UCRL-TR-217762

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Synthetic Event Reconstruction Experiments for Defining Sensor Network Characteristics

J. K. Lundquist, B. Kosovic, R. Belles

December 16, 2005

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Synthetic Event Reconstruction Experiments for Defining Sensor Network Characteristics

Julie K. Lundquist, Branko Kosović, and Rich Belles

Abstract

An event reconstruction technology system has been designed and implemented at Lawrence Livermore National Laboratory (LLNL). This system integrates sensor observations, which may be sparse and/or conflicting, with transport and dispersion models via Bayesian stochastic sampling methodologies to characterize the sources of atmospheric releases of hazardous materials. We demonstrate the application of this event reconstruction technology system to designing sensor networks for detecting and responding to atmospheric releases of hazardous materials. The quantitative measure of the reduction in uncertainty, or benefit of a given network, can be utilized by policy makers to determine the cost/benefit of certain networks.

Herein we present two numerical experiments demonstrating the utility of the event reconstruction methodology for sensor network design. In the first set of experiments, only the time resolution of the sensors varies between three candidate networks. The most "expensive" sensor network offers few advantages over the moderately-priced network for reconstructing the release examined here. The second set of experiments explores the significance of the sensors' detection limit, which can have a significant impact on sensor cost. In this experiment, the expensive network can most clearly define the source location and source release rate. The other networks provide data insufficient for distinguishing between two possible clusters of source locations. When the reconstructions from all networks are aggregated into a composite plume, a decision-maker can distinguish the utility of the expensive sensor network.

1. Introduction

To detect the atmospheric transport of hazardous materials, new and innovative sensor networks are currently being designed and deployed. These networks can serve one or more of several purposes: they can detect the spread of hazardous materials before large populations have been exposed so that emergency response officials can organize evacuations; they can identify the size of the release so that officials can respond during an event with evacuations or inoculations; they can be used in a forensic role, post-event, to describe the release so that decontamination efforts can be prescribed.

Many types of sensors and sensor networks for detecting atmospheric releases of hazardous materias have been designed; some networks have been deployed, such as the BioWatch network that is designed to provide early warning in the case of a mass pathogen release (Shea and Lister, 2003). These networks have varying degrees of detection sensitivities, false-alarm rates, and frequency of data collection. However, the utility of these networks in characterizing the sources of atmospheric releases of hazardous materials has not yet been demonstrated systematically to our knowledge.

Reconstructing the source of a detected atmospheric release is a crucial step in predicting the consequences of such a release. The primary source of uncertainty in prediction the consequence of an atmospheric release is determining the source term characteristics, such as location, magnitude, and duration of the release. The type of source is also a crucial component in determining consequences: sources may be instantaneous (like an explosion) or continuous (a long-term release), localized to one point or over a wide area, static or moving, at the surface or elevated. Even if the source is perfectly characterized, the complexity of atmospheric flow, especially in urban environments or in complex terrain, presents additional challenges for a dispersion model.

An event reconstruction technology system has been designed and implemented at Lawrence Livermore National Laboratory (LLNL). This system integrates sensor observations, which may be sparse and/or conflicting, with transport and dispersion models via Bayesian stochastic sampling methodologies to characterize the sources of atmospheric releases of hazardous materials. The event reconstruction methodology identifies source characteristics (such as location, magnitude, duration) that are most consistent with the observed data, given a quantification of the errors expected in the both the observations and in the forward dispersion model. Once the source is characterized, an ultimate prediction of likely affected areas is possible. This ultimate prediction is a composite of all the likely sources and can guide emergency responders more effectively than a single forward prediction from a single (possibly incorrect) estimate of source characteristics. The composite prediction provides a measure of uncertainty in source characterization and the result of the release of hazardous material.

Ideally, the observations describing an event would provide enough information about an event so that the uncertainty regarding the source location or magnitude is very low. To ensure this minimal uncertainty, sensor networks must be designed with that goal in mind. It is possible, using the event reconstruction system, to examine certain scenarios of interest using different sensor networks to determine which sensor network(s) will provide the greatest reduction of uncertainty in source characterization and response. This quantitative reduction in uncertainty can be provided to policy makers to determine the cost/benefit of certain networks. Questions such as the following can be addressed:

- Would a network consisting of fewer instruments that are more sensitive protect my facility better than a network with more instruments that are less sensitive?
- If I have time constraints on my response to the detection of a release, how often must I collect information from my sensors?
- To reduce costs while still protecting my city, is a dense network of instruments with a high false alarm rate a better choice than a sparse network of more reliable instruments?

To demonstrate the utility of the event reconstruction system for answering questions like these, we present a pair of numerical experiments designed to determine and quantify the advantages of using sensors of varying time resolutions and varying detection limits for identifying a source of a hazardous release that affects a suburban domain of size 6 km by 6 km. This experiment quantifies the importance of time resolution and sensor detection limit thresholds in otherwise identical instruments deployed to identify the source of a 1.5-hour-long release of a neutrally-buoyant gas in the suburban area. Although these experiments were loosely based on an actual atmospheric tracer experiment (the Copenhagen release, Grying and Lyck, 1984), such experiments using the LLNL event reconstruction methodology require only a definition of the region of interest, a climatology of atmospheric conditions for that region, some specification of the types of releases required to be considered, a transport-and-dispersion model suitable for simulating the dispersion of the material(s) of interest, and a measure of error for the sensor observations and the transport-and-dispersion model predictions.

2. Description of the event reconstruction methodology

LLNL's event reconstruction methodology (Kosovic et al., 2005) integrates Bayesian stochastic methodologies with "forward" atmospheric transport and dispersion models and observations of atmospheric concentrations due to an atmospheric release to determine unknown source parameters. The event reconstruction system can provide optimal characterization of unknown source term parameters, given a set of measurements of atmospheric concentrations M_{ij}

at locations i and times j, an atmospheric dispersion model that predicts concentrations C_{ij} at locations i and times j as a function of source term parameters, and quantification of the error in both model predictions of concentrations C and observed measurements M.

For example, event reconstruction is often used to determine probabilistic estimates of two unknown source terms parameters, source location X and source magnitude R. This final probabilistic estimate is known as the posterior distribution, which is calculated over many iterations in the following way. At an nth iteration of the reconstruction, a Markov chain samples unknown source term parameters Xn and Rn from a large set of possibilities $\boldsymbol{X}$ and $\boldsymbol{R}$. These source parameters Xn and Rn are provided to an atmospheric transport and dispersion model, which uses Xn and Rn to predict atmospheric concentrations at sensor locations i and times j. The measurements M_{ij} and the model predictions $C_{ij(XnRn)}$ are compared; details of that comparison are discussed below. Based on that comparison, the probability of source location Xn and source magnitude Rn are evaluated via comparison to previous guesses $Xn\text{-}1$, $Xn\text{-}2$, … and $Rn\text{-}1$, $Rn\text{-}2$, …. If the comparison is favorable for Xn and Rn, their values are retained for subsequent comparisons of $Xn+1$ and $Rn+1$. Eventually, convergence to a final posterior distribution is attained, and that final posterior distribution summarizes the most likely of source term parameters X and R given measurements and their error, prior knowledge about the characteristics of the source, and prior estimates of the transport and dispersion model error.

This process, known as Markov Chain Monte Carlo sampling for Bayesian inference, is discussed in detail in popular texts such as Robert and Casella (2005) and Liu (2001), e.g.. The sampling procedure used herein relies on a Metropolis-Hastings algorithm for generated samples Xn and Rn from a domain of possibilities $\boldsymbol{X}$ and $\boldsymbol{R}$. Multiple Markov Chains can proceed through the domains $\boldsymbol{X}$ and $\boldsymbol{R}$ simultaneously; four Markov Chains are used in the reconstructions presented here.

a. The “forward” atmospheric transport and dispersion model

A core component of an atmospheric event reconstruction is the efficient use of an atmospheric dispersion model. The present methodology has been used with a wide array of transport and dispersion models. These models include a relatively simple and fast 2D Gaussian puff model INPUFF (Petersen and Lavdas, 1986), a Lagrangian particle dispersion model LODI which is used at LLNL’s National Atmospheric Release Advisory Center (NARAC) (Ermak and Nasstrom, 2000, Larson and Nasstrom, 2001), and a building-resolving computational fluid dynamics code FEM3MP (Chan and Stevens, 2000; Chow et al., 2006). In each case, thousands of possible source term parameters Xn and Rn, are provided to the forward dispersion model, meaning that thousands of forward simulations are carried out. Computational efficiencies such as a Green’s function approach (Chow et al. 2006) are available to reduce the number of forward simulations, but have not been employed for the study presented herein.

The National Atmospheric Release Advisory Center's Lagrangian particle dispersion model, LODI, provided the forward atmospheric transport and dispersion simulations for this experiment. Based on a given source location and release rate, LODI generates a number of Lagrangian particles that disperse within its simulation domain based on meteorological and turbulence parameters calculated by LODI and provided to it by a meteorological data assimilation model, ADAPT (Sugiyama and Chan 1998), also developed at the National Atmospheric Release Advisory Center.

b. The likelihood function and assumed error

The quality of the reconstruction, or the precision of the final posterior distribution of the unknown source term parameters, is related to the error assumed or known in both the actual measurements used in the reconstruction and the forward model (as well as to the quality of the data used in the reconstruction). For the reconstructions discussed here, these two errors are incorporated into one error parameter, σ, which is utilized in the comparison described above.

The measurements M_{ij} and modeled concentrations C_{ij} are first compared to the detection range of the instruments used. Any measurements or modeled concentrations above the saturation level of the instrument are set to the saturation level; any measurements or modeled concentrations below the detection limit are set to the detection limit or sensor sensitivity threshold. The natural log of the likelihood function L for source Xn and Rn over all the sensor measurements N is a function of the difference between the measurements and the modeled concentrations assuming source parameters Xn and Rn:

$$\ln\left(L_{XnRn}\right) = \frac{\sum_{ij}\left(M_{ij} - C_{ij}\right)^2}{2N\sigma^2}$$

This likelihood value L_{XnRn} is compared to that of previously-tested values such as $L_{Xn-1Rn-1}$. Other likelihood functions are possible. Generally, values of source term parameters that lead to smaller values of L are retained, although some violations of that rule are allowed to ensure wide sampling of the source term domains $\boldsymbol{X}$ and $\boldsymbol{R}$ and to prevent any Markov chain from being caught in a local minimum of L. Successful likelihood values contribute to the final posterior distribution.

