
# Mobile Robots with Novel Environmental Sensors for Inspection of Disaster Sites with Low Visibility

Project start: January 1, 2015

Duration: 3.5 years

## Deliverable 5.1

## Algorithms for Sensor Planning

Due date: month 36 (January 2018)

Lead beneficiary: ORU

## Dissemination Level: Public

## Main Authors:

Victor Hernandez Bennetts (ORU)
Asif Arain (ORU)
Erik Schaffernicht (ORU)
Achim Lilienthal (ORU)

## Version History:

0.2: initial version, VH, June, 2018
0.1: initial version, VH, May, 2018

## Contents

# A Introduction and purpose of this document

In this deliverable, algorithms are proposed to efficiently conduct an exploration mission to cover the target area as fast as possible while acquiring relevant information. For example, the robot prototype should identify interesting areas where emitting gas source locations might be located. In the Smokebot scenario, the exploration strategy (robot trajectory and stop positions) is decided by the human operator. Thus, the planning algorithms presented in this deliverable should be able to suggest positions of interest (POI). These POIs are places where the multiple sensor readings will be most informative to integrate into the GDIM, will help localizing a gas source or where dropping a wireless repeater will ensure stable communication with the operator. Two different types of problems are considered: coverage - building an environment model of the whole operation area as quickly as possible - and exploitation - observing a specified small area for detailed information gathering over time.

The POIs are presented to the human operator, which then decides where to move the robot based on the priority that each of the tasks has at a given point in a given mission. For example, the operator might decide to prioritize POIs related to gas sensing, and move the robot accordingly, if localizing gas leaks is a priority during the mission.

In the following sections, the different sensor planning algorithms developed in Smokebot, are presented. Based on the sensors' characteristics, the algorithms are grouped as follows:

**a) Sensor planning for in-situ gas sensors (Section B):** For example, suggesting interesting positions to localize emitting gas sources.

**b) Sensor planning for remote sensors (Section C):** For example, suggesting possitions for LIDAR sensors to minimize thermal reflection.

**c) Sensor planning for communication coverage (Section D):** For example, predicting areas with low RF signal coverage ahead of the robot.

# B Sensor Planning for In-situ gas sensors

In-situ gas sensors are devices that require direct contact with the analyte under study in order to report a measurement. This means that the reported measurement corresponds to only a few centimeters around the location of the sensor, which makes data acquisition time consuming. In order to efficiently conduct an exploration mission under limitations such as battery life (as in the case of Smokebot), a sensor planning strategy is required. We developed two different sensor planning strategies for in-situ gas sensors for the task of gas source localization. The first strategy is a model-free, data driven approach that creates a lattice where at each cell, an estimation of the distance to a gas source is stored. Areas that are predicted as being nearby the source can be considered as POI. The distance estimations at those cells not visited by the robot are computed using Gaussian regression guided by wind flow measurements. The second approach is model based. Specifically, it uses Partial Differential Equations for gas source localization. Constructing a realistic mathematical model of gas dispersion is complex and computationally expensive to solve. To address this problem, we propose a probabilistic model based on diffusion PDE to approximate the complexities of gas dispersion. The model suggest informative measurement locations as well as likely source locations. The proposed approaches were evaluated in real-world and with hardware-in-the-loop simulations respectively. The following sections present a deeper overview of the algorithms and the results achieved in the experimental validation processes.

## B.1 Model-free sensor planning approach

This algorithm represents the target environment M as a lattice of cells of identical size: M={x1,...,xN}. The algorithm assumes that only one gas source is present in the environment. Regarding on-board sensing modalities, the algorithm assumes that the robot is equipped with gas and airflow sensors and moreover, it assumes that the robot's position is available.

The robot moves between the centers of the cells. After each movement it records measurements of gas concentration and wind information for a certain amount of time in order to compute the source distance estimation. We define a function f(x) that is unknown a-priori and is updated from noisy observations. f(x) indicates the distance to the source at location. Using measurements at visited locations, the robot extrapolates f(x) to estimate non visited locations. The algorithm uses these estimations to decide where to measure next in order to improve the model efficiently. The overall process is computed online as the robot traverses the target area.

Fig. 1(a) shows the three functional blocks of our proposed model-free algorithm. In the first block, an estimation of the distance to the gas source is computed. We use the method proposed by Schmuker and co-authors in [1], which correlates the variability of the sensor response to the distance to the source. A key element to the algorithm proposed by the authors is the detection of bouts in the sensor response. These bouts correspond to the rising edges of the sensor signal, as shown in the example in Fig. 1(b). In [1], it is reported that there is a strong correlation between the number of bouts and the distance to the gas source: the higher the bout count, the closer the sensor to the gas source. To detect the bouts, a cascaded filtering approach is used to detect fast transients in the sensor signal. A low-pass filter is first used in order to remove high-frequency noise.

