of the particles at each sensor is measured at discrete time instances. These measurements are inaccurate due to additive random noise and will be processed under a particle filter algorithm to estimate the origin of contaminant particles. Particle filters in nature involve heavy computational cost and it is shown that it can be greatly reduced by executing the algorithm in multi stages, modifying the importance weights formula, generating initial points using observations and reducing the dimensionality of the mathematical model. Extensive simulation experiments are carried out to show the effectiveness of particle filters in calculating the origin of contaminant.**

I. INTRODUCTION

There is no question that revolutionary technological advancements have brought many comforts to the man, yet they also have their own undesirable and complicated side effects. Nature on its own, sometimes expel hazardous gases or liquids into the atmosphere. In comparison, manufacturing plants constantly produce chemical agents at a much higher scale, and many are eventually released to the environment. 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.

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 solutions have been derived for the aforementioned problem and have been extremely popular in the past [2], [3]. However, if the actual dynamical system deviates from a linear dynamical system or assumptions on characteristics of noise are incorrect, the filter may tend 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.

In the past two decades, the essential ideas behind Kalman filtering have been extended in many forms to cope with more complicated problems. Unscented Kalman filter, and Gaussian sum filter [8] are two of the examples resulted from this line of thinking. Since they either directly or indirectly use Kalman estimation techniques, they also suffer from similar shortcomings as the extended Kalman filter.

In recent years, computational power has reached to an extraordinary peak, hence 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, which belong to the family of Sequential Monte-Carlo algorithms [6], [7], [9], [14]. Recently particle based sampling filters have been proposed and used successfully to recursively update the posterior distribution of $p(x_k/\{y_1,\dots,y_k\})$ 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 $\{y_1,\dots,y_k\}$ available up to $t = k$. This class of problems tend to be more difficult and computationally more expensive compared to estimating a state for $t > 0$ given x_0 [10], [16]. Here it is shown that particle filtering provides a solution to the localization problem and it exhibits excellent results.

In recent years there is a great deal of suspicion that chemical and biological warfare agents have reached into the hands of terrorists, and it is considered that chances of them being used against civilians is a near certainty. Since the consequences of biological agents are unimaginable, it is foremost important to locate the origin of such a source in a very short amount of time to contain the contaminant or to evacuate the people to a safer place. In addition, quick

0-7803-8335-4/04/$17.00 ©2004 AACC

location of the source of release would be of a great deal of help to identify culprits and apprehend them. Here it is shown that appropriate modification of particle filtering is feasible so as to dramatically reduce computational time, and can be carried out only using desktop computing power well within practical limits required for containment or evacuation.

II. Particle Filtering

Consider the following discrete time non-linear system

$$x_{k+1} = f(x_k) + \omega_k \tag{1}$$

$$y_k = g(x_k) + \theta_k \tag{2}$$

where $x_k \in \mathcal{R}^n,\ y_k \in \mathcal{R}^d$ and $\omega_k,\ \theta_k$ are independent noise processes of appropriate dimensions. It is assumed that the initial distribution x_0 is independent of ω_k and θ_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 $p(x_k|x_{k-1})$. It is also assumed that the conditional distribution of the outputs is $p(y_k|x_k)$.

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

$$\hat{x} = E\left[x_k|Y_k\right] = \int x_k p(x_k|Y_k) dx_k \tag{3}$$

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

$$p(X_k|Y_k) = \frac{p(Y_k|X_k)p(X_k)}{\int p(Y_k|X_k)p(X_k)dX_k}. \tag{4}$$

$$\tag{5}$$

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

$$p(X_{k+1}|Y_{k+1}) = p(X_k|Y_k)\frac{p(y_{k+1}|x_{k+1})p(x_{k+1}|x_k)}{p(y_{k+1}|Y_k)}. \tag{6}$$

Marginal distribution of $p(x_k|Y_k)$ can be calculated as follows:

$$p(x_k|Y_k) = \frac{p(y_k|x_k)p(x_k|Y_{k-1})}{\int p(y_k|x_k)p(x_k|Y_{k-1})dx_k} \tag{7}$$

Particle filter (PF) is an approximation that uses sequential Monte Carlo methods to reach a solution with a finite number of calculations. This method involves representing marginal distribution using particles and a set of corresponding weights [6].

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.

A. ***Algorithm*** *Particle Filtering*

Step 1

- Draw N samples x_0 from the state space with importance weights= $\frac{1}{N}$ and set $t = 1$

Step 2

- Draw N samples $\tilde{x}_t^{(i)}$ from $p(x_t|x_{t-1}^{(i)})\quad i = 1, \ldots, N$
- Evaluate the importance weights

$$\tilde{w}_k^{(i)} = p(y_k|\tilde{x}_k^{(i)})$$

- Normalize the weights

Step 3

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

Conditional probability of each particle ($x_t^{(i)}$) 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 $x_t^{(i)}\quad i = 1, \ldots, N$ 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.