If larger errors in measurements or modeled concentrations are appropriate, larger values of σ should be assumed. For all reconstructions presented herein, σ=0.2. Larger values of the error range, or s, will generally lead to broader final posterior distributions. Possible bias in the measurements or in the model is not accounted for in the formulation of the likelihood function as presented here, but may be incorporated into the likelihood function.

3. Description of the numerical simulations

These simulations were roughly based on the Copenhagen tracer experiments (Gryning and Lyck, 1984), using a domain, terrain features, and meteorology from the 19 July 1979 release of sulfur hexafluoride tracer gas in the suburban Copenhagen area. Hourly averages of wind speed and wind direction from the TV tower from which the tracer gas was released were used to define a wind field in NARAC's ADAPT model; observations from four levels above the surface were available (10m, 60m, 120m, and 200m). Boundary-layer height (2090m) was estimated from a sounding released within 10km of the source. Friction velocity (0.77 m/s) was estimated from the mean wind profile. Surface roughness for the suburban domain was estimated at 0.6m. Although the meteorology was prescribed for this event reconstruction, the characteristics of meteorology could also be included in the set of unknowns that the event reconstruction system seeks to identify. Wind speed and wind direction profiles for hours 1000, 1100, and 1200 UTC (local time – 1 hour) appear in Figure 1. Note that wind direction was reported only at 10m, 120m, and 200m levels.

Two numerical studies are presented herein. The first study explores the role of time resolution of the instruments used in this study. The second study explores the role of sensor sensitivity or detection limit. Each study includes three reconstructions; all reconstructions attempt to identify the location (in the horizontal plane) and rate of release of the tracer gas. Four sensors with known, fixed locations are distributed within the domain, nominally 2-4 km from the source. (The locations of the sensors correspond to actual locations of sensors used during the 1979 experiments; these sensors are numbers 1-22, 1-38, 3-23, and 3-32 using Gryning's sensor identification system (Gryning 1981).) This situation is analogous to a scenario in which sensor locations are predetermined by logistical constraints, but sensor characteristics, such as averaging time or sensitivity, are flexible. All simulations use the same hourly meteorology and seek to characterize the same release, which commences at 1050 UTC and concludes at 1220 UTC, releasing material at a constant release rate of 3.2e+09 ng/sec.

Each simulation uses sensors from the same four locations for each simulation, as well as "synthetic" concentrations reported by LODI given the actual source location and release rate. The synthetic concentrations recorded at each location were based not only on LODI's predicted concentrations at the 100m x 100m (x 20m high) grid cell encompassing each sensor's location, but also including the eight nearest neighbors of that grid cell using the weighting scheme shown in Figure 3. Because sensors in the Copenhagen experiment were typically mounted on street poles at altitudes 0-20m above the surface, LODI concentrations for the lowest 20m are considered.

The reconstructions are summarized in Table 1. For the first study, the three reconstructions differ in the time resolution of the sensor providing data. Although all sensors recorded data from 1038-1238 UTC, the “60m resolution” sensors (where “m” indicates “minutes”) reported averaged concentrations for 1038-1138 and 1138-1238; the “10m resolution” sensors reported averaged concentrations for a total of twelve ten-minute intervals; and the “5m resolution” sensors reported averaged concentrations for a total of twenty-four five-minute intervals. In a domain of this size (6km length scale) and for wind speeds of this magnitude (average of 9.2 m/s at 60m altitude over the three relevant hours), ten minutes are required for material to be transported throughout the domain. Therefore, only an instrument with time resolution at or greater than this ten-minute time scale is expected to provide adequate information to characterize the source location and magnitude. All the sensors can detect atmospheric concentrations between 10 and 10000 ng/m3, or three orders of magnitude, based on the reported detection limits used in the original Copenhagen study (Gryning, 1981).

For the second study, the detection limit (or sensor sensitivity) varies. All the sensors report data at ten-minute intervals, as the “10m resolution” sensors in the first experiment. The “low-threshold” reconstruction uses data from instruments with an expanded lower detection limit, reporting in a range from 0.01 to 10000 ng/m3, or over six orders of magnitude. The moderate threshold sensor network reports data over three orders of magnitude, from 10 to 10000 ng/m3. The “high-threshold” reconstruction uses limited data from instruments reporting from 1000 to 10000 ng/m3.

Event reconstruction was carried out for 5000 iterations for each sensor set, searching over source x and y location and release rate. All simulations were carried out on Livermore Computing’s mcr platform, using less than 12 cpu hours on 68 2-processor nodes.

	Time Resolution of sensors (minutes)	Sensor detection range (orders of magnitude)	Qualitative description of network
four_05mres	5	3	Expensive
four_10mres	10	3	Moderate
four_60mres	60	3	Inexpensive
low_thresh	10	6	Expensive
four_10mres	10	3	Moderate
high_thresh	10	1	Inexpensive

Table 1: Summary of reconstructions discussed herein. The “four_10mres” reconstruction is utilized in both the time-resolution study and the detection limit study.

4. Assessment of the sensor networks tested

Several metrics can assess which type of network provides optimal information to users wishing to understand the source of material responsible for the observed data. Before considering these metrics, it is advisable to ensure that a reconstruction has converged. These metrics include histograms of source characteristics, probability contours of source location, and finally, composite plumes based on the posterior distribution from the reconstruction.

a. Convergence metrics

The posterior distribution can only be determined if a reconstruction has converged. Only information generated after convergence should be considered when evaluating a sensor network. Little information can be gleaned from convergence tests other than the fact that convergence has been attained, which is necessary for subsequent analysis of the posterior distribution to which the reconstruction has converged.

Convergence is typically defined (Gelman et al., 2004, p. 297) by a measure of the variation between the chains used in the reconstruction to variation within each chain. When this ratio, R_hat, approaches 1 (in practicality, is less than 2), convergence is said to be attained. Each of the three reconstructions converged rapidly, within the first 1000 iterations. Iterations 2000-5000 constitute the posterior distributions, presented here via histograms, probability contours, and composite plumes. The convergence metric R_hat for all three reconstructed parameters (x-location, y-location, and release rate) are shown in Figure 4 (for the time-resolution study) and in Figure 5 (for the sensor-threshold study).

b. Location histograms

After convergence has been attained, the posterior distribution of the reconstruction reveals the characterization of the source. For synthetic studies and reconstructions based on studies in which the exact characteristics of the source are known, as presented here, a comparison of the histogram to “truth” can generate confidence in the reconstruction. In cases for which “truth” is unknown and to be determined, the histogram’s nature (flat vs. sharp) can indicate how much information is attainable from available data.

The histograms for each of the three source characteristics reconstructed (y-location; x-location; and release rate), along with an indicator of “truth”, are shown below. The largest uncertainty in the reconstruction of a source location is typically in the direction along the mean wind, which in this case, is in the x-direction. Therefore, the histograms of y-locations should be narrowly focused and more correct.

In the time-resolution experiment, all sensor arrays identify the y-location of the source within a 3-km range, while the reconstruction explored a 10-km range (Figure 6). However, the 60-min (Figure 6c) resolution sensors provide a reconstruction with a bimodal distribution of y-location, with one peak at the correct location (indicated with the heavy vertical line) and one location 1.5 km north of the correct location. This bimodal probability distribution indicates a suboptimal network.

In the detection-threshold experiment, the “expensive” network with instruments with a low detection threshold provides a reconstruction that clearly and correctly identifies the y-location of the source (Figure 7a), indicating the utility of this type of instrument. The moderate-threshold instruments constitute a suboptimal network (Figure 7b), indicating a bimodal distribution of y-location, with one peak at the correct location (indicated with the heavy vertical line) and one location 1.5 km north of the correct location. Finally, the inexpensive network of instruments with a high detection limit provides no information on the y-location of this source, indicated by the flat distribution in Figure 7c.

As noted above, the greatest location uncertainty in a reconstruction is typically in the direction of the mean wind, which in this case was from the west. This large uncertainty is seen in the broad probability distributions for x-location.

In the time-resolution experiment, distributions of the x-locations include probable locations within a 5-km range around the correct source location (and would probably extend further upwind had the domain been large enough to include such locations). The superior time resolution of the 5-minute resolution sensor network did not provide the ability to reduce the uncertainty in x-location for this case (Figure 8a) over the uncertainty determined with the moderate network (Figure 8b). This failure disproves the hypothesis that the 5-minute resolution sensors would resolve the arrival time of the plume more precisely than the 10-minute resolution sensors, because the advection time from an upwind sensor to a downwind sensor (distance of nominally 4 km) with mean winds of 9.2 m/s is less than 8 minutes. The tracer gas release did last for ninety minutes, however; an instantaneous release of tracer gas would likely be reconstructed better by the higher time resolution sensors.

In the detection-threshold experiment, a more marked difference between the networks is evident. The low-detection threshold instruments do identify the correct x-location within 3 km, with a Gaussian distribution (Figure 9a) and a significant peak close to the real source. The moderate network identifies a broader range of possible x-locations (Figure 9b), while the high-detection limit instruments provide no information at all to reduce the range of possibilities (Figure 9c).

c. Joint location histograms

When a source characteristic, such as location, is defined by more than one parameter, such as x and y, more insight can be gleaned by the inspection of joint histograms. Presented in Figure 10 and Figure 11 are joint histograms of probability of source location for both x and y, superimposed on a map including the real source location (the red triangle) and the sensor locations (the four green diamonds), for the two experiments. Shading indicates the joint probability of a particular cell being the source location; the more intense blues represent higher probability.

