Sequential Monte Carlo and Bayesian methodology in the stochastic event reconstruction problems

Piotr Kopka[1,2], Anna Wawrzynczak[2,3], and Mieczyslaw Borysiewicz[2]

[1] Institute of Computer Science, Polish Academy of Sciences,
[2] National Centre for Nuclear Research, Świerk-Otwock, Poland
[3] Institute of Computer Science, Siedlce University, Poland

Abstract

In many areas of application it is important to estimate unknown model parameters in order to model precisely the underlying dynamics of a physical system. In this context the Bayesian approach is a powerful tool to combine observed data along with prior knowledge to gain a current (probabilistic) understanding of unknown model parameters. We have applied the methodology combining Bayesian inference with Sequential Monte Carlo (SMC) to the problem of the atmospheric contaminant source localization. The algorithm input data are the on-line arriving information about concentration of given substance registered by the downwind distributed sensor's network. We have proposed the different version of the Hybrid SMC along with Markov Chain Monte Carlo (MCMC) algorithms and examined its effectiveness to estimate the probabilistic distributions of atmospheric release parameters.

Keywords: Bayesian inference, stochastic reconstruction, MCMC methods, SMC methods

1 Introduction

Accidental atmospheric releases of hazardous material pose great risks to human health and the environment. Examples, like Chernobyl nuclear power plant accident in 1986 in Ukraine or Seveso disaster in 1978, prove that it is necessary to have properly fast response to such incidents. In the case of an atmospheric release of chemical, radioactive or biological materials, emergency responders require relatively fast tools to predict the current and future locations and concentrations of substance in the atmosphere. One of the fields of application of the Bayesian approach can be problem of the localization of the dangerous substance release based only on the measured concentration sparse data.

Knowledge of the temporal and spatial evolution of a contaminant released into the atmosphere, either accidentally or deliberately, is fundamental to adopt efficient strategies to protect the public health and to mitigate the harmful effects of the dispersed material. However, to create the model realistically reflecting the

real situation based only on a sparse point-concentration data is not trivial. This task requires specification of set of parameters, which depends on the considered model. Non-inverting problems of this type are termed inverse problems: problems that can be solved in one direction but for some physical reason cannot be solved in the opposite direction. Such problems are widely encountered in several fields [1]. For instance the group method of data handling (GMDH) [2], [3] and it modifications seem to be successful as a method of inductive modeling and forecasting of complex processes and systems. The main idea of the GMDH is to create the algorithm able to construct a model of optimal complexity based only on the data. The goal is to get mathematical model to describe the processes, which will take place at object in the future. GMDH solves it, by sorting-out procedure, i.e. consequent testing of models, chosen from set of models-candidates in accordance with the given criterion. More recent developments utilize genetic algorithms or the idea of active neurons and multileveled self-organization to build models from data e.g. [4], [5].

The key idea behind statistical inversion methods is to recast the inverse problem in the form of statistical inference by means of Bayesian statistics. In the framework of Bayesian statistics all quantities included in the mathematical model are modeled as random variables with joint probability distributions. This randomness can be interpreted as parameter variability, and is reflected in the uncertainty of the true values expressed in terms of probability distributions. The solution of the inverse problem corresponds to summarizing probability distribution when all possible knowledge of the measurements, the model and the available prior information, has been incorporated. This distribution, referred as posterior distribution, describes the degree of confidence about the estimated quantity conditioned on the measurements [6].

It is clear that given a known gas source and wind field we can calculate the expected gas concentration for any downwind location. In the case of gas dispersion, the unknown state is the source strength and its location. It is obvious that if we are able to create the model giving the same point concentration of considered substance, as we get from the sensors' network, we could say that we understand the situation we face up. However, to create the model realistically reflecting the real situation based only on a sparse point-concentration data is not trivial. This task requires specification of set of models' parameters, which depends on the applied model.

A comprehensive review of past works on solutions of the inverse problem for atmospheric contaminant releases can be found in [7]. A variety of approaches to solve the atmospheric dispersion inverse problem have been explored including non-linear optimization, back-trajectory, Greens function, adjoint, and Kalman filter methods [8]. However, these methods often fail due to the inherent complexities, high-dimensionality, and/or non-linearity of the underlying physical system [9]. In [9] was introduced dynamic Bayesian modeling and the Markov chain Monte Carlo (MCMC). In [10] and [11] were presented sampling approaches to reconstruct a contaminant source for synthetic data.

In our previous work we have presented the application of the classical MCMC methods [20]. We have applied the methodology combining Bayesian inference with MCMC algorithms to the problem of the source localization. We have shown

the advantage of the MCMC algorithms that in different ways use the source location parameters' probability distributions, obtained based on available measurements, to update the marginal probability distribution of considered parameters with use of the newly received information. In this paper we examine the application of the Sequential Monte Carlo (SMC) methods combined with the Bayesian inference to the problem of the localization of the atmospheric contamination source. We present the possibility to connect MCMC and SMC to provide additional benefit in the process of event reconstruction. Proposed algorithms are tested on the synthetic release experiment.