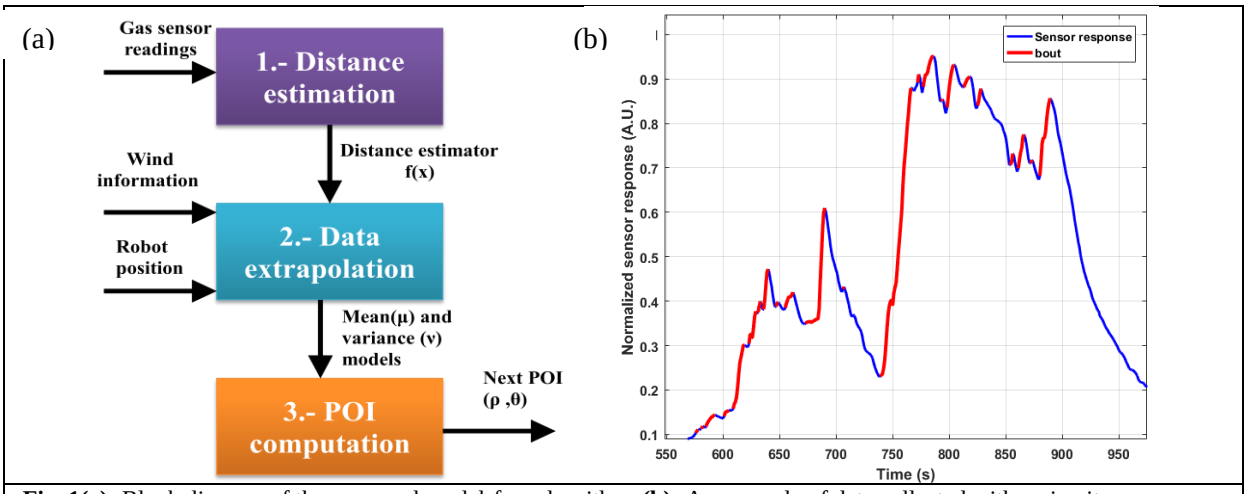



**Fig. 1(a):** Block diagram of the proposed model-free algorithm. **(b):** An example of data collected with an in-situ gas sensor. Notice the detected bouts in red.

To evaluate the applicability of the bout approach, we used an array of six commercial MOX sensors[1] on-board a mobile robot to collect data at different locations of an indoor corridor where a gas source is located. The gas source is a plastic container filled with ethanol. A bubbler facilitates evaporation while a fan placed nearby the source allows to spread the analytes away. As shown in Figure 2(a), no correlation between bout count and distance to the source is observed. This is in stark contrast with what is stated in [1]. The reason for this is that the authors conducted their experiments inside a wind tunnel with controlled airflow conditions. We proposed a novel indicator of source proximity using the bout approach. Instead of counting bout occurrences, we calculate the bout amplitudes. As can be seen

[1] https://www.sgxsensortech.com/

in Fig 2(b), the average bout amplitude is a good indicator of the distance to the gas source in uncontrolled environments. In some cases, when moving away from the source by 0.5 m the average amplitude slightly increases. However, in most cases, the average bout amplitude decreases when moving away from the source for more than 1 m.

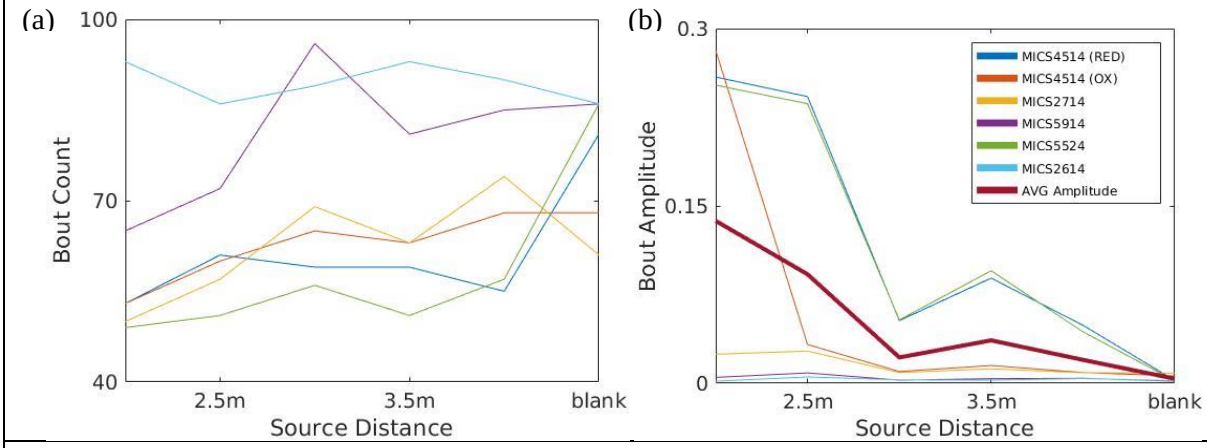



**Fig 2:** Performance of 2 proximity indicators computed from an array of MOX sensors. (a) bout count. (b) Bout amplitude.

The next step in the algorithm is the estimation of f(x) at non visited locations (Fig 1(a)). We thus model the average bout amplitude μ as a Gaussian process with a radial kernel function, stretched according to the wind direction as proposed in [2]. One of the advantages of using Gaussian regression process is that it estimates the a-posteriori variance *v*. The kernel corresponds to the assumption that positions in upwind direction have bout amplitudes similar to the measurement point. It also expresses the exploitation component of the exploration strategy, which leads the robot to follow a gas plume. Fig. 3(a) show examples the effect of the wind vector over the kernel while Fig. 3(b) shows the result of the extrapolation method from data collected with the robot.