In the time-resolution experiment (Figure 10), the joint probability distributions based on data from the 10-minute and 60-minute resolution sensor networks clearly illustrate that these networks cannot distinguish between two clusters of possible source locations: one band includes the correct location, and another band to the north includes a peak at the wrong location. The 10-minute resolution network has an especially strong peak at an incorrect location (x=343, y = 6481) that is weighted more strongly than the peak at the correct location. Only reconstruction based on the 5-minute resolution sensor network correctly emphasizes the southern band, which encompasses the true source location, although it does include both bands.

In the detection-limit threshold experiment (Figure 11), the advantage of the low-detection-limit network is obvious. In the joint histogram for the expensive network, only a few locations are highlighted, and those locations are very close to, within 500m of, the actual source, with some upwind uncertainty (Figure 11a). The network composed of sensors with a moderate detection limit cannot distinguish between two clusters of possible source locations: one band that includes the correct location, and another band to the north that includes a peak at the wrong location. The moderate network identifies an especially strong peak at an incorrect location (x=343, y = 6481) that is weighted more strongly than the peak at the correct location (Figure 11b). Finally, the high-detection-limit network (Figure 11c) cannot reduce the infinite range of possible source locations beyond identifying that the source location is not immediately upwind of the sensors, as noted by the white areas (indicating zero probability) upwind of the sensors. The rest of the domain consists of possible source locations, indicating that this sensor network provides no information at all to decision-makers seeking to understand the characteristics of the source of an atmospheric release of this magnitude.

d. Release rate histograms

When identifying the source of an unknown atmospheric release, the magnitude of the release is often an important parameter. Knowing the size of the source term can guide emergency-response actions, such as determining whether evacuation or sheltering-in-place is appropriate. The size of the source term is also important for post-release cleanup efforts. An ideal sensor network, coupled with event reconstruction methodologies, should therefore be able to quantify the size of a detected atmospheric release. Histograms of the release rate reconstructions for these two experiments appear in Figure 12 and Figure 13.

Inspection of the release rate histogram clearly indicates the problematic nature of reconstruction with very coarse time-resolution instruments (Figure 12). The histogram based on 60-min sensor data (Figure 12c) is very flat, filling almost the entire range of release rates considered. Both the 10-min (Figure 12b) and the 5-min (Figure 12a) sensor data narrow the field of possibilities to acceptable limits. The 10-min sensor data reconstruction indicates a slight peak in probability at the correct release rate, although the determination of the “best” sensor array should not be based on the histogram of one quantity alone, but on the aggregate evaluation of all desired source parameters, as available in the composite plume.

In the detection-limit experiment, the least expensive network again fails to provide useful information, filling almost the entire range of release rates considered (Figure 13c). The moderate network indicates a slight peak in probability at the correct release rate (Figure 13b), although this release rate peak corresponds to an incorrect location, as discussed above in Figure 11b. Surprisingly, the reconstruction based on data from the expensive network cannot precisely identify the strength of the source (Figure 13a) as well as it identifies the location of the source. Quantification of the utility of different sensor networks, via articulation of the networks’ capabilities, would be helpful information for decision-makers.

e. Composite and aggregate plume predictions

A key piece of information for a decision-maker choosing between sensor networks is a composite plume, a reconstruction of the original plume based only on the data provided from the sensors and the forward model via the Bayesian stochastic inversion. This composite summarizes the posterior distribution of all possible source characteristics, weighting each source characteristic (here, X and R) by their probability of occurring as determined by the reconstruction (and seen in the histograms presented herein). Because the refinement of the reconstruction is determined by the characteristics of the sensor network, this measure of network performance provides a quantitative means of evaluating the utility of a given sensor network.

Composite plumes are generated by aggregating together all runs from the forward model, based on the Xn and Rn tested during the reconstruction. At each grid cell in the forward model’s domain, at each time step simulated, there exists a distribution of atmospheric concentrations due to the dispersion predicted by the forward model based on each Xn and Rn. A plume dispersion plot consisting of the total concentration expected in each cell, normalized by the total number N of Xn and Rn that contributed to that concentration, would create an aggregate plume. This aggregate plume representation does not explicity incorporate the probabilistic information obtained via the event reconstruction. To incorporate the probabilistic information, a confidence threshold level is defined by the decision-maker. (In this study, the 90% confidence level is utilized.) The composite plume indicates, for each cell, that the reconstruction based on the sensor data is 90% confident that concentrations at that cell are above a certain threshold.

Composite plumes for the detection-limit experiment are shown in Figure 14. The composite plume from the reconstruction based on the low-detection-limit sensor network (Figure 14a) is able to reproduce both the finely-structured stochastic nature of the plume edges and a high-concentration contour (outlined in black) that is also evident in the original plume (Figure 14d). The reconstruction based on the moderate-detection-limit sensor network (Figure 14b) can also reproduce the general shape of the original plume. The reconstructed composite plume in Figure 14b is also broader than the original and includes the “alternate” source to the north of the real source, potentially providing misleading information. Finally, the reconstruction using the high-detection limit sensors, shown in Figure 14c, provides minimal useful information to a decision-maker, failing to include the real source location or identify a high-concentration-level contour (as represented by the black line seen in Figure 14a, b, or d). The only information provided in Figure 14c is that a release may have happened somewhere in the domain of interest and that it affected the two downwind sensors at this timestep; no resolution of the features of the plume is possible.

The composite plume can be very useful for a decision-maker seeking to determine how to utilize sensors to protect an asset. Numerical experiments such as these, which incorporate sensors with different characteristics (detection thresholds, time resolution, false-alarm frequency, and expense), along with likely release scenarios, can quantitatively illustrate what information different types of networks would provide in the case of an atmospheric release of hazardous materials. In the detection-limit experiment shown above, the decision-maker would weigh the utility of defining a high-concentration contour (outlined in black in Figure 14, which could correspond with a Protective Action Guideline), available from the low-detection-limit and moderate-detection-limit networks, against the higher cost of the low-detection-limit network.

5. Conclusions

Sensor networks must be designed to provide enough quantitative information about an event to reduce uncertainty in emergency response. We have demonstrated the application of an event reconstruction technology system to designing sensor networks for detecting and responding to atmospheric releases of hazardous materials. This system, developed at Lawrence Livermore National Laboratory, integrates observations with transport and dispersion models via Bayesian stochastic sampling methodologies to characterize the sources of atmospheric releases of hazardous materials. The event reconstruction methodology identifies source characteristics that are most consistent with the observed data, and then can provide an ultimate prediction, or composite plume, describing likely affected areas. This ultimate prediction is a composite of all the likely sources. In a real event, it can guide emergency responders more effectively than a single forward prediction from a single estimate of source characteristics as it provides a measure of uncertainty in source characterization and the result of the release of hazardous material. Before an event, this quantitative measure of the reduction in uncertainty, or benefit of a given network, can be utilized by policy makers to determine the cost/benefit of various networks.

Herein we present two numerical experiments demonstrating the utility of the event reconstruction methodology for sensor network design. Both experiments are loosely based on the Copenhagen tracer experiment (Gryning, 1981; Gryning and Lyck, 1984), but numerical sensor network design experiments require only climatological weather data, a dispersion model, and specifications of the types of sensors being considered – no actual tracer experiment data is required. In the present study, data from each network was provided to the event reconstruction system in order to identify the location and magnitude of a 1.5-hour release of a neutrally-buoyant gas in a 6km x 6 km suburban area.

In the first set of experiments, only the time resolution of the sensors varies between the three reconstructions. The most "expensive" sensor network, which provided data every five minutes (as compared to every ten minutes – "moderately-priced" or every sixty minutes – "inexpensive" – over this 1.5-hour release), offers only a few advantages over the moderately-priced network when attempting to reconstruct the location of the source explored here. Utilizing data from either the "moderate" or the "expensive" network, the event reconstruction methodology could identify the source location of the release within 5km, and could identify the magnitude of the source within 25%.

The second set of experiments presented herein explore the significance of the sensors' detection limit, which can have a significant impact on sensor cost. All sensors report data every ten minutes. The "expensive" network had a very low detection limit, and could distinguish data within a range of six orders of magnitude. The "moderate" network could identify data within a range of three orders of magnitude, while the "inexpensive" network could identify data within one order of magnitude. The upper limit, or saturation level, of the instruments in all three networks was identical. In this set of experiments, the expensive network can most clearly define the source location and source release rate. The other networks provide data insufficient for distinguishing between two possible clusters of source locations. When the reconstructions from all networks are aggregated into a composite plume, a decision-maker can distinguish the network that best suits needs. Reconstructions from both the expensive network and the moderately-priced network can reproduce certain high-threshold contours of atmospheric concentrations from the release considered here.

However, because of the limited sensitivity of the moderately-priced network, reconstruction from that network incorrectly predicts effects of the release in regions that would not be affected. A decision-maker could thus weigh the potential false-positive risk against the cost savings of that network.

The experiments presented herein have explored only a single type of release and a single meteorological scenario, in order to demonstrate the application of event reconstruction to sensor network design. More complete sensor network studies would consider multiple climatological conditions (i.e. wind speed, wind direction, atmospheric stability) representative of the region of interest. A range of possible source magnitudes may also be explored to ensure that the network would provide useful composite plumes in most likely scenarios.

6. References

Chan, S. and D. Stevens, 2000, An Evaluation of Two Advanced Turbulence Models for Simulating the Flow and Dispersion Around Buildings, The Millennium NATO/CCMS Int. Tech. Meeting on Air Pollution Modeling and its Application, Boulder, CO, May 2000, 355-362.

Chow, K. K., B. Kosović, and S. T. Chan, 2006: Source Inversion for Contaminant Plume Dispersion in Urban Environments using Building-Resolving Simulations, American Meteorological Society's 6th Symposium on the Urban Environment, Atlanta, Georgia, Jan 29 – Feb 2, 2006.