2 Theoretical preliminaries

2.1 Bayesian inference

A good introduction to Bayesian theory can be found in [11] and [12]. Bayes' theorem, as applied to an emergency release problem, can be stated as follows:

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)} \tag{1}$$

where M represents possible model configurations or parameters and D are observed data. For our problem, Bayes' theorem describes the conditional probability $P(M|D)$ of certain source parameters (model configuration M) given observed measurements of concentration at sensor locations (D). This conditional probability $P(M|D)$ is also known as the posterior distribution and is related to the probability of the data conforming to a given model configuration $P(D|M)$, and to the possible model configurations $P(M)$, before taking into account the sensors' measurements. The probability $P(D|M)$, for fixed D, is called the likelihood function, while $P(M)$ is the prior distribution. $P(D)$ is the marginal distribution of D and is called prior predictive distribution [10]. $P(D)$ serves as a scaling factor and is crucial for model comparison; so in our case the Bayes theorem can be written as follows:

$$P(M|D) \propto P(D|M)P(M) \tag{2}$$

To estimate the unknown source parameters M using (2), the posterior distribution $P(M|D)$ must be sampled. $P(D|M)$ quantifies the likelihood of a set of measurements D given the source parameters M.

Value of likelihood for a sample is computed by running a forward dispersion model with the given source parameters M. Then the model predicted concentrations M in the points of sensors location are compared with actual data D. The closer the predicted values are to the measured ones, the higher is the likelihood of the sampled source parameters.

We use a sampling procedure with the Metropolis-Hastings algorithm to obtain the posterior distribution $P(M|D)$ of the source term parameters given the concentration measurements at sensor locations [10], [11]. This way we completely replace the Bayesian formulation with a stochastic sampling procedure to explore the model parameters' space and to obtain a probability distribution for the source location.

2.2 The likelihood function

A measure indicating the quality of the current state of Markov chain is expressed in terms of a likelihood function. This function compares the concentrations predicted from model and observed data at the sensor locations as:

$$ln[P(D|M)] = ln[\lambda(M)] = -\frac{\sum_{i=1}^{N}[log(C_i^M) - log(C_i^E)]^2}{2\sigma_{rel}^2} \tag{3}$$

where λ is the likelihood function, C_i^M are the predicted by the forward model concentrations at the sensor locations i, C_i^E are the sensor measurements, σ_{rel}^2 is the standard deviation of the combined forward model and measurement errors, N is the number of sensors. The value of σ_{rel}^2 can vary depending on the observation errors and model formulation (the assumed here value is given in chapter 3).

After calculating value of the likelihood function for the proposed state its acceptance is performed as follows:

$$\frac{ln(\lambda_{prop})}{ln(\lambda)} \geq U(0,1) \tag{4}$$

where λ_{prop} is the likelihood value of the proposal state, λ is the previous likelihood value, and $U(0,1)$ is a random number generated from a uniform distribution in the interval $(0,1)$.

It is important to note that condition (4) is more likely to be satisfied if the likelihood of the proposal is only slightly lower than the previous likelihood value. It gives a chance to choose even a little "worse" state, because the probability of acceptance depends directly on the quality of proposed state. Different likelihood functions can also be applied [13].

2.3 Posterior distribution

The posterior probability distribution (2) is computed directly from the resulting samples defined by the algorithm described above and is estimated with

$$P(M|D) \equiv \hat{\pi}^N(M) = \frac{1}{N}\sum_{i=1}^{N}\delta(M_i - M). \tag{5}$$

$P(M|D)$ represents the probability of a particular model configuration M giving results that match the observations at sensors locations. Equation (5) is a sum over the entire samples set of length N of all the sampled values M_i. Thus $\delta(M_i - M) = 1$ when $M_i = M$ and 0 otherwise. Consequently, if a Markov chain spends several iterations at the same location value of $P(M|D)$ increases through the summation (increasing the probability for those source parameters).

2.4 Sequential Monte Carlo

Sequential Monte Carlo (SMC) is designed to sample from dynamic posterior distributions. The SMC methods are easy to parallelize - the different Monte Carlo proposals can be generated and evaluated in parallel. A good introduction to SMC is present in [14, 15, 16] .

2.5 Sequential importance resampling

Sequential importance resampling (SIR) is a sequential version of importance sampling (IS) and combines IS with resampling procedure [17] . At the center of the SMC approach in our case is the generation of a weighted sample using IS method. IS uses a proposal distribution $q(.)$, that is close to target distribution $\pi(.)$ and from which it is easy to generate samples. The basic methodology is given below.

1. Generate a sample of size N from the proposal distribution $q(M)$:

$$M_{(i)} \sim q(M), i = 1, ..., N \tag{6}$$

2. Compute the importance weights:

$$\check{w}(M_{(i)}) \propto \frac{\pi(M_{(i)})}{q(M_{(i)})}, i = 1, ..., N \tag{7}$$

and define

$$w(M_{(i)}) = \frac{\check{w}(M_{(i)})}{\sum_{j=1}^{N} \check{w}(M_{(j)})} \tag{8}$$

3. The distribution $\pi(\cdot)$ is then approximated by

$$\check{\pi}^N(M) \equiv \sum_{i=1}^{N} w(M_{(i)})\delta(M_i - M) \tag{9}$$

which places the probability mass $w(M_{(1)}), ..., w(M_{(N)})$ on the support points $M_{(1)}, ..., M_{(N)}$.