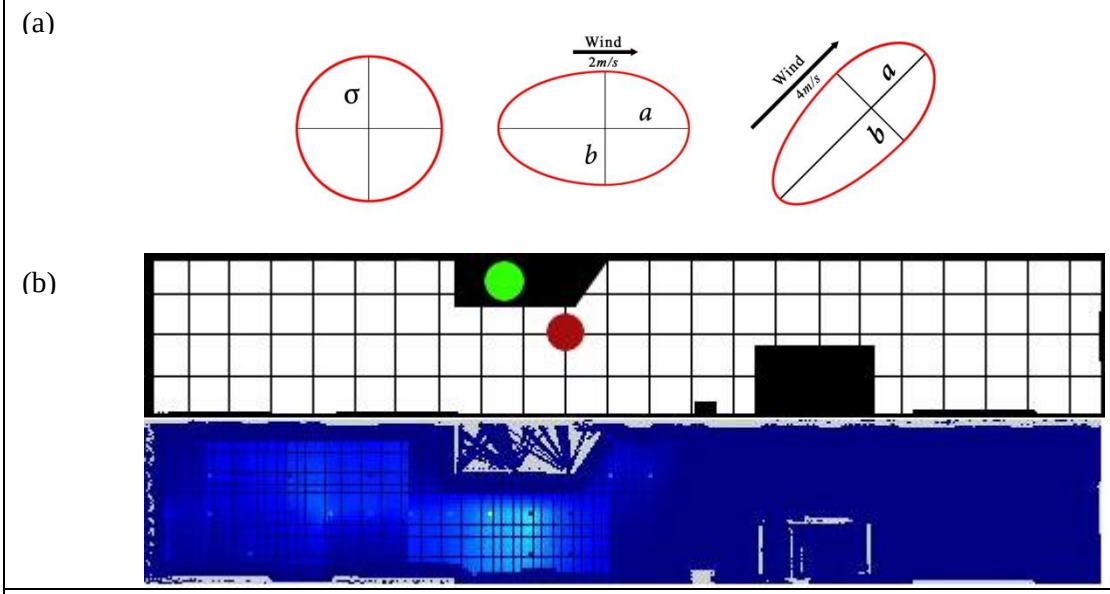



**Fig. 3: (a)** Influence of the wind speed and direction on the shape of the kernel. The ellipse is rotated according to the wind direction. The semimajor axis a is stretched in the upwind direction and shrunk in the downwind direction according to the wind speed. **(b)** An example of a bout amplitude map. In the upper plot (i.e. the grid map), the green circle indicates the gas source and the red circle the final position of the robot. The bottom plot corresponds to the mean bout estimate. The dark blue regions map areas with low bout amplitude estimates, whereas the light blue indicates high mean values.

To suggests Positions Of Interest (POI) to the operator (Step 3 in figure 1(a)), a trade-off between exploration of unvisited areas (following the variance gradient) and exploitation (following the direction to the highest bout amplitude estimate). The next POI corresponds to a movement of ρ meters from the current position with a direction of θ. The next POI will either be a position closer to the location with the highest variance (exploration) or a position closer to the highest bout amplitude estimate.

To test our proposed approach, we ran 12 experiments in the indoor scenario previously described. We consider an experiment successful if the robot chooses as the final position the reachable cell nearest to where the gas source is placed. A total of 8 runs were deemed successful, giving our method a 67% success rate. Reasons that could explain the failed experiments include the high wind speeds (which lead to bouts that cannot be resolved) and that the plane in which the robot sampled gas concentrations was at a substantially lower height than the gas source.

## B.2 Model-based sensor planning approach

In Smokebot, we also developed a model-based approach for sensor planning. We used a gas dispersion model based on Partial Differential Equations (PDE), in order to address gas distribution mapping and gas source localization. To address concerns related to computational cost, we propose a probabilistic model based on diffusion PDEs to approximate the complex behavior of gas dispersion. Such model suggests likely locations of gas sources and to suggest POI to human operators.

Physical mechanisms causing gas propagation are not trivial, and in case of turbulence can even exhibit non-deterministic and chaotic behavior [3]. Nonetheless, for on-line mapping scenarios a simplified approximation of the physical phenomenon with low computational complexity might be of great use. To this end, we investigated the capability of a PDE to approximate spatial gas dynamics for the purpose of identifying sources that drive the gas propagation.

Regarding sensor planning, we build upon known methods that use, for example, linear-quadratic control techniques [4], optimal experimental design and probabilistic approaches [5]. We then propose an exploration strategy that minimizes the uncertainty of the source localization following a criterion similar to an A-optimality [6]. Additionally, similar to [7] we exploit the assumption that the sources causing gas dispersion are sparsely distributed and use sparse Bayesian learning techniques to model this.

From a practical perspective, approximating complex gas dynamics with simple models can be of an advantage. In Smokoebot, we use a 2D diffusion model that can be formally described with a linear parabolic partial differential equation (PDE) that creates a spatio-temporal model of the gas concentration as follows:

| | |
|---|---|
| C.1 | $$\frac{\partial f(\boldsymbol{x},t)}{\partial t} - \kappa \Delta f(\boldsymbol{x},t) = u(\boldsymbol{x},t)$$ |

In the above equation, **x** corresponds to a 2-D position, *t* corresponds to time and κ corresponds to a diffusion coefficient. The term u(**x**,t) on the right hand side represents a spatio-temporal inflow of material; in this work we aim to identify this process.

In order to solve a PDE, numerical approximation methods are often the instrument of choice. To this end we begin with a classical Finite Difference Method [8] which "transforms" our diffusion PDE into a finite dimensional linear system by appropriately discretizing both space and time. In other words,

we divide our region of interest into a finite number of grid cells and consider temporal evolution of concentration values in each cell at discrete time steps n.

To facilitate numerical estimation of the PDE, we assume that gas dispersion mechanisms are complex and not always deterministic and that gas concentrations are not observed directly, but are rather measured with a sensor. This permits us taking a probabilistic approach toward PDE solution by treating all variables as random. We also assume that the gas sources are sparse in space, i.e., the gas distribution is only driven by a few unknown discrete sources, which allows us to use Sparse Bayesian Learning (SBL) techniques [9, 10]. Using Bayes theorem, the overall problem is reduced to stimating the PDFs of variables such gas concentrations at the cells in the grid, the location of the gas sources. The derivation of the equations of our algorithm can be consulted in Appendix II and III.