Ermak, D.L., and J.S. Nasstrom, 2000: A Lagrangian Stochastic Diffusion Method for Inhomogeneous Turbulence, *Atmos. Environ*., 34, 7, 1059-1068.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin, 2004: *Bayesian Data Analaysis, Second Edition*. Chapman & Hall/CRC. 668 pp.

Gryning, S.-E., 1981: *Elevated Source SF6-Tracer Dispersion Experiments in the Copenhagen Area*. Risø-R-446. Risø National Laboratory, 187pp.

Gryning, S.-E. and E. Lyck,1984: Atmospheric Dispersion from Elevated Sources in an Urban Area: Comparison between Tracer Experiments and Model Calculations. *J. Clim. Appl. Meteorol.*, **23**, 651-660.

Kosović, B., and co-authors, 2005: Stochastic Source Inversion Methodology and Optimal Sensor Network Design. 9th Annual George Mason University Conference on "Atmospheric Transport and Dispersion Modeling", Fairfax, VA, July 18-20, 2005. UCRL-PRES-213633.

Larson, D.J., and J.S. Nasstrom, 2002: Shared and Distributed Memory Parallelization of a Lagrangian Atmospheric Dispersion Model, *Atmos. Environ*., **36**, 1559-1564.

Liu, J. S., 2001: *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics, 343 pp.

Robert, C. P. and G. Casella, 2005: *Monte Carlo Statistical Methods, Second Edition*. Springer Texts in Statistics, 645 pp.

Petersen, W.B., and Lavdas L.G., 1986. INPUFF 2.0 - A Multiple Source Gaussian Puff Dispersion Algorithm. User's Guide. U.S. E.P.A., Research Triangle Park, NC.

Shea, D. A. and S. A. Lister, 2003: The BioWatch Program: Detection of Bioterrorism. Congressional Research Service Report No. RL 32152. Available at http://www.fas.org/sgp/crs/terror/RL32152.html, downloaded 12/15/2005.

Sugiyama, G., and S.T. Chan, 1998. A new meteorological data assimilation model for real-time emergency response. *10th Joint Conference on the Applications of Air Pollution Meteorology*, Phoenix, AZ, Jan. 11-16, 1998, American Meteorological Society, Boston, MA. 285-289.

7. Acknowledgements

Appreciation is extended to the entire Event Reconstruction team at LLNL, including Chuck Baldwin, Stevens T. Chan, Fotini K. Chow, Kathleen Dyer, Ron Glaser, William Hanley, Gardar Johannesson, Milovan Krnjajić, Shawn Larsen, Gwen Loosmore, Arthur Mirin, Stephanie Neuman, John Nitao, Radu Serban, Alan Sicherman, Gayle Sugiyama, and Charles Tong. The authors also appreciate discussions with Roger Aines and Jeffrey Mirocha.

8. Figures

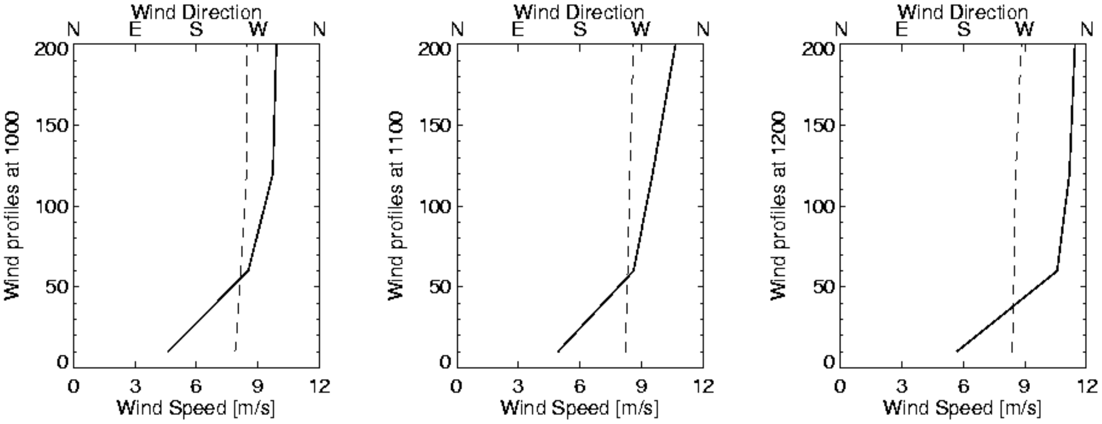


Figure 1: Vertical profiles of wind speed (solid line) and wind direction (dashed line) for the three hours relevant to these simulations

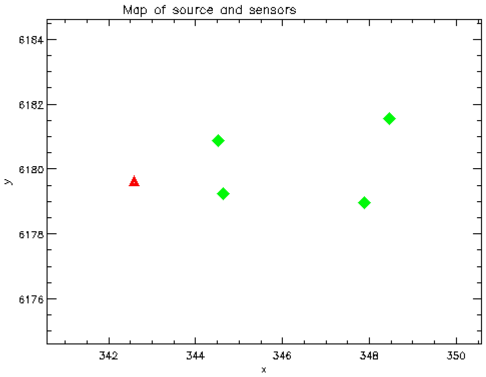
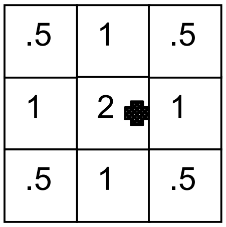


Figure 2: The source (red triangle) and sensors (green diamonds) used in this study. The winds, as noted in Figure 1, are from the west.



.5	1	.5
1	2	1
.5	1	.5

Figure 3: Weighting of concentrations calculated for cells including and around a sensor (indicated with cross).

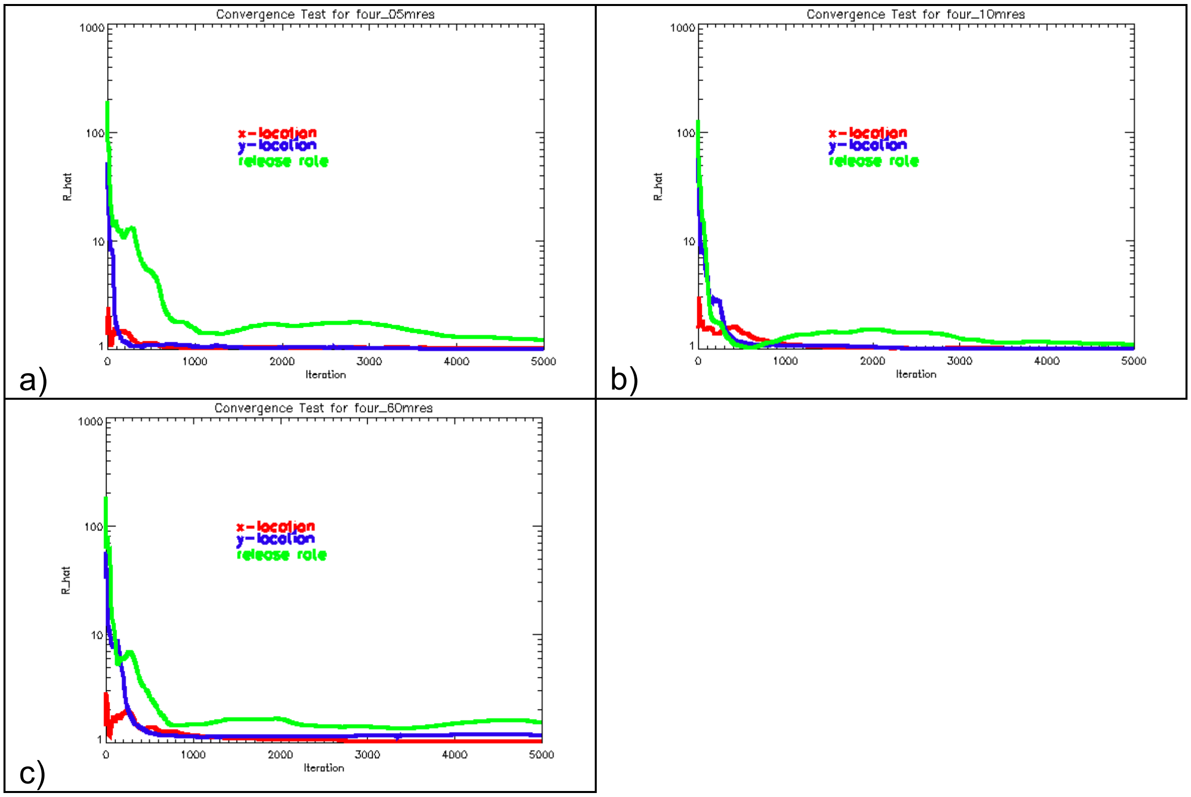


Figure 4: Convergence metrics for the time-resolution study using a) the five-minute network, b) the ten-minute network, c) the 60-minute network