Hence, the weights would be proportional to the value of likelihood. In our case to calculate the weight we use of the following formula,which is related to the likelihood function (3):

$$\check{w}(M_{(i)}) \propto -\frac{1}{ln[\lambda(M_{(i)})]}, i = 1, ..., N \tag{10}$$

Resampling is used to avoid the situation when almost all (except only a few) of the importance weights are close to zero (problem of degeneracy of the algorithm). Basic idea of resampling methods is to eliminate samples which have small normalized importance weights and to concentrate upon samples with large weights. So,:

1. for $i = 1, ..., N$ are chosen samples with indexes $k(i)$ distributed according to the discrete distribution with N elements satisfying

$$P(k(i) = l) = w(M_{(i)}) \tag{11}$$

for $l = 1, ..., N$,

2. then for $i = 1, ..., N$ for samples $M_{k(i)}$ are assigned the weights

$$w(M_{k(i)}) = \frac{1}{N}. \tag{12}$$

2.6 MCMC prior to SMC

The SMC algorithm needs some set of samples to be initialized. An ideal way to generate this initial sample is using MCMC data from first K iterations in all time steps. The resulting equally weighted MCMC set of samples can then be passed on to SMC for processing in the subsequent iteration.

We assume that the information from the sensors arrive subsequently in intervals (time steps). We start to search for the values of the model parameters M ($M \equiv M(x, y, q, \zeta_1, \zeta_2)$ for details see chapter 3) after first sensors' measurements (based on the data in time $t = 1$, see (Fig. 2)). Thus, scanning algorithm is run just after obtaining the first measurements from the sensors. Based on this information we obtain the probability distributions of the searched parameters (9) starting from the randomly chosen set of parameters M (i.e. first we start from the "flat" priori). This assumption reflects lack of knowledge about the release. The forward calculation are performed for the actual state M and likelihood function λ is calculated. Then we apply random walk procedure "moving" our Markov chain to the new position. Precisely, we change each model M parameter by the value draw from the Gaussian distribution with the zero mean and variance σ^2_M characteristic for each parameter. Standard deviations for sampling parameters are determined by the problem's domain size and refined with a trial and error procedure to ensure that the Markov chains had access to realistic ranges with minimal occurrences of stuck problem. Problem of stuck in chains can occur when the standard deviations chosen for the next iteration lead to a large number of rejected samples, causing that the chain remains in a given position for many iterations. For the proposal state the forward calculation are performed and the likelihood function λ_{prop} is again estimated. We compare this two values λ and λ_{prop} according to (4). If comparison is more favorable than the previous chain location, the proposal is accepted (Markov chain "moves" to the new location). If the comparison is "worse", new state is not immediately rejected. Random variable from binomial distribution is used to decide whether or not to accept the new state of chain. After K iteration we pass all the samples (from all m chains)to the sequential procedure. We compute importance weights by (10) and normalize them. Next we use roulette procedure to draw N samples from the set generated by Markov Chain.

This random component is important because it prevents the chain from becoming trapped in a local minimum. The pseudo code for one time step of the algorithm is given below.

2.6.1 Pseudocode 1:MCMC Prior to SMC

```
FOR j=1:ChainNumber
    Draw M(1) from a priori distribution;
    ForwardDispersion(M(1));
```

```
    Read C^M(1);
    Compute likelihood(M(1));
    Compute W(1);
    FOR i=1:K
        ChainSample(j,i)=M(i);
        ChainWeight(j,i)=W(i);
        M'=M(i)+N(0,sigma2M);
        ForwardDispersion(M');
        Read C^M';
        Compute likelihood(M');
        Compute W';
        IF likelihood(M')/likelihood(M(i-1))
                >=RND(0,1)
        THEN
            M(i+1)=M';
            W(i+1)=W';
        ELSE
            M(i+1)=M(i);
            W(i+1)=W(i);
        END IF
    END FOR
END FOR
SAMPLES=[ChainSample(1,:) ... ChainSample(ChainNumber,:)]
WEIGHTS=[ChainWeight(1,:) ... ChainWeight(ChainNumber,:)]
WEIGHT=WEIGHT/sum(WEIGHT)
FOR i=1:N
    SumOfWeights=0;
    RuletIter=1;
    rand=RND(0,1);
    WHILE(SumOfWeights<=rand)
        SumOfWeights=SumOfWeights+WEIGHTS(RuletIter);
        RuletIter++;
    END WHILE
RESAMPLES(i)=SAMPLES(RuletIter)
END FOR
```

Statistical convergence (to the posterior distribution) is monitored by computing between-chain variance and within-chain variance [11]. If there are m Markov chains of length N, then we can compute between-chain variance B with

$$B = \frac{N}{m-1}\sum_{j=1}^{m}(\bar{M}_j - \bar{M})^2 \tag{13}$$

where $\bar{M}_j$ is the average value along each Markov and $\bar{M}$ is the average of the values from all Markov chains. The within-chain variance W is

$$W = \frac{1}{m}\sum_{i=1}^{m} s_i^2 \tag{14}$$

where

$$s_i^2 = \frac{1}{N-1}\sum_{i=1}^{N}(M_{ij} - \bar{M}_i)^2 \tag{15}$$

The convergence parameter R is then computed as

$$R = \frac{var(M)}{W} \tag{16}$$

where $var(M)$ is estimate variance of M and is computed as

$$var(M) = \frac{N-1}{N}W + \frac{1}{N}B. \tag{17}$$

In this paper, we consider the following variants of scanning algorithms:

1. **Classic MCMC**

   In this algorithm, the parameter space scan in each time step t is independent form the previous ones. So, in this case we don't use information from past calculations. Classic MCMC don't use sequential mechanism.

2. **MCMC prior to SMC**

   The SMC algorithm use the set of samples generated by K iterations of Clasic MCMC algorithm as a prior distribution, but in subsequent SMC iterations don't use information from SMC results from previous time step.