III. 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. Applications for these basic physical processes ranges from environmental science to astrophysics

In the past few years, particle dispersion in buildings and urban areas have received increasing attention from the scientific community. Much focus is being concentrated on the study of chemical and biological particle dispersion where models incorporate ultra-fine particles to solid particles. However, knowledge and technology 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 is based on the ideas of Brownian motion [1].

Consider a foreign particle immersed in a flow of dense fluid. The trajectory of such particle follow 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 $\mathbf{u}$ represents the velocity of the particle then the frictional force is assumed to be proportional to $\mathbf{u}$. Using the Stoke's law it is calculated to be $-\gamma'\mathbf{u}$ where γ' is the frictional constant. γ' is given by $6\pi a\eta$, where η

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:

$$m\frac{d\mathbf{u}}{dt} = -\gamma'\mathbf{u} + A'(t) \tag{8}$$

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

$$\frac{d\mathbf{u}}{dt} = -\zeta\mathbf{u} + A(t) \tag{9}$$

where $\zeta = \gamma'/m$ and $A(t) = A'(t)/m$. The following assumptions are crucial for the solution of (9).

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

$$\mathbf{E}\{A(t)\} = 0$$

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

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

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:

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

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

Assumption (2) describes the sampling interval of time Δt during which, rapid changes to $A(t)$ is expected where as changes in $\mathbf{u}(t)$ 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 $W(\mathbf{u}, \mathbf{t}; \mathbf{u_0})$ is $\mathbf{u}$ at the time t, given $\mathbf{u} - \mathbf{u_0}$ 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 (9)

$$\mathbf{u} = \mathbf{u_0}e^{-\zeta t} + e^{-\zeta t}\int_0^t e^{\zeta\tau}A(\tau)d\tau. \tag{10}$$

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

Rewrite (9),

$$u_k - u_{k-1} = -\zeta u_{k-1}\Delta t_k + \sigma A_k(\Delta t_k) \tag{11}$$

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

$$u_k = u_{k-1}(1 - \zeta\Delta t_k) + \sigma A_k(\Delta t_k) \tag{12}$$

and this can be solved iteratively.

IV. 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 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 a 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 $x,\ y$ and in z direction using spline curves. It is assumed that the origin of contaminant is within the scope of one or more sensors. Thus the time of contaminant release can be calculated using the characteristics of the sensor. A fair estimate of the number of particles released is also assumed to be known.

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

TABLE I
SYSTEM DATA

Name of the parameter	Value
Number of particles used	100
time step	0.15 sec
Process noise variance	0.05
System noise variance	0.05
velocity coefficient	0.25
random coefficient	1.0

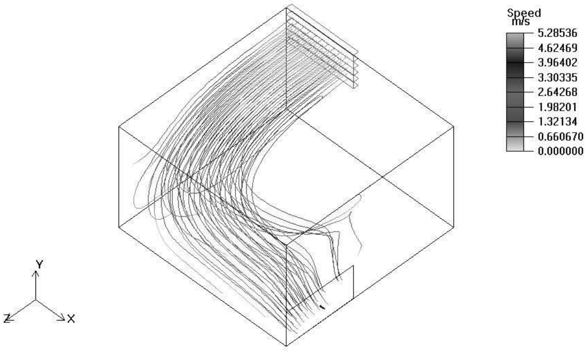

Fig. 1. Fluid flow in a 3D room

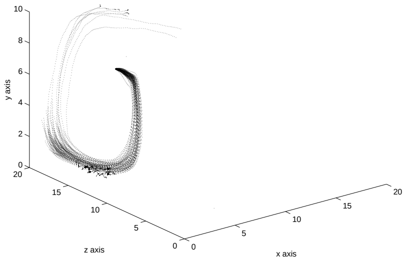

Fig. 2. Contaminant dispersion inside a room

The sensor characteristics for this examples are described by

$$y = \sum_{i=1}^{N} e^{a\sqrt{(x_s-x)^2+(y_s-y)^2)+(z_s-z)^2}} + \omega_k, \tag{13}$$

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 $2.0m$ in $x,\ y,\ z$ directions. Table II shows the convergence of the solution. It should be noted that the origin of the contaminant is covered by the initial points generated.

A. Modified Importance Weights Formula

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

TABLE II
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

It was noted that accuracy of the solution can be improved by multiplying each importance weight $q_{old}(i)$ by a factor determined by considering all the observations at sensors.