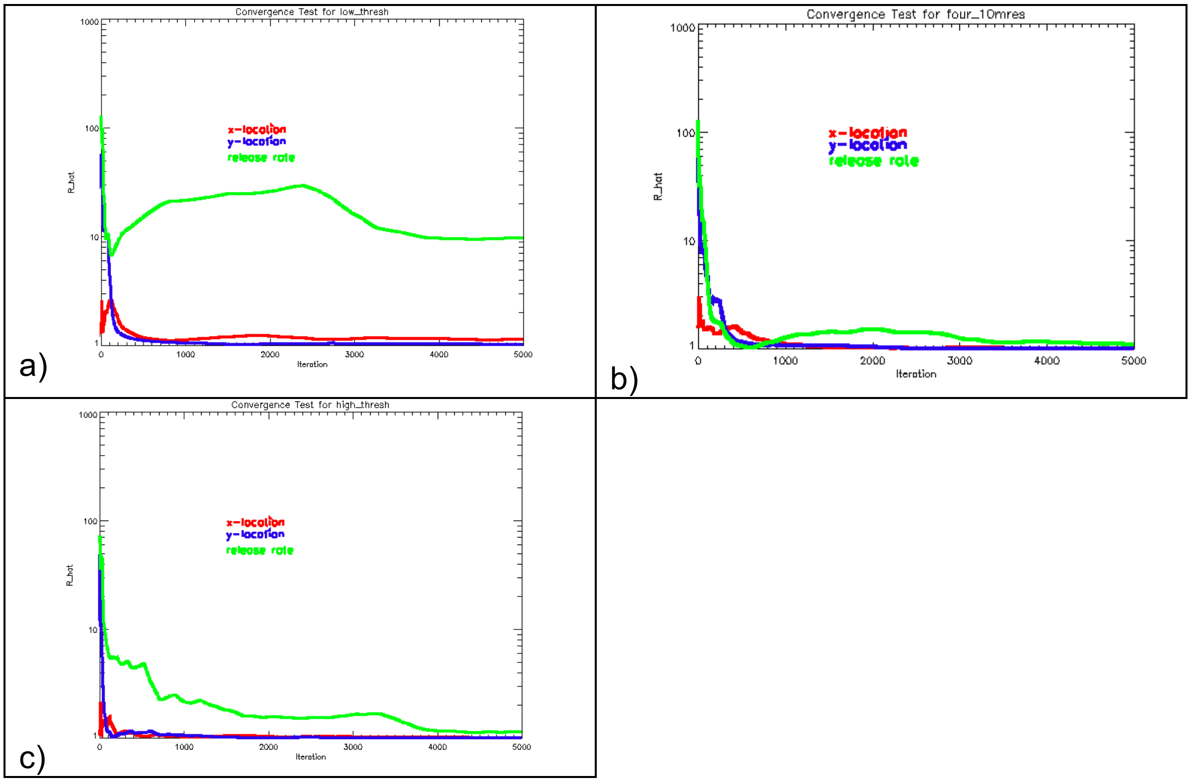


Figure 5: Convergence metrics for the sensor-sensitivity study using a) the low-threshold network, b) the moderate-threshold network, c) the high-threshold network

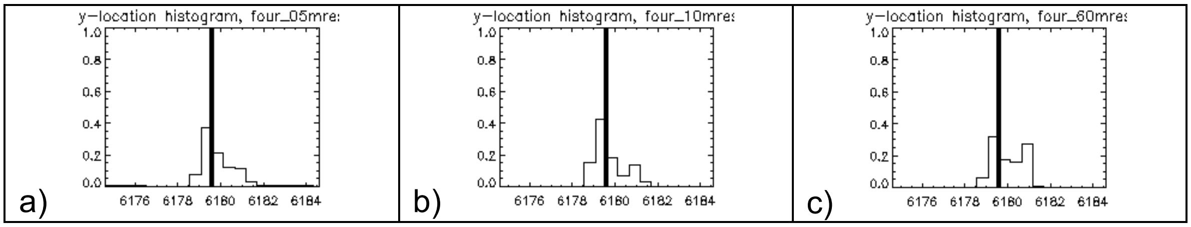


Figure 6: Histograms for y-location for simulations with a) 5-minute b) 10-minute, c) 60-minute resolution sensors.

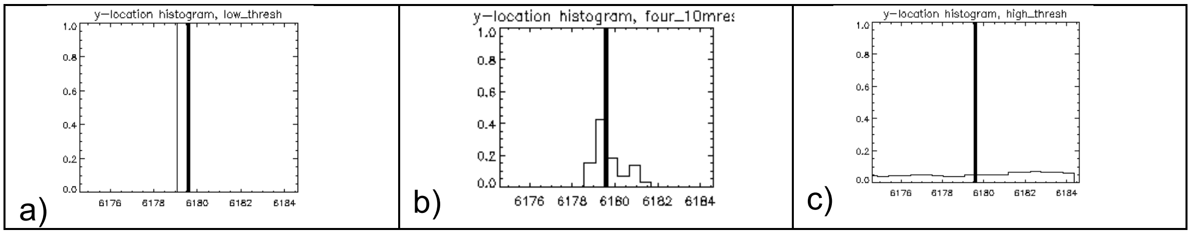


Figure 7: Histograms for y-location for simulations with a) low-threshold, b) moderate-threshold, and c) high-threshold sensors.

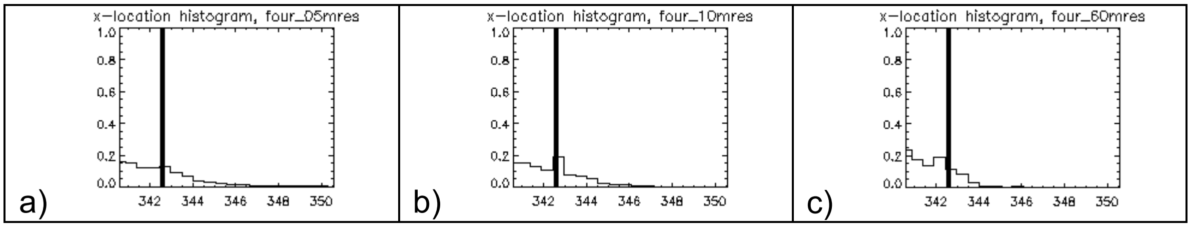


Figure 8: Histograms for x-location for simulations with a) 5-minute b) 10-minute, c) 60-minute resolution sensors.

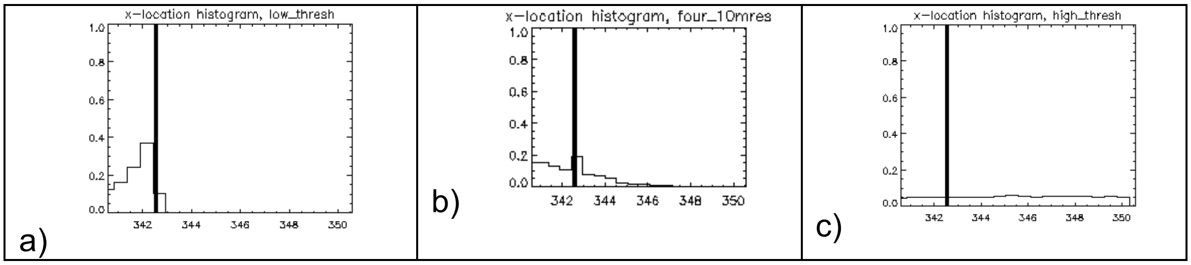


Figure 9: Histograms for x-location for simulations with a) low-threshold, b) moderate-threshold, and c) high-threshold sensors.

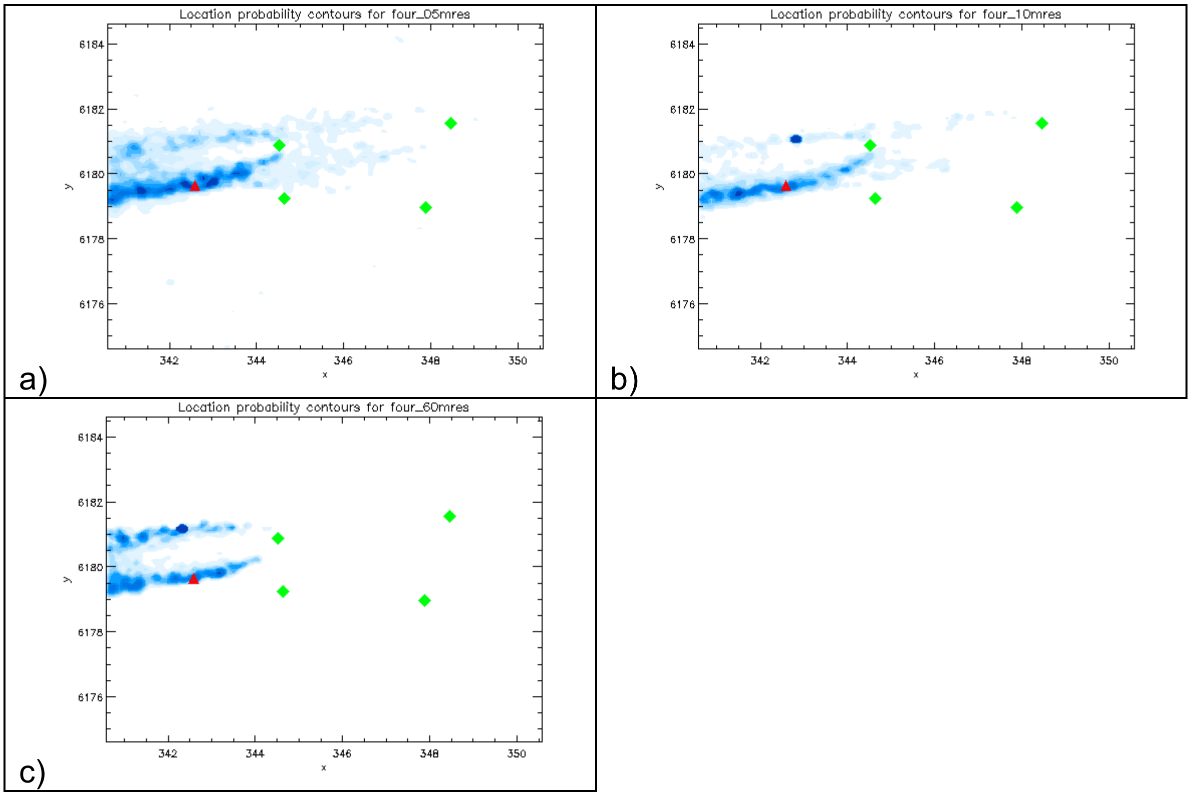


Figure 10: Joint histogram for x and y location for simulation with sensors with a) 5-minute, b) 10-minute, and c) 60-minute time resolution