3. **MCMC prior to SMC via Maximal Weights**

   This algorithm is similar to MCMC prior to SMC, but in subsequent SMC calculations uses the results obtained by SMC in the previous time steps to run calculation with use of the new measurements. As the first location of Markov chain M_0^t it select the set of M parameters for which weight in previous time step procedure was the highest. So, for $t > 1$:

   $$M_0^t \sim \ arg\ (M \in \{M_0^{t-1}, ..., M_n^{t-1}\})\ max\{w(M_i^{t-1})\} \tag{18}$$

   With this approach, we always start with the best values of the model (previously found) and correct the result with new information from sensor.

4. **MCMC prior to SMC via Rejuvenation and Extension**

   In contrast to the MCMC prior to SMC via Maximal Weights this algorithm as the first location of Markov chain M_0^t at the time $t > 1$ chooses the set of parameters M selected randomly from previous realization of resampling procedure in $t-1$ with use of the uniform distribution:

   $$M_0^t \sim U(M_0^{t-1}, M_1^{t-1}, ..., M_n^{t-1})\ a\ uniform\ distribution\ \{1, ..., n\} \tag{19}$$

   Applying the new knowledge (new measurements) the current chain is "extended" starting from selected position with use of the new data in the likelihood function calculation.

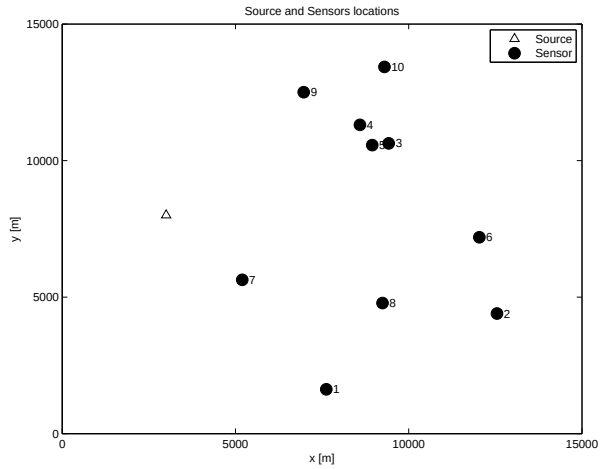

FIGURE 1: Distribution of the sensors and the release's source over the domain

3 Synthetic data

We have implemented stochastic event reconstruction algorithms grounded on the hybrid MCMC and SMC sampling to find the contamination source location based on the concentration of given substance registered by the 10 sensors distributed randomly over artificial domain 15000m x 15000m. (Fig. 1). The contamination source was located at $x = 3000m$, $y = 8000m$, $z = 30m$. The synthetic concentration measurements data (Fig. 2), used in testing the algorithm, were generated with use of the atmospheric dispersion Gaussian plume model [18], [19]. The release rate was assumed to change with time within interval $q \approx 5000g/s$ up to $q \approx 7000g/s$ which resulted in the change of the concentration measured by the sensors in subsequent time intervals (Fig. 2).The wind was directed along x axis with average speed $5m/s$.

The Markov chains are initialized by taking samples from the prior distribution.For practical reasons and to lower the computational cost we limit the prior distribution to the two dimensional coordinate space (x, y) of the source location. The vertical position of source location was fixed on $z = 30m$ at which were also located the sensors.

In our calculation we use $m = 10$ Markov chains in each time step in MCMC procedure. The traces of three independent Markov chains for the x and y parameters are presented in Fig. 3 and Fig. 4, horizontal line represents the target value. Fig. 5 presents the chain's traces in the two dimensional plane within the scanned domain, the target source location is marked by triangle and sensors by squares.The variance parameters σ_M^2 uses in random walk procedure are equal $\sigma_x^2 = 200$, $\sigma_y^2 = 200$, $\sigma_q^2 = 100$ and $\sigma_{\zeta_1}^2 = 0.02$, $\sigma_{\zeta_2}^2 = 0.02$. In this case, when we use in the reconstruction the synthetic data, the value $\sigma_{rel}^2 = 0.05$, because disorder of measurement data was low and assumed as 5%. For real measurements, if large errors in the measurements are expected, larger values of σ_{rel}^2 should be assumed.

The number of iteration for each Markov chain in Hybrid algorithm was equal $K = 10000$ (for comparision in classic MCMC $N = 20000$ to balance the number of iteration).This number was chosen based on the numerical experiments as the number of iteration needed to reach convergence for each sampled model parameters [20].One of the important aspects of stochastic procedure of calculating the posterior distribution is

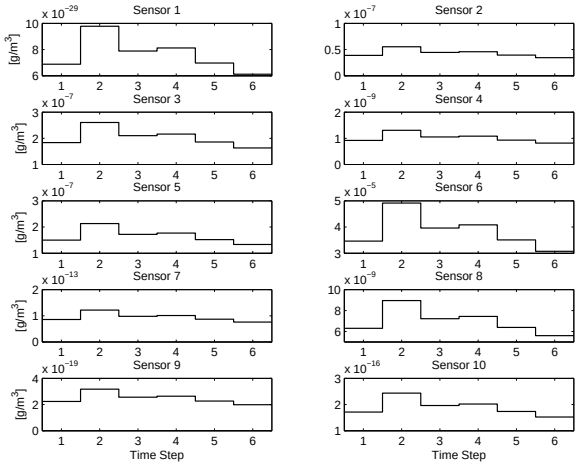

FIGURE 2: The synthetic concentration registered by the 10 sensor in 6 subsequent intervals (time steps)