With what respect to sensor planning and the identification of POI, we take advantage of our probabilistic framework. Specifically we consider that a high variance of the estimated gas source marginal PDFs implies a high uncertainty. This quantity can be used to rate all cells in the region of interest, and cells with highest uncertainties can be provided to the human operator as proposals for new measurement locations.

In addition, we implemented an automatic exploration strategy that can be used for one or several autonomous platforms. In our approach, the robot selects a cell to explore, based on the cell's uncertainty value and its distance to the current robot position. In other words, the robots not only prefer cells with high uncertainty but also cells which are close to their current positions. When a target is reached, a measurement is taken, which maximally reduces the uncertainty of the corresponding grid cells.

For evaluation purposes, we used a hardware-in-the loop approach where gas dispersion was generated using a state-of-the-art simulator. We generated a dynamic gas distribution driven by one source in an area of 20m times 60m. In this environment, we used our proposed approach and a small number of robots. Whenever a robot is triggered to collect a measurement, the simulated gas concentration at the robots' position is used as a synthetic measurement for the exploration algorithm.

Fig. 4 visualizes the result of the exploration at a certain time step, whereas Fig. 5 depicts the performance of the estimates over time. The source localization error is calculated by comparing the center of mass of the estimated source distribution with the actual position of the source used in the simulation. The concentration estimate is assessed by the Normalized Mean Square Error (NMSE) with respect to the concentration given by the simulator. Although the difference between the simulated gas concentration and the estimated concentration based on the simplified model is rather high due to the over simplifying diffusion approximation, the estimated source strength distribution is quite accurate and the robots are able to localize the source in a reasonable time frame with good accuracy.

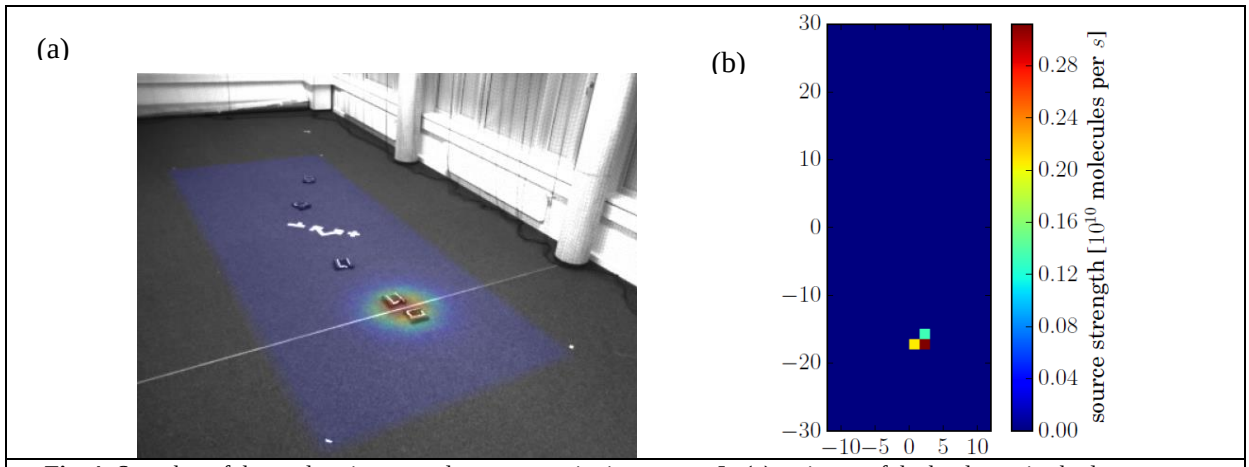



**Fig. 4:** Snapshot of the exploration procedure at a certain time stamp. In **(a)** a picture of the hardware-in-the-loop system where the simulated gas dispersion is shown as an overlay. Plot **(b)** displays the distribution of POI.