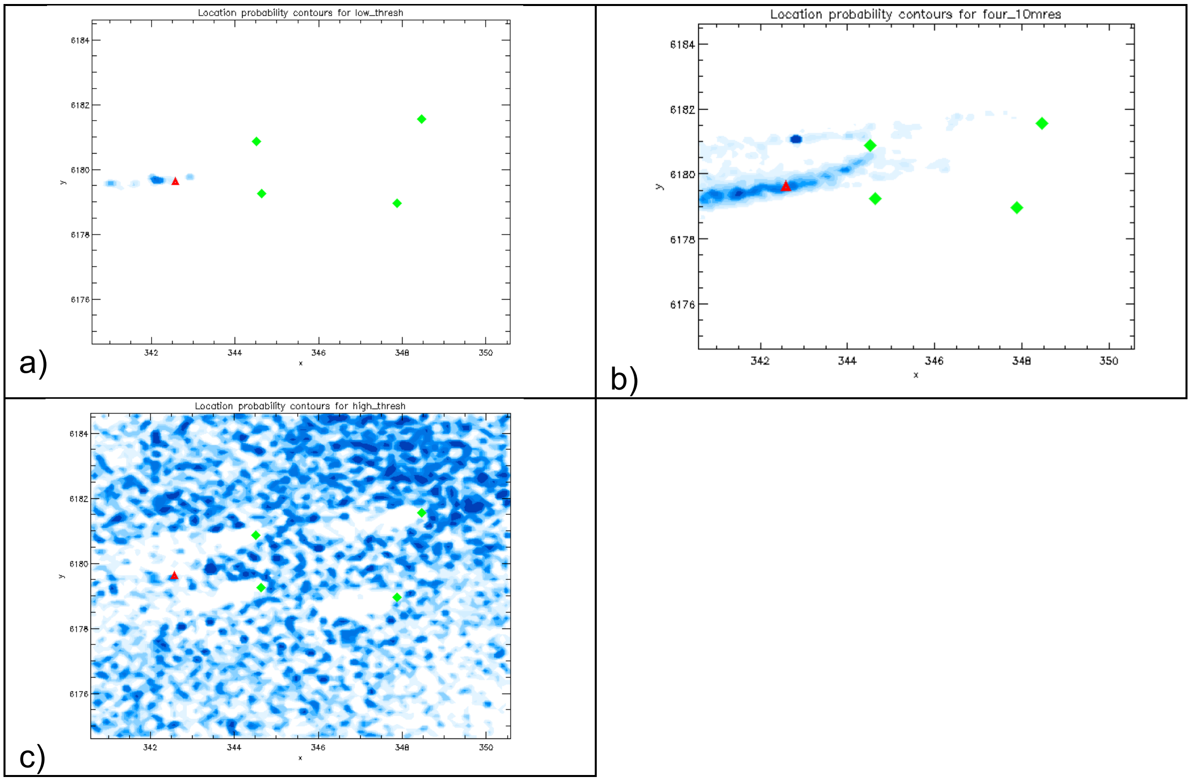


Figure 11: Joint histogram for x and y location for simulation with sensors with a) low, b) moderate, and c) high thresholds. Darker blues indicate higher probability locations.

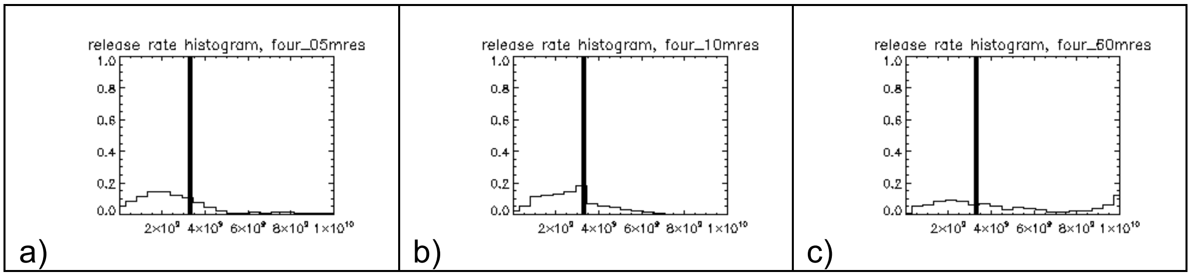


Figure 12: Histogram for release rate for simulation with sensors with a) 5-minute, b) 10-minute, and c) 60-minute time resolution

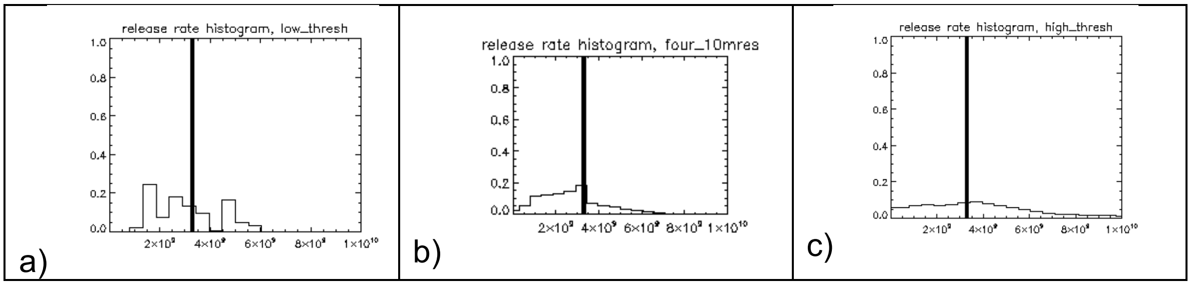


Figure 13: Histogram for release rate for simulation with sensors with a) low, b) moderate, and c) high detection limits

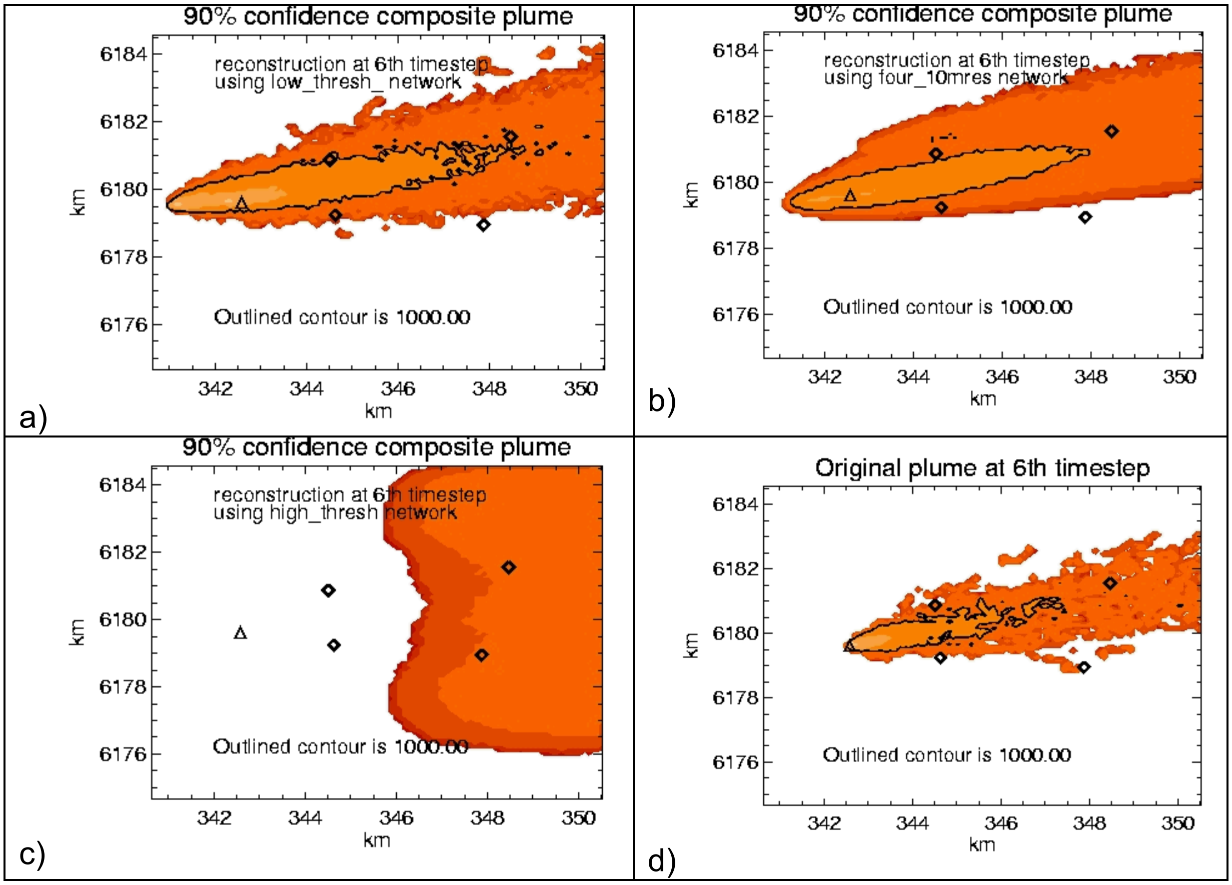


Figure 14: Composite plumes with logarithmic color contours, generated from a) the low-detection level , b) the moderate-detection level, and c) the high-detection level sensor networks. Colored contour levels indicate the magnitude of atmospheric concentrations in which a decision-maker can have 90% confidence. Highest concentrations are in yellow and light orange; lower concentrations are in red and dark red. Note that the reconstruction with the high-detection-level network c) provides confidence only that half of the domain will experience concentrations above a low level. Plume d) represents the original plume, from which the data for the reconstruction were generated. Note that both the reconstructions using the low-detection limit instruments, a), and the moderate-detection limit instruments, b), can reproduce the high-concentration contour of 1000 ng/m3, outlined in a), b), and d) with a solid black line. Note also in b) that the plume extends further in the north-south direction to encompass the probability that the source might be at a second alternate location to the north of the real location.