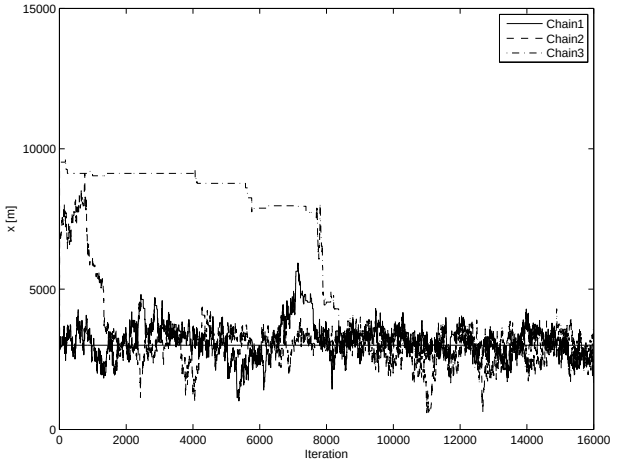

FIGURE 3: The traces of three Markov chains in the x space. The true value is marked by horizontal line. The samples came from results of Classic MCMC algorithm.

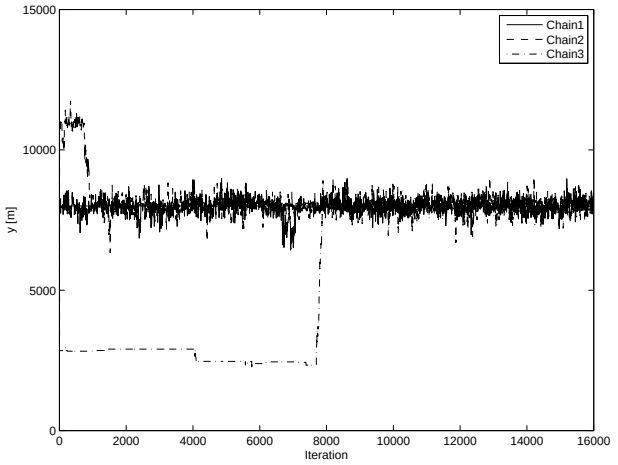

FIGURE 4: The traces of three Markov chains in the y space. The true value is marked by horizontal line. The samples came from results of Classic MCMC algorithm.

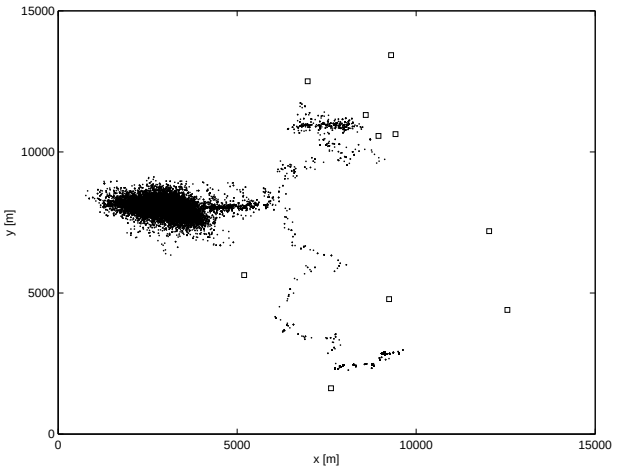

FIGURE 5: The traces of three Markov chains in the x,y space. The source location is marked by triangle and the sensors by squares. The samples came from results of Classic MCMC algorithm.

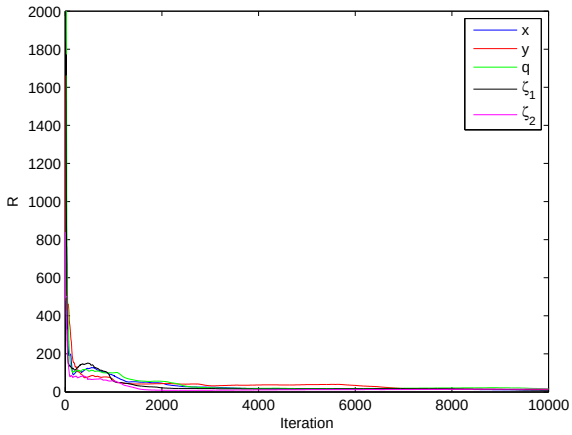

FIGURE 6: Convergence rates for position x. The samples came from results of MCMC algorithm.

choosing burn-in phase. The burn-in factor represents the number of samples needed at the beginning for the Markov chain to actually reach the search state where it is sampling from the target distribution. These initial samples are discarded and not used for inference. In our calculation the burn-in was fixed at 2000 iterations. This value was chosen based on the numerical experiments as the number of iteration needed to reach the target distribution with same approximation [20].The convergence R value vs. the number of iteration for searched parameters presents Fig. 6. One can see that the 10000 iterations satisfy the convergence condition $R \approx 1$.

3.1 Forward dispersion model

A forward dispersion model is needed to calculate the concentration C_i^M at the points i of sensors locations for the tested set of model parameters M at each Markov chain step. As a testing forward model we selected the fast-running Gaussian plume dispersion model [18],[19].