Let

$$a_{ij} = \frac{\text{output at sensor } i \text{ for initial point } j}{\sum_{j=1}^{N} \text{output at sensor } i \text{ for initial point } j} \tag{14}$$

Let

$$v = \begin{bmatrix} \sum_{i=1}^{6} a_{i1} \\ \vdots \\ \sum_{i=1}^{6} a_{iN} \end{bmatrix} \tag{15}$$

where N is the number of initial points. Now importance weights are multiplied by v and normalized to obtain modified importance weights $q_{new}(i)$.

$$q_{new}(i) = \frac{v(i) q_{old}(i)}{\sum_{i=1}^{N} v(i) q_{old}(i)} \tag{16}$$

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

B. Expanding initial points in multi-stages

Contaminants are introduced at ($x = 8.320$, $y = 7.400$, $z = 6.760$) and initial points (14,079 points) are generated at the nodes of a grid with intervals of 1.0 , 0.5, 0.5 meters

TABLE III
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

TABLE IV

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

TABLE V

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

TABLE VI

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

TABLE VII

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

in x, y and z direction respectively. Table V indicates that it takes considerable time with large number of initial points while resulting a less accurate estimate.

Let us try to solve the same problem by introducing 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 VI shows that the solution reach an accurate estimate while dramatically reducing the computational time (clearly more than ten times faster compared to single stage result).

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 $(x = 8.3200,\ y = 7.4000,\ z = 6.7600)$. 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 VII).

If three sensors report positive concentrations at $t = 0$, the number of initial points reduces to 50 and the results are shown in table VIII.

V. Conclusions and Suggestions for Future Work

A. Conclusions

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. Extensive simulation experiments are carried out to show the effectiveness of particle filters in calculating the origin of contaminant.

TABLE VIII

USING OBSERVATIONS OF THREE SENSORS

Step	Time (sec)	x_0	y_0	z_0
1	3.9850	8.1429	7.3878	6.8265
2	4.1100	8.2917	7.5859	6.7865
3	4.2350	8.2553	7.5266	6.7500
4	4.3440	8.2553	7.5000	6.7500
5	4.4530	8.2500	7.5000	6.7500

The simulations were performed on a 2.0Ghz computer using Matlab routines that generally consumes more time compared to low level languages. The calculation time can be further reduced by implementing the code in a lower level language (e.g. C, Fortran) or/and using faster computing power. It is also noticed that calculations for each initial point is independent of the other initial points, hence one can further reduce the computational time by executing the code on a parallel computer with multiple processors.

B. Future Works

In recent times, many improvements has been made to the particle filter algorithms to improve convergence properties. Use of Kalman filters (e.g. extended Kalman filter, unscented Kalman filter) have given rise to a better class of particle filters. These modification contribute additional computational time while improving the convergence. One should weigh the computational time against the accuracy for time sensitive applications (e.g. hazardous particle detection/tracking, missile tracking) It is clear that the avenues of possibilities in improving particle filters are immense in this type of problems.

VI. 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, *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York, 2001

[7] Dan Crisan and Arnaud Doucet, ”A Survey of Convergence 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, Volume: 6, page(s): 701- 704.

[9] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, Robust Monte Carlo Localization for Mobile Robots, *Artificial Intelligence Journal*, 2001.

[10] Kenneth R. Muske, James W. Howse, ”Comparison of recursive estimation techniques for position tracking radioactive sources”, *Proceedings of the American Control Conference*, 2001, Volume: 2 , June 2001 Page(s): 1656 -1660

[11] T. K. Flesch, J. D. Wilson, and E. Yee, ”Backward-time Lagrangian stochastic dispersion models and their application to estimate gaseous emissions”, *Journal of Applied Meteorology*, Volume 34, page(s):1320-1332, 1995.

[12] Petra Seibert, Andreas Frank, ”Backward Lagrangian Particle Modelling and Source-Receptor Matrices as a Tool in Environmental Meteorology”, *Proceeding of D-A-CH Meteorologentagung, Wien Oesterreichische Beitraegezu Meteorologie und Geophysik*, 27, 399, 2001

[13] P. Seibert, A. Frank and H. Kromp-Kolb, ”Inverse Modelling of Atmospheric Trace Substances on the Regional Scalewith Lagrangian Models”, *Proceedings from the EUROTRAC-2 Symposium*, 2002

[14] Javier Nicolas Sanchez, Adam Milstein, and Evan Williamson, ”Robust Global Localization Using Clustered Particle Filtering”, *Proceedings of AAAI Robotics, Artificial Intelligence*, 2002

[15] Pedro De Vries, Kristofer Doos, ”Calculating Lagrangian Trajectories Using Time-Dependent Velocity Fields”, *Journal of Atmospheric and Oceanic Technology*, Volume 18, Page(s): 1092-1101, June 2001

[16] 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 Page(s): 1667 -1674