9. Appendix A – adapt.nml, observ.met, stnloc.met files for the meteorology used in this example

a. adapt.nml

```
 *** adapt.nml automatically generated on 19-Aug-2004 08:53:44 via code written by
Michael Dillon on 5-21-04


 &adapt_control
  flag_debug   = .true.
 /

 &adapt_grid
  file_met_grid     = "../main_grd_copenhagen.nc"
  opt_grid_file     = "gridgen"
 /

 *** Beginning adapt parameters for met time1979JUL19_100000

 &adapt_metdata
  file_met_field     = "met_field_1979JUL19_100000.nc"
  opt_src_obs        = "ascii2"
  opt_src_field      = "none"
  file_src_obs       = "observ.met"
  file_src_station   = "../stnloc.met"
  flag_station_km    = .true.
  nmethod            = 2
 /

 &adapt_field2D
  hgt_vert_coord         = "zAGL"
  hgt_boundary_layer     = 2090
  hgt_geostrophic_layer  = 2090
  z0                     = 0.6000000
  inv_monin_obukhov_len  = -2.617801e-003
  friction_velocity      = 7.700000e-001
 /

 &adapt_method
  opt_wind_horz              = "spddir"
  flag_use_missing_wind = .true.
  opt_met_type          = "wind2d"
  obs_date_time         = "1979JUL19_100000"
  blend_exp             = 0.100000
  flag_upr_in_sl        = .false.
  flag_twr_local_only   = .false.
  flag_mc_adjust        = .false.
  met_x_border          = 100000.0
  met_y_border          = 100000.0
  blend_max_veer        = 180
  sl_pwr_exp            = 0.1900000
 /

 &adapt_mc_adjust
 /

 &adapt_method
  opt_method = "turb"
 /

 &adapt_turbulence
  sigmav_tavg       = 3600
  sigmav_tavgo      = 3600
  sigmav_t_lagran_h = 400
  turb_param_h      = "sigmav_simthry"
  sigmav            = 1.71
  sigmav_meas_hgt   = 115
  turb_param_z      = "simthry"
  sim_kz_c          = 4
  sim_kz_trop       = 0.01
 /

 *** Beginning adapt parameters for met time1979JUL19_110000

 &adapt_metdata
  file_met_field     = "met_field_1979JUL19_110000.nc"
  opt_src_obs        = "ascii2"
  opt_src_field      = "none"
  file_src_obs       = "observ.met"
  file_src_station   = "../stnloc.met"
  flag_station_km    = .true.
  nmethod            = 2
 /

 &adapt_field2D
  hgt_vert_coord         = "zAGL"
  hgt_boundary_layer     = 2090
  hgt_geostrophic_layer  = 2090
  z0                     = 0.6000000
  inv_monin_obukhov_len  = -2.617801e-003
  friction_velocity      = 7.700000e-001
 /

 &adapt_method
  opt_wind_horz              = "spddir"
  flag_use_missing_wind = .true.
  opt_met_type          = "wind2d"
  obs_date_time         = "1979JUL19_110000"
  blend_exp             = 0.100000
  flag_upr_in_sl        = .false.
  flag_twr_local_only   = .false.
  flag_mc_adjust        = .false.
  met_x_border          = 100000.0
  met_y_border          = 100000.0
  blend_max_veer        = 180
  sl_pwr_exp            = 0.1900000
 /

 &adapt_mc_adjust
 /

 &adapt_method
  opt_method = "turb"
 /

 &adapt_turbulence
  sigmav_tavg       = 3600
  sigmav_tavgo      = 3600
  sigmav_t_lagran_h = 400
  turb_param_h      = "sigmav_simthry"
  sigmav            = 1.71
  sigmav_meas_hgt   = 115
  turb_param_z      = "simthry"
  sim_kz_c          = 4
  sim_kz_trop       = 0.01
 /

 *** Beginning adapt parameters for met time1979JUL19_120000

 &adapt_metdata
  file_met_field     = "met_field_1979JUL19_120000.nc"
  opt_src_obs        = "ascii2"
  opt_src_field      = "none"
  file_src_obs       = "observ.met"
  file_src_station   = "../stnloc.met"
  flag_station_km    = .true.
  nmethod            = 2
 /
```

```
&adapt_field2D
 hgt_vert_coord         = "zAGL"
 hgt_boundary_layer     = 2090
 hgt_geostrophic_layer  = 2090
 z0                     = 0.6000000
 inv_monin_obukhov_len  = -2.617801e-003
 friction_velocity      = 7.700000e-001
/

&adapt_method
 opt_wind_horz              = "spddir"
 flag_use_missing_wind = .true.
 opt_met_type          = "wind2d"
 obs_date_time         = "1979JUL19_120000"
 blend_exp             = 0.100000
 flag_upr_in_sl        = .false.
 flag_twr_local_only   = .false.
 flag_mc_adjust        = .false.
 met_x_border          = 100000.0
 met_y_border          = 100000.0
 blend_max_veer        = 180
 sl_pwr_exp            = 0.1900000
/

&adapt_mc_adjust
/

&adapt_method
 opt_method = "turb"
/

&adapt_turbulence
 sigmav_tavg       = 3600
 sigmav_tavgo      = 3600
 sigmav_t_lagran_h = 400
 turb_param_h      = "sigmav_simthry"
 sigmav            = 1.71
 sigmav_meas_hgt   = 115
 turb_param_z      = "simthry"
 sim_kz_c          = 4
 sim_kz_trop       = 0.01
/

*** Beginning adapt parameters for met time1979JUL19_130000

&adapt_metdata
 file_met_field     = "met_field_1979JUL19_130000.nc"
 opt_src_obs        = "ascii2"
 opt_src_field      = "none"
 file_src_obs       = "observ.met"
 file_src_station   = "../stnloc.met"
 flag_station_km    = .true.
 nmethod            = 2
/

&adapt_field2D
 hgt_vert_coord         = "zAGL"
 hgt_boundary_layer     = 2090
 hgt_geostrophic_layer  = 2090
 z0                     = 0.6000000
 inv_monin_obukhov_len  = -2.617801e-003
 friction_velocity      = 7.700000e-001
/

&adapt_method
 opt_wind_horz              = "spddir"
 flag_use_missing_wind = .true.
 opt_met_type          = "wind2d"
 obs_date_time         = "1979JUL19_130000"
 blend_exp             = 0.100000
 flag_upr_in_sl        = .false.
```

```
  flag_twr_local_only   = .false.
  flag_mc_adjust        = .false.
  met_x_border          = 100000.0
  met_y_border          = 100000.0
  blend_max_veer        = 180
  sl_pwr_exp            = 0.1900000
 /

 &adapt_mc_adjust
 /

 &adapt_method
  opt_method = "turb"
 /

 &adapt_turbulence
  sigmav_tavg       = 3600
  sigmav_tavgo      = 3600
  sigmav_t_lagran_h = 400
  turb_param_h      = "sigmav_simthry"
  sigmav            = 1.71
  sigmav_meas_hgt   = 115
  turb_param_z      = "simthry"
  sim_kz_c          = 4
  sim_kz_trop       = 0.01
 /
```

b. observ.met

```
METDATASET '1979JUL19_100000'
SFC
'TV TWR'     236.7     4.60
'TV TWR2'     -1.0     8.53
'TV TWR3'     253.3     9.73
'TV TWR4'     253.3     9.90
UPR
'TV TWR'  60    -1.0     8.53
'TV TWR'  120   253.3     9.73
'TV TWR'  200   253.3     9.90


METDATASET '1979JUL19_110000'
SFC
'TV TWR'     246.7     4.93
'TV TWR2'     -1.0     8.62
'TV TWR3'     253.3     9.55
'TV TWR4'     258.3     10.67
UPR
'TV TWR'  60    -1.0     8.62
'TV TWR'  120   253.3     9.55
'TV TWR'  200   258.3     10.67


METDATASET '1979JUL19_120000'
SFC
'TV TWR'     251.7     5.68
'TV TWR2'     -1.0     10.58
'TV TWR3'     256.7     11.18
'TV TWR4'     265.0     11.45
UPR
'TV TWR'  60    -1.0     10.58
'TV TWR'  120   256.7     11.18
'TV TWR'  200   265.0     11.45


METDATASET '1979JUL19_130000'
SFC
'TV TWR'     240.0     5.60
'TV TWR2'     -1.0     9.70
'TV TWR3'     260.0     10.80
```

```
'TV TWR4'     260.0     10.80
UPR
'TV TWR'  60    -1.0     9.70
'TV TWR'  120   260.0     10.80
'TV TWR'  200   260.0     10.80
```

c. stnloc.met

```
SFC
'TV TWR'      342.580     6179.610     10
'TV TWR2'     342.580     6179.610     60
'TV TWR3'     342.580     6179.610     120
'TV TWR4'     342.580     6179.610     200
UPR
'TV TWR'      342.580     6179.610
```