The Gaussian plume dispersion model for uniform steady wind conditions can be written as follows:

$$C(x,y,z) = \frac{q}{2\pi\sigma_y\sigma_z V} exp\left[-\frac{1}{2}\left(\frac{y}{\sigma_y}\right)\right] \times \{exp\left[-\frac{1}{2}\left(\frac{z-H}{\sigma_z}\right)^2\right] + exp\left[-\frac{1}{2}\left(\frac{z+H}{\sigma_z}\right)^2\right]\} \quad (20)$$

where $C(x,y,z)$ is the concentration at a particular location, V is the wind speed directed along x axis, q is the emission rate or the source strength and H is the height of the release; y and z are the distance along horizontal and vertical direction, respectively. In the equation (20) σ_y and σ_z are the standard deviation of concentration distribution in the crosswind and vertical direction. These two parameters were defined empirically for different stability conditions in [21] and [22]. In this case we restrict the diffusion

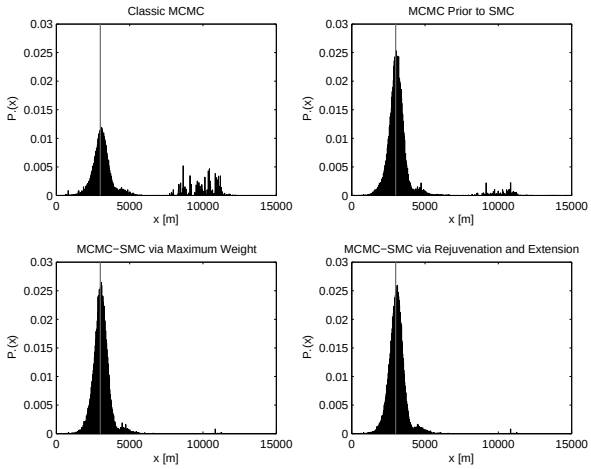

FIGURE 7: Posterior distribution as inferred by the Bayesian event reconstruction for all applied algorithms for x parameter. Posterior distributions were averaged based on the data for all time steps. Vertical lines represent the target x value.

to the stability class C (Pasquill type stability for rural area). Thus, in creation of the synthetic data we have fixed these coefficients as:

$$\sigma_y = 0.22x \cdot (1 + x \cdot 4 \cdot 10^{-5})^{-0.5}, \sigma_z = 0.2x. \tag{21}$$

However, we assume in scanning algorithm that we do not know exact behavior of the plume and consider those coefficients as not completely known. Thus, the parameters σ_y,σ_z are taken as:

$$\sigma_y = \zeta_1 \cdot x \cdot (1 + x \cdot 4 \cdot 10^{-5})^{-0.5}, \sigma_z = \zeta_2 \cdot x \tag{22}$$

where values ζ_1 and ζ_2 are sampled by scanning algorithm within interval $[0, 0.4]$. The size of the sampling interval is directly related to the choice of Pasquill Stability Class [21].
To summarize, in this paper the searched model's parameters' space is

$$M \equiv M(x, y, q, \zeta_1, \zeta_2) \tag{23}$$

where x and y are spatial location of the release, q release rate and ζ_1, ζ_2 are stochastic terms in the turbulent diffusion parameterization given in (22).

3.2 Results

All algorithms described in chapter 2.6 have been tested on the same synthetic data set. Figs. 7, 8 and 9 presents the results of calculation with use of all four above described algorithms for x, y and ζ_1 parameters. Presented distributions were calculated based on the scanning algorithms results from all time steps and all generated samples.

One can see from Figs. 7 and 8 that the classic MCMC algorithm have some the unwanted samples of x and y in ranges $x \in (1000, 1200)$ $y \in (2000, 3000)$ and marks them with a bit higher probability. MCMC prior to SMC algorithm also shows some local minima but with a lower probability value. At the same time all other algorithms

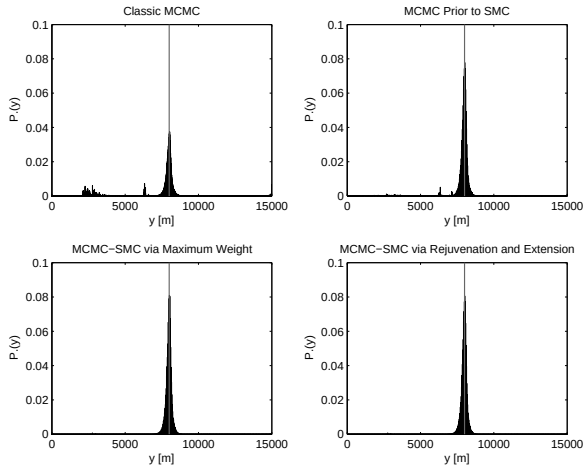

FIGURE 8: Posterior distribution as inferred by the Bayesian event reconstruction for all applied algorithms for y parameter. Posterior distributions were averaged based on the data for all time steps. Vertical lines represent the target y value.

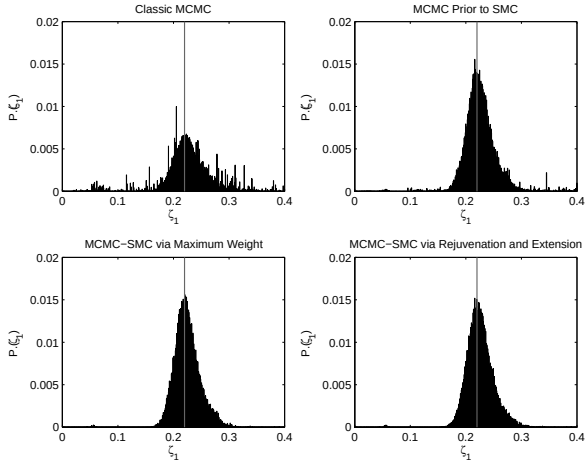

FIGURE 9: Posterior distribution as inferred by the Bayesian event reconstruction for all applied algorithms for ζ_1 parameter. Posterior distributions were averaged based on the data for all time steps. Vertical lines represent the target ζ_1 value.