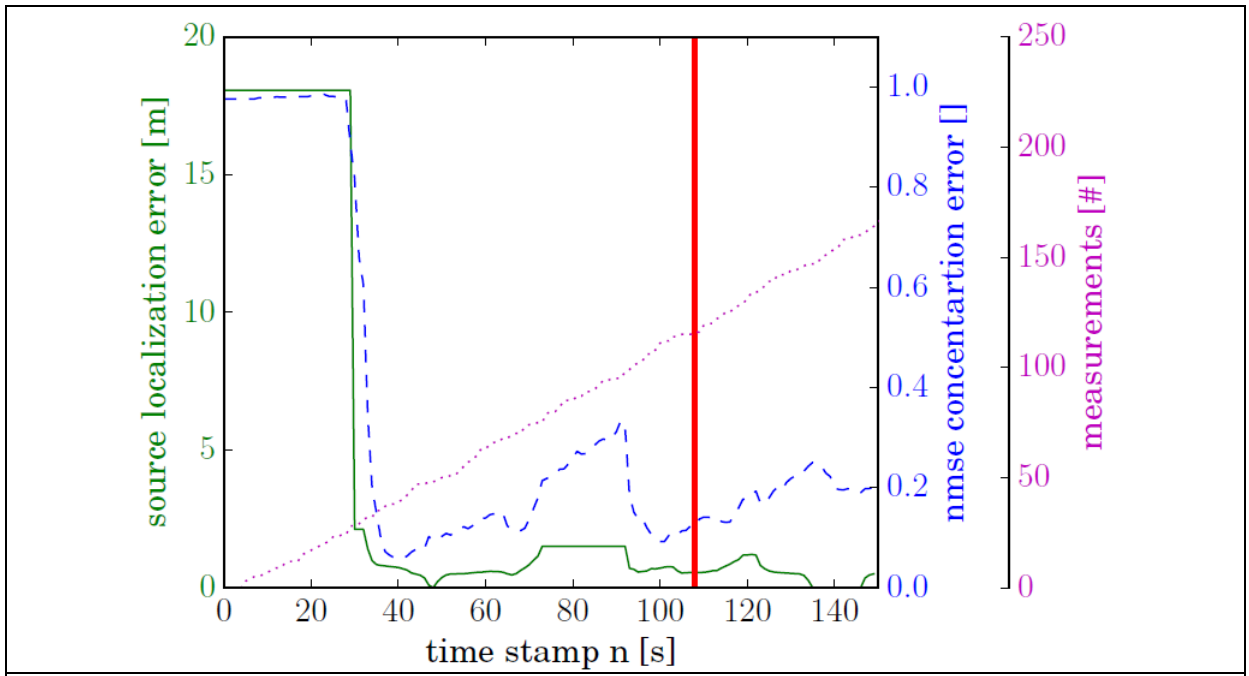



**Fig. 5:** Exploration performance. The vertical red marker denotes the time stamp at which Fig. 4(b) was generated.

## C Sensor Planning for remote sensors

Remote sensors acquire information or measurements without direct contact with the object under study. Examples of remote sensors are LIDARs, gas sensors based on spectroscopy principles (TDLAS) and thermal cameras. In Smokebot, we developed an adaptive sensor planning algorithm that considers two key parameters in remote sensing, namely field of view and the range of the sensor. By field of view we refer to the extent of the observable world that is sensed at a given time. We define field of view by an angle through which the device is sensitive.

We test our proposed algorithm by addressing the task of temperature estimation using a remote thermal camera. This is a challenging task because thermography depends on environmental influences/conditions in many ways. Only to name a few, there are dependencies on material category, viewing angle, surface character, surrounding mediums, etc. The influence of these variables can led

to missinterpretations in the thermail images. As explained in the following subsection, the main challenge in thermography is the unknown emissivity when looking at objects. Emissivity can have a value between 0 and 1. An emissivity close to unity eliminates the reflected radiation intensity part, whereas an emissivity close to zero makes it difficult to infer the real object temperature.

## C.1 Emissivity and temperature estimation

An example of the effects of emissivity in temperature estimation is shown in Fig. 6(a) where the whole thermal image is interpreted using an emissivity of $\varepsilon = 1$. In the front, there is a heated, polished aluminum plate (low emissivity) with a stripe of duct tape (high emissivity) on it. Although the aluminum plate and the duct tape have roughly the same temperature on the entire surface (about 330 K), their temperatures are interpreted differently. Due to the previously discussed relations between the true object temperature and the actually measured intensity, the temperature of the blank metal surface is interpreted much lower than the temperature of the dielectric duct tape.

Emissivity is highly dependent on the viewing angle. Depending on this angle the emissivity of metals and dielectrics behave differently. This effect is illustrated in Fig. 6(b). While for metals the emissivity is constant low for viewing angles between 0° and 60°, it then grows to a maximum just before reaching its minimum at 90°. For ideal dielectrics, the emissivity has a constant high value at viewing angles between 0° and 50°. At higher angles, it monotonically decreases to zero

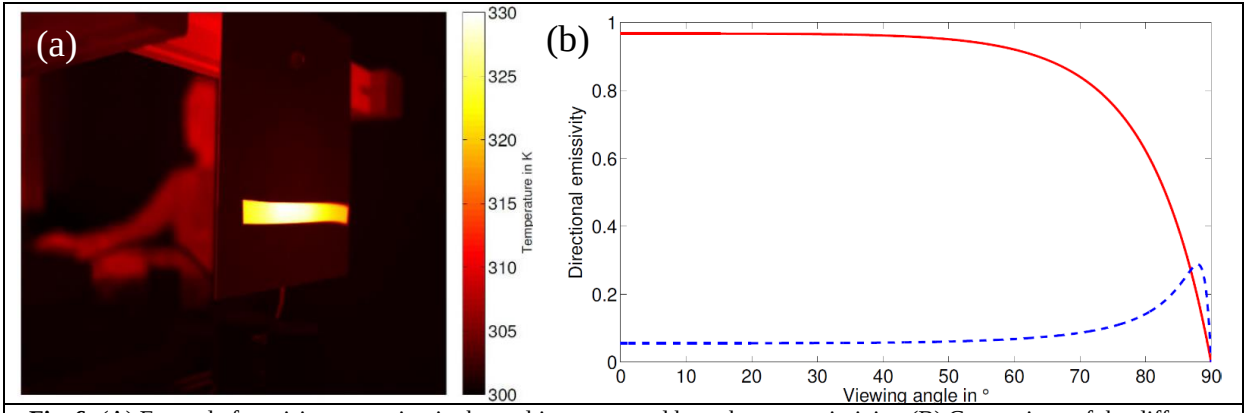



**Fig. 6:** (A) Example for misinterpretation in thermal image caused by unknown emissivity. (B) Comparison of the different emissivity characteristics of metals (dashed line) and dielectrics (solid line) at varying viewing angles.

Considering the above challenges, we developed in D3.1 an algorithm to estimate the actual temperature of a body regardless of the material it is made of. We framed the estimation of the emissivity and the temperature of a surface point as a least square minimization. In our approach, measurements (i.e. detector signals) from two different points of view are used to estimate parameters such as object's temperature ($T_{obj}$), refractive index (n) and extintion coefficients (k). The parameter $T_{obj}$ corresponds to the temperature of the surface point while k and n are meta parameters of the emisivity model $\varepsilon(\theta)$, which depends on the viewing angle theta (as shown in Fig. 6(b)).