10. Appendix B – the .pyin file for the 60m resolution reconstruction

```
# File: four_60mres.pyin
#
# Input file for mcmc_app_copenhagen, generated automatically by Julie
#
################################################################################
# Tells python to look for .py files in the current (working) directory.
import os

from mcmc_app.mcmc_drivers import make_target_sample
from mcmc_app.misc         import make_proc_grp
from mcmc_app.seedmaker    import SeedMaker

# Maximum number of iterations
itermax = 5000

# Number of iterations for burn-in (used only for postprocessing
# or convergence monitoring purposes)
burn_in = 200

# Number of independent sequences
num_seqs = 4

# Number of processors per forward model
# ??? Problem: Allow this to be set to not-1, but degrade gracefully
# ??? in serial mode.
num_procs_per_mod = 32

# Number of processors per sequence
# (total number of processors will be equal to num_seqs times this value.
# (note: for mpi job, r.h.s. must be integer because it is parsed by mpi script)
num_procs_per_seq = 1

# Processor group for mpi jobs; can be set to None for non-mpi jobs
#proc_grp = main_driver.make_proc_grp(num_seqs, num_procs_per_seq)
proc_grp = make_proc_grp(num_seqs, num_procs_per_seq)


# Seed generator -----------------------
# Creates different seeds for different chains, even if they are running
# on different processors
seed = 38895
seed_generator = SeedMaker(seed, itermax, num_seqs, proc_grp=proc_grp)

# MCMC algorithm -----------------------

synthetic_data = {
    # required input:
    'class_name'  : 'LODI_mcmc_et.sampler.ExampleSampler',
    # application specific input:
```

```
    'step_size_xy' : 1.0,
    'step_size_z' : 1.0,
    'step_size_q' : 1.0,
    # for this example the base state consists of (x,y) with
    # that are gaussian with following means and sigma's
    'x_mean' : 342580.0,
    'x_sigma' : 1.,
    'x_min' : 342577.0,
    'x_max' : 342582.0,
    'y_mean' : 6179610.0,
    'y_sigma' : 1.,
    'y_min' : 6179607.0, #4704237.0,
    'y_max' : 6179612.0, #4705417.0,
    'q_mean' : 3.3e+09,
    'q_sigma' : 1.e+07,
    'q_min' : 3.28e+09, #0.07779,
    'q_max' : 3.32e+09 #0.07781
}


base_sampler_input = {
    # required input:
    'class_name'  : 'LODI_mcmc_et.sampler.ExampleSampler',
    # application specific input:
    'step_size_xy' : 0.1,
    'step_size_q' : 0.1,
    # for this example the base state consists of (x,y) with
    # that are gaussian with following means and sigmas
    'x_mean' : 345580.0,
    'x_sigma' : 10000.,
    'x_min' : 340580.0,
    'x_max' : 350580.0,
    'y_mean' : 6179610.0,
    'y_sigma' : 10000.,
    'y_min' : 6174610.0,
    'y_max' : 6184610.0,
    'q_mean' : 3.2e+09,
    'q_sigma' : 3.e+09,
    'q_min' : 1.0e+4,
    'q_max' : 1.0e+15
}

# Make the target state (synthetic truth) using the base sampler
# (not needed if not used by log_like_fun_input, below)
target_sample = make_target_sample(synthetic_data, seed_generator)

class LODINmlTemplate :
    def __init__(self) :

        self.template_lines = (
      '&prob_setup',
      '    title           = " Copenhagen Experiment - IOP10 " ',
      '    tstart_str      = "1979JUL19_103800"',
      '    tstop_str       = "1979JUL19_123800"',
      '    dt_part_str     = "02:00:00"',
      '    nbins           = 1',
      '    nsrc            = 1',
      '    num_met_times   = 3',
      "    met_time_strs = ",
               '              "1979JUL19_100000"',
               '              "1979JUL19_110000"',
               '              "1979JUL19_120000"',
      '     dt_dump_str    = "0::0:0:0"',
      '     dt_min         = 0',
      '     dt_fact_adv    = 1',
      '     dt_fact_dif    = 1',
      '     dt_limit       = 3600',
      '     dz_dep         = 20',
      '     met_format     = "arac"',
      '     out_bin_ascii  = .false.',
      '     out_part_ascii = .false.',
      '     rd_grid        = "gridgen"',
      '     rdm_dist       = "nongauss"',
      '     reflect        = "vertical"',
      '     solver_id      = "rk2"',
      ' /',
      '',
      '&thist_param',
      ' /',
      '',
      '&src_param',
      '    source_id      = "Source  1"',
      '    max_num_part   = 10000',
      '    species        = "SF6"',
      '    mass_distrib   = "table"',
      '    m_bin_fract    = 1.0',
      '    m_bin_diam_max = 0.0',
      '    m_bin_diam_min = 0.0',
      '    nset_dep_vel   = 0.0000000E+00',
      '    geom_time_strs = "1979JUL19_105000"',
      '    geom_type      = 2  ',
      'key_x_pt', #'    mean_x         =    342580.0',
      'key_y_pt', #    mean_y         =    6179610.',
      '    std_x          =    1.000000',
      '    std_y          =    1.000000',
      '    cutoff_dx_min  =    2.500000',
      '    cutoff_dy_min  =    2.500000',
      '    cutoff_dx_max  =    2.500000',
      '    cutoff_dy_max  =    2.500000',
      '    mean_z         =    10.00000',
      '    std_z          =    1.000000',
      '    cutoff_dz_min  =    2.500000',
      '    cutoff_dz_max  =    2.500000',
      '    er_time_strs   = "1979JUL19_105000      1979JUL19_122000"',
      'key_emiss_rates', #    emiss_rates    =    3.200000e+009      0.0000000E+00',
      '    er_units_type   = "mass"',
      '    decay_param     = "none"',
      '    half_life       = 0.0',
      '    lambda          = 0.0',
      '    decay_chain     = .false.',
      '    start_time_str  = "1979JUL19_105000"',
      '    stop_time_str   = "1979JUL19_122000"',
      '    dt_hold_str     = "0::0:0:0"',
      '    source_model    = "neutral"',
      '    src_generation_method = "new"',
      '    src_agl_flg     = .true.',
      ' /',
      '',
      '&bin_param',
      '    bin_id         = "Bin  1"',
      '    samp_type      = "average"',
      '    type           = "air"',
      '    orientation    = "xy"',
      '    bin_agl_flg    = .true.',
      '    position       = 10.0',
      '    width          = 20.0',
      '    dt_samp_str    = "0::01:00:00"',
      '    dt_bin_out_str = "0::01:00:00"',
      '    source_list    = "Source  1"',
      '    species_name   = "SF6"',
      ' /',
      '',
      '&turb_param',
      '    read_adapt_turb = .true.',
      ' /',
      '',
      '&met_param',
      ' /  ',
         '',
         '',
         '',
        )
```

```
    def create_nml(self, state, output_file_name) :
        # Create the LODI nml file, using a template and the new state
        # information.
        file_out = open(output_file_name, 'w')

        q=state.sampler_data['q']
        for line in self.template_lines:
            if line == 'key_x_pt':
                print >>file_out, '   mean_x =', state.sampler_data['x']
            elif line == 'key_y_pt':
                print >>file_out, '   mean_y =', state.sampler_data['y']
            elif line == 'key_emiss_rates':
                print >>file_out, '   emiss_rates =', state.sampler_data['q']
            else:
                print >>file_out, line

        file_out.flush()
        os.fsync(file_out.fileno())
        file_out.close

        return


LODI_nml_template = LODINmlTemplate()

# Lines to put in LODI_files.nml file
LODI_files_dir = os.getcwd() #+ '/../experiment5' #jkl

LODI_files_nml = (
    "&grid_name",
    "   num_m_grids = 1",
    "   m_grid_name = '" + LODI_files_dir + "/grid/main_grd_copenhagen.nc'",
    "   c_grid_name = '" + LODI_files_dir + "/grid/conc_grd_copenhagen.nc'",
    "/",
    "",
    "&metfiles",
    "   grid_num = 1",
    "   met_file_name = ",
    '             "' + LODI_files_dir + '/iop10/met_field_1979JUL19_100000.nc"',
    '             "' + LODI_files_dir + '/iop10/met_field_1979JUL19_110000.nc"',
    '             "' + LODI_files_dir + '/iop10/met_field_1979JUL19_120000.nc"',
    "/",
    "",
    "&decay_chains_file",
    '   decay_chains_file_name = "decaychains.dat"',
    "/",
    "",
    "",
    "",
)


# Likelihood function -----------------------
# there needs to be one likelihood function for each stage; for this
# example we have only a single stage
log_like_fun_1 = {
    # required input:
    'class_name'       : 'LODI_mcmc_et.likefun.LogLikeFunA',

    # following is information necessary for parallelization
    'num_seqs'          : num_seqs,
    'num_procs_per_mod' : num_procs_per_mod,

    # following is set for random synthetic truth measurements
    'target_sample'     : target_sample,

    # following is set for random synthetic truth measurements
    #'measurement_data'   : data,
    # the iteration at which the likelihood function is actually turned on
    # (for single stage, this should be set to 1; otherwise during staging
    # the likelihood is ignored until the iteration hits the following value)
    'start_iter'      : 1,

    # Settings for model_driver
    'LODI_files_nml' : LODI_files_nml,  # Lines to put in LODI_files.nml file
    'LODI_nml_template' : LODI_nml_template,

    'sensors' : [
                  [344629., 6179248., 3600.  ], # Arc 1-22
#                  [344607., 6180387., 3600.  ], # Arc 1-33
                  [344509., 6180871., 3600.  ], # Arc 1-38
#                  [346559., 6179040., 3600.  ], # Arc 2-23
#                  [346412., 6181080., 3600.  ], # Arc 2-33
#                  [345961., 6181562., 3600.  ], # Arc 2-36
                  [347873., 6178952., 3600.  ], # Arc 3-23
                  [348468., 6181563., 3600.  ], # Arc 3-32
#                  [347792., 6182526., 3600.  ], # Arc 3-36
                  [344629., 6179248., 7200.  ], # Arc 1-22
#                  [344607., 6180387., 7200.  ], # Arc 1-33
                  [344509., 6180871., 7200.  ], # Arc 1-38
#                  [346559., 6179040., 7200.  ], # Arc 2-23
#                  [346412., 6181080., 7200.  ], # Arc 2-33
#                  [345961., 6181562., 7200.  ], # Arc 2-36
                  [347873., 6178952., 7200.  ], # Arc 3-23
                  [348468., 6181563., 7200.  ] # Arc 3-32
#                  [347792., 6182526., 7200.  ]  # Arc 3-36
                 ], # Arc 3-36

    # application specific input (e.g. likelihood function parameters)
    'param_a' : 1.0,
    'param_b' : 0.0,
    'lowerbound' : +1,
    'upperbound' : +4,
    'sigma_rel' : 0.2,
    'option' : 1,
}

log_like_fun_input = [ log_like_fun_1 ]

# Datadumps, plots, monitoring --------------------

state_out = {
    'class_name'    : 'LODI_mcmc_et.dumpers.DumpTextC',

    'burn_in'       : burn_in,
    'starting_iter' : 0,
    'single_file'   : 1,   # Dump all output for a sequence to one file.
}
states_out = [state_out]

# Restart -----------------------

# Settings to write out restart files.
restart_write = {
    'class_name'    : 'mcmc_app.outputs.RestartOutput',
    'starting_iter' : 1000,
}

# Settings to read restart files.
restart_read = {
    # If following is present and is not 0 or None, use initial state from
    # restart file
    'use_restart_new'      : 1,
    'restart_iter_new'     : 4427,       # Iteration for reading restart
}
```