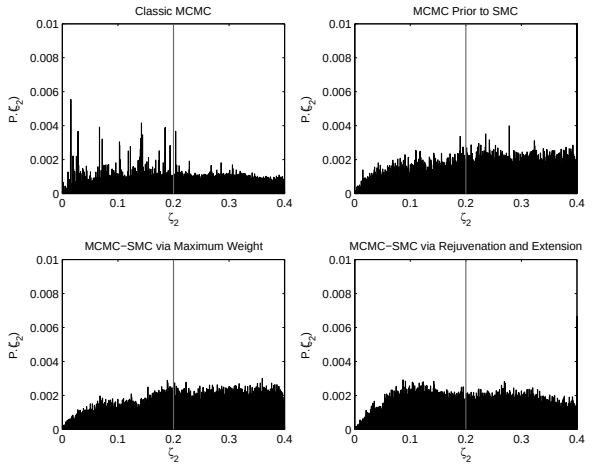

FIGURE 10: Posterior distribution as inferred by the Bayesian event reconstruction for all applied algorithms for ζ_2 parameter. Posterior distributions were averaged based on the data for all time steps. Vertical lines represent the target ζ_2 value.

reach the target value of x and y with a quite smooth and balance empirical distributions. The same in seen for the ζ_1 parameter (Fig. 9), but here the Classical MCMC algorithm do not mark the target value of ζ_1 as the most probable, while all other algorithms successfully hit its target value. The reason of the high peek in the histograms presented in Fig. 8 is that y is the crosswind direction, and applied model is quite sensitive to this parameter. In contrast, all methods do not find the target value of the ζ_2 parameter (Fig. 10) being responsible for the dispersion in vertical direction. We do not consider the probability of the release rate distribution, as far it was changing during creation of the synthetic data.

Figs. 11 and 12 presents the probability distributions of x parameter obtained in subsequent time steps by classic MCMC and SMC via Rejuvenation and Extension, respectively. One can see that in case of SMC via Rejuvenation and Extension algorithm with time the probability of the target value is reached with higher probability. Whereas in classic MCMC algorithm we observe unwanted peaks at each time step. With subsequent time steps, the algorithm MCMC prior to SMC via Rejuvenation and Extension eliminates samples with small weights, thus improves the quality of the desired distribution (e.g. local maximum is reduced to $x \in (4000, 5000)$). Confirmation of vanishing samples with small weights can be seen in Fig. 13. In first time step we can observe some outliers, while in the following steps their are discarded. A similar situation occurs for algorithms MCMC prior to SMC via Maximal Weights and MCMC prior to SMC.

4 Final conclusions

We have presented a methodology to localize a source causing an area contamination, based on a set of downwind concentration measurements. The method combines Bayesian inference with sequential Monte Carlo techniques and produces posterior probability distributions of the parameters describing the unknown source. The approach successfully provide the solution to the stated inverse problem i.e. having the downwind concentra-

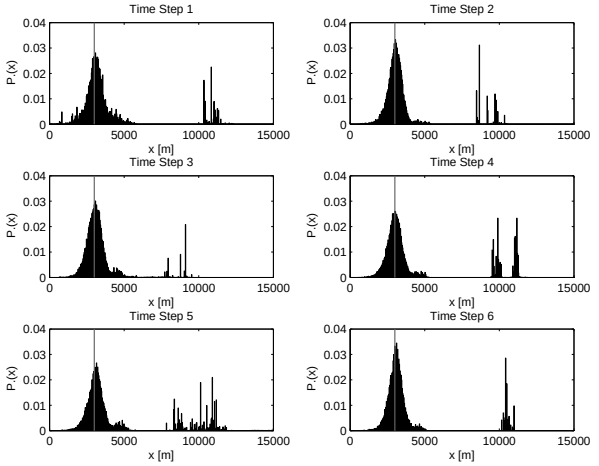

FIGURE 11: Posterior distribution of x parameter in subsequent time steps for classic MCMC algorithm. Vertical line represents the target value of x.

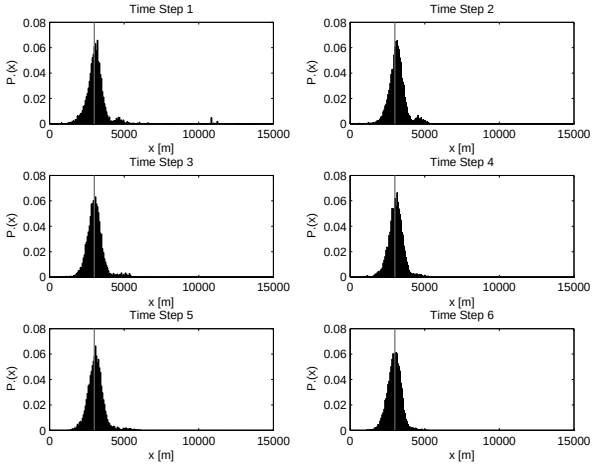

FIGURE 12: Posterior distribution of x parameter in subsequent time steps for MCMC prior to SMC via Rejuvenation and Extension algorithm. Vertical line represents the target value of x.