In the next subsection we explain our proposed sensor planning algorithm and we explain how it can be used in the task of temperature estimation. When a thermal camera is mounted on a mobile robot, data can be acquired at different poses $(x,y,\theta)$, which we refer to as a sensing configuration c. Thus, a solution for the surveillance task is a tour composed of a number of sensing configurations $\{c_1,c_2,...,c_n\}$. An efficient plan has a limited number of sensing configurations and a short traveling distance while at the same time, it provides a high sensing coverage with an accurate estimation of temperature profiles.

## C.3 Adaptive sensor planning for remote sensor

As previously stated, Smokebot considers coverage and exploitation as key problems in sensor planning. In the context of remote sensing with thermal cameras, coverage refers to scanning the whole target environment in order to identify "areas of interest" that show high temperature levels. Exploitation refers to gathering detailed information about these "areas of interest" in order to estimate accurate temperature values.

In our proposed algorithm, we conduct exploration and exploitation in a single measurement tour. Initially, a set of sensing configurations that guarantee the full exploration of the area under surveillance are estimated (as originally presented in [11]). Then, when an interest location is identified (i.e. an area that shows high temperatures), exploitation is conducted by suggesting sensing configurations that minimizes the effect of the emissivity and thus maximizes the accuracy of the temperature estimation.

We represent the environment in a Cartesian grid of occupied and free cells. In addition, we consider a discrete number of candidate sensing configurations; this means that we consider only the locations at the centre of each cell and a limited number of orientations (e.g. 0º, 90º, 180º, 270º). The coverage is the capture by a matrix V of size equal to the number of candidate configurations times the number of free cells **S**.

A sensor planning solution for exploration is refered to as $\boldsymbol{\pi}_{\mathbf{detect}}$, which is a list of sensing configurations that provides the desired sensing coverage. $\boldsymbol{\pi}_{\mathbf{detect}}$ is estimated by solving the optimization problem in Equation C.1, where C is a binary decision vector of candidate sensing configurations, and $\boldsymbol{C}$ is the sensing coverage provided by the selected configuration. The parameter $\mathbf{n}_{\mathbf{cov}}$ determines the required coverage percentage.

| | |
|---|---|
| C.1 | $\pi_{\text{detect}} = \text{argmin } \lvert C \rvert \text{ s.t. } \mathcal{C} \geq \mathfrak{n}_{\mathbf{cov}}$ |

For the exploitation problem, we estimate $\boldsymbol{\pi}_{\mathbf{tomo}}$, which aims to provide accurate temperature estimations. The accuracy of the estimation is measured by $\boldsymbol{Q}$, which is an arbitrary quantity inversely proportional to the error between the estimated and the true temperature value. To determine $\boldsymbol{Q}$ we can use, for example, the emisitivy curves shown in Fig. 6(b). As $\boldsymbol{\pi}_{\mathbf{detect}}$ is being executed, areas of interest are identitied, which then need to to be sampled from different view points. Having different view points is critical for the temperature estimation algorithm, as explained in the previous subsection. Equation C.2 is the optimization problem for exploitation, where $\mathbf{n}_{\mathbf{tomo}}$ is the desired reconstruction quality.

| | |
|---|---|
| C.2 | $\pi_{\text{tomo}} = \text{argmin } \lvert C \rvert \text{ s.t. } \mathcal{Q} \geq \mathfrak{n}_{\mathbf{tomo}}$ |

Given the above solutions for the exploration and exploitation, our sensor planning algorithm can be summarized as follows: First, we estimate $\boldsymbol{\pi}_{\mathbf{detect}}$ and we start executing it by collecting measurements at the suggested locations. If high temperatures are detected, then $\boldsymbol{\pi}_{\mathbf{tomo}}$ is computed for all the neighbouring areas. The plan $\boldsymbol{\pi}_{\mathbf{tomo}}$ is iteratively executed and improved for the each selected configuration. At the end of an exploitation process, $\boldsymbol{\pi}_{\mathbf{detect}}$ is re-computed for the remaining unexplored areas. This process continues until no configuration is left in $\boldsymbol{\pi}_{\mathbf{detect}}$.

To illustrate how the proposed algorithm works, we ran a simulated environment using Matlab, were we considered a field of view $\varphi = 45º$, and a range r = 20m for thermal camera. Fig. 7 shows the environment setup (obstacle locations and locations of the high temperature bodies) and the robot's measurement configurations. The location of the high temperature bodies are denoted by red markers. $\boldsymbol{\pi}_{\mathbf{detect}}$ is shown as a series of green circular segments with angles and radius equal to the sensor's field

of view and range respectively. Notice that at configuration 21, high temperatures are reported near the topmost outher wall. This triggers the computation of $\pi_{tomo}$, which is a re-planning process for exploitation. In our example, $\pi_{tomo}$ corresponds to the red circular segments, for example configuration 22 when the first gas source is detected. After $\pi_{tomo}$ is executed, $\pi_{detect}$ is re-computed and those configurations that are no longer needed are discarded (gray markers in Fig. 7). Similarly, when another high temperature body is detected by configuration 11, the computation of a new $\pi_{tomo}$ is triggered, which leads to the suggestion of configuration 12 (in red) for exploitation purposes.