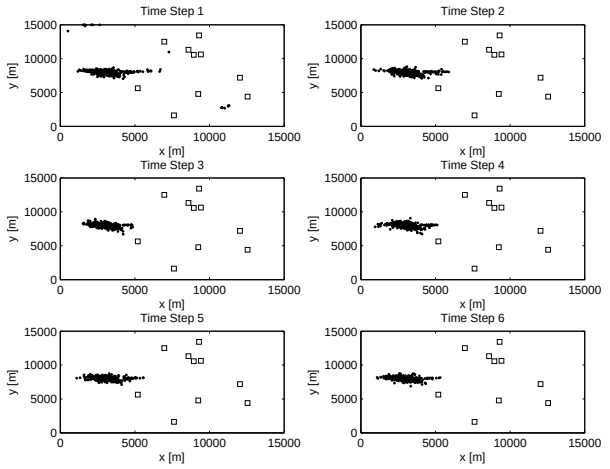

FIGURE 13: Scatter plot of all samples in subsequent time steps for MCMC prior to SMC via Rejuvenation and Extension algorithm. Squares represents the sensors.

tion measurement and knowledge of the wind field algorithm founds the most probable location of the source.

We have examined various version of the Hybrid SMC with MCMC algorithms i.e. classic MCMC, MCMC prior to SMC, MCMC prior to SMC via Rejuvenation and Extension, MCMC prior to SMC via Maximal Weights in effectiveness to estimate the probabilistic distributions of searched parameters. We have shown the advantage of the algorithms that in different ways use the source location parameters probability distributions obtained basing on available measurements to update the marginal probability distribution. As the most effective we pointed the modifications of MCMC prior to SMC.

The stochastic approach used in this paper is completely general and can be used in other fields where the parameters of the model best fitted to the observable data should be found.

Acknowledgments.

This work was supported by the Welcome Programme of the Foundation for Polish Science operated within the European Union Innovative Economy Operational Programme 2007-2013 and by the EU and MSHE grant nr POIG.02.03.00-00-013/09.

References

[1] Thomson, L. C., Hirst, B., Gibson, G., Gillespie, S., Jonathan, P., Skeldon, K. D., Padgett, M. J.: An improved algorithm for locating a gas source using inverse methods. Atmospheric Environment, 41, 1128–1134, (2007)

[2] Ivakhnenko, A.G.: Group method of data Handling- A Rival of the Method of Stochastic Approximation, Soviel Automatic Control, 13, 43–71, (1966)

[3] Madala H.R., Ivakhnenko A.G.: Inductive Learning Algorithms for Complex Systems Modeling, CRC Press, (1994)

[4] Fujimoto, K., Nakabayashi S.: Applying GMDH algorithm to extract rules from examples, Systems Analysis Modelling Simulation, 43, 10, 1311–1319, (2003)

[5] Puiga V.Witczak M.,Nejjari F.,Quevedo J.,Korbicz J.: A GMDH neural network-based approach to passive robust fault detection using a constraint satisfaction backward test, Engineering Applications of Artificial Intelligence, 20, Issue 7, 886–897, (2007)

[6] Watzenig,D.: Bayesian inference for inverse problems - statistical inversion. Elektrotechnik and Informationstechnik ,124/7/8, 240, (2007)

[7] Keats, A., Yee, E. ,Lien, F.S.: Bayesian inference for source determination with applications to a complex urban environment. Atmos. Environ., 41,465–479, (2007)

[8] Pudykiewicz, J. A.: Application of adjoint tracer transport equations for evaluating source parameters. Atmos. Environ., 32, 3039–3050, (1998)

[9] Johannesson, G. et al.: Sequential Monte-Carlo based framework for dynamic data-driven event reconstruction for atmospheric release, Proc. of the Joint Statistical Meeting, Minneapolis, MN, American Statistical Association and Cosponsors, 73–80, (2005)

[10] Gilks, W., Richardson, S., Spiegelhalter, D.: Markov Chain Monte Carlo in Practice. Chapman & Hall/CRC,(1996)

[11] Gelman, A.,Carlin, J.,Stern, H.,Rubin, D.: Bayesian Data Analysis, Chapman & Hall/CRC, (2003)

[12] Bernardo, J. M., Smith, A. F. M.:Bayesian Theory, Wiley, (1994)

[13] Senocak I.,Hengartner, N. W., Short, M. B. ,Daniel,W. B.:Stochastic Event Reconstruction of Atmospheric Contaminant Dispersion Using Bayesian Inference, *Atmos. Environ.* 42(33) 7718–7727, (2008)

[14] Doucet, A., de Freitas, J. F. G., Gordon, N. J.: Sequential Monte Carlo methods in practice. New York: Springer-Verlag, (2001)

[15] Pitt, M. K. Shephard, N.: Sequential Monte Carlo methods in practice, chapter Auxiliary variable based particle filters. New-York: Springer-Verlag, (2001)

[16] Liu, J. S.: Monte Carlo Strategies in Scientific Computing. New York: Springer, (2001)

[17] Gordon, N. J.; Salmond, D. J., Smith, A. F. M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEEE Proceedings F on Radar and Signal Processing 140 (2): 107–113, (1993)

[18] Panofsky, H. A., Dutton, J. A.: Atmospheric Turbulence, John Wiley, (1984)

[19] Turner D. B.: Workbook of Atmospheric Dispersion Estimates, *Lewis Publishers*, USA, (1994)

[20] Borysiewicz, M., Wawrzynczak A.,Kopka P.:Stochastic algorithm for estimation of the model's unknown parameters via Bayesian inference, Proceedings of the Federated Conference on Computer Science and Information Systems pp. 501–508, IEEE Press, Wroclaw, ISBN 978-83-60810-51-4, (2012)

[21] Pasquill, F.: The estimate of the dispersion of windborne material, *Meteorol Mag.*,90, 1063,: 33–49, (1961)

[22] Gifford, F. A. Jr.: Atmospheric dispersion calculation using generalized Gaussian Plum model, *Nuclear Safety*, 2(2):56–59,67–68, (1960)