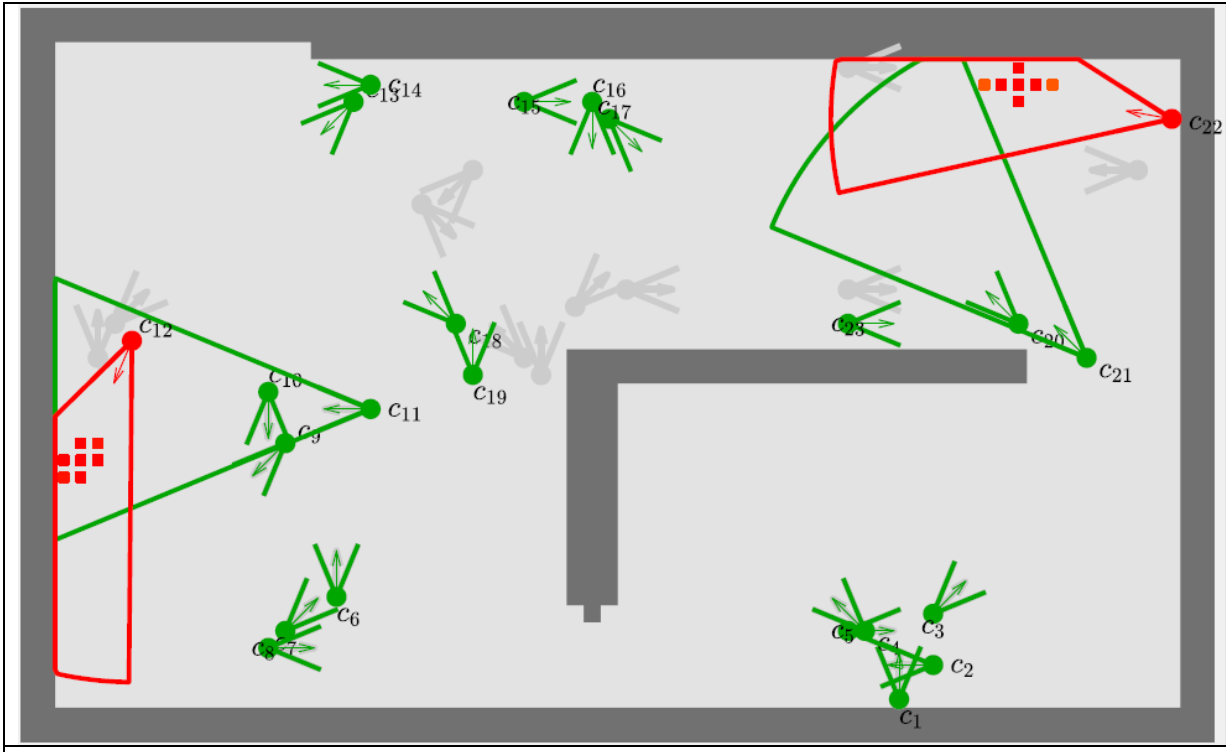



**Fig.7:** Simulation environment where an exploration tour is conducted. $\pi_{detect}$ is shown in green while $\pi_{tomo}$ is shown in red. The location of high temperatures are denoted by squared red markers.

## D Sensor planning for communication coverage

To maintain a communication link between the human operator and the robotic platflorm, Smokebot developed a series of ruggedized wireless repeaters that are deployed as the mission is being conducted (T7.3) in order to extend the range of the Wireless (WLAN) network. However, it could be the case that the robot goes out of the WiFi range when in a mission. As part of the self preservation functionality (T5.2) , the robot will continuously monitor radio frequency (RF) signal strength and augment the General Disaster Model with it.

To predict the connectivity of the network and to suggest a possible location to drop a repeater, Smokebot developed a ray-tracing-based algorithm that estimates the WLAN range of the network and extend it if needed by suggesting POIs where more repeaters could be dropped (D5.2).

Ray tracing is widely used in computer graphics to render high quality images: For each pixel on the screen, the path of the light is traced back into the scene it is coming from; the individual pixel is then colored in the color of the object(s) the light ray encounters. This method can not only be used to generate images, it is also used in physics to simulate the behavior of waves and particles.

Ray tracing is a suitable approach to simulate WiFi equipment since the estimation of WLAN coverage needs to be computed in real time and approaches that use, for example, the equations that

describe the propagation of electromagnetic waves, could provide a subpar performance due to computational requirements. Related work, such as [11, 12] have shown that ray tracing can be used to simulate WiFi with a good level of accuracy. Moreover, commercial tools to plan wireless networks are based on ray tracing[2].

We implemented the proposed ray-tracing-based algorithm for WiFi coverage in ROS. We modeled each repeater as a laser scanner device with a range that depends on the traversed objects. Thus, the scanner code was implemented in such a way that the projected rays do not stop on the first object they hit but instead, they keep track of the traveled distance and the number of hit objects. The maximum range is then adjusted according to the traversed objects. In addition, if another scanner is encountered this is recorded and the information can be used to create a map of connected network devices. The result of the modified ray tracing can be seen in Fig. 9. The area that is covered by the Wifi repeater in the middle is colored red. It can be seen that the signal reaches the furthest in the bottom of the image, while it is attenuated by walls in the other directions.

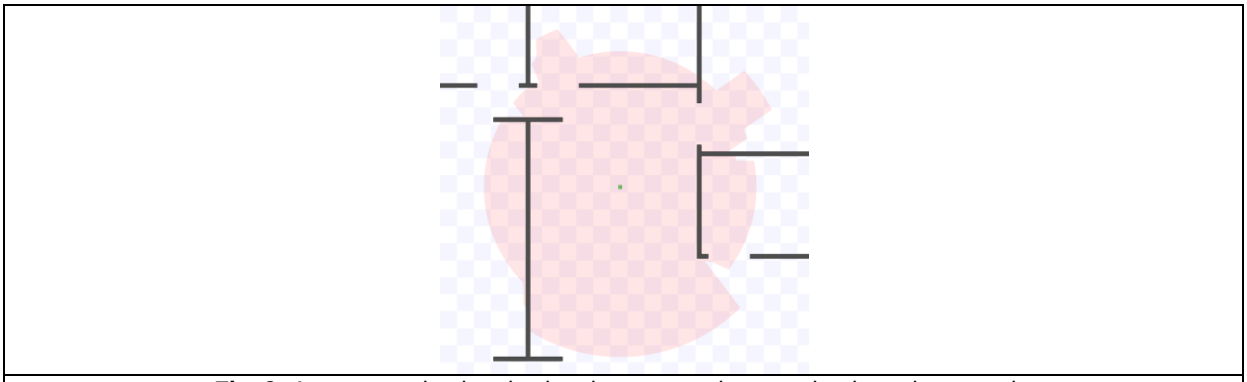


**Fig. 9:** A repeater simulated using the proposed ray-tracing-based approach.

Figs. 10(a) and 10(b) show how the proposed algorithm works. The algorithm was implemented as a ROS node that listens to the topics coming from the simulated repeaters currently in use. For each repeater, a polygon consisting of the translated points is kept updated, and to check if a point on the map is covered by the Wifi network one simply has to check if the point lies within one of these polygons. The algorithm also considers the input from the user in the form of the next position computed by the path planning algorithm. If the user commands the robot outside the area of coverage, a new Wifi repeater is placed early enough so that the robot always stays connected. Fig. 10(a) shows the coverage at the start of the exploration path and after it moves from its starting point in the lower left corner to the destination in the upper right corner Fig 10(b). Whenever the robot it is about to get outside the Wifi range, another repeater is dropped. The benefit of this approach is that it is completely independent of the specific path planner that is used.

[2] https://www.remcom.com/wireless-insite-em-propagation-software/

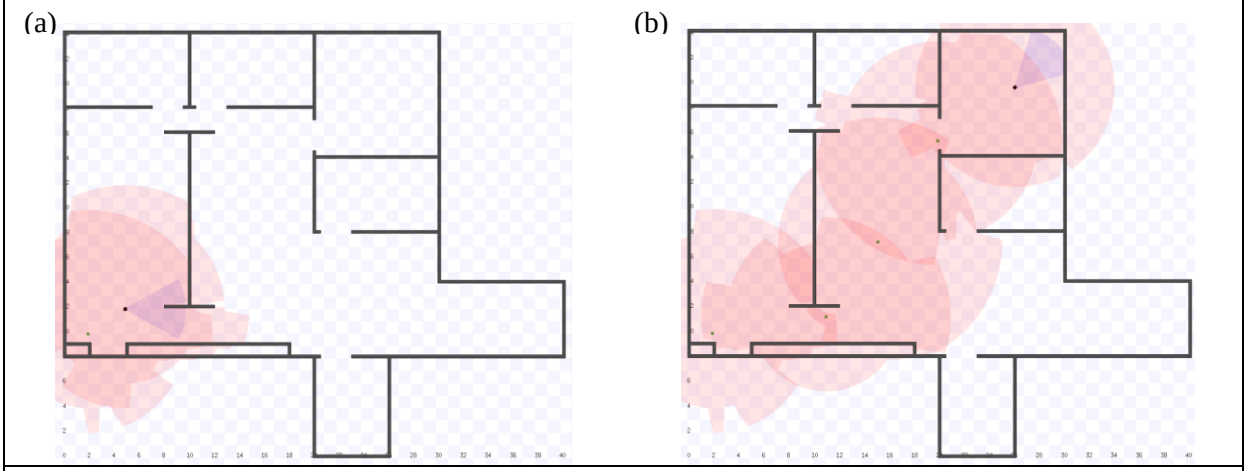



**Fig 10:** (a) Network coverage after the robot starts its exploration tour. (b) Network coverage after the exploration tour. In order to extend the connectivity of the network, several repeaters are dropped during the exploration path.

# E Summary and Outlook

In Smokebot we have developed different algorithms for sensor planning based on the sensor characteristics. More specifically, we developed algorithms for remote sensing, in-situ sensing and communication coverage. A key characteristic of Smokebot is that the human operator is in control of the exploration missions and thus, the sensor planning algorithms suggest new measurement locations as Positions Of Interest (POI) instead of controlling the robot's trajectory. It is up to the user operator (and the task that has higher priority at a given point) to decide where to move the robot.

For in-situ sensor planning, we explored two different approaches for gas source localization and gas distribution mapping. We first proposed a sensor planning algorithm that does not make any a-priori assumption about the gas dispersion (i.e. a model-free approach). This approach suggests POIs based on the variability of the gas measurements and airflow data and for non-visited locations, Gaussian extrapolation is used. This extrapolation model then allows to suggest POIs to explore the target area and to exploit the acquired information in order to detect the gas leaks. While the use of wind data could be perceived as a limitation of the algorithm in the Smokebot scenario, a possible solution is to use an isotropic Gaussian kernel. Regarding our model-based approach, we found that the performance. In addition, we proposed a model-based approach that relies on PDE to localize the gas sources and to map gas distribution. This algorithm proposes POIs based on hypotheses of where the gas source location might be (exploration) and areas where the uncertainty can be reduced (exploration). In our hardware-on-the-loop evaluation, we found that the algorithm converges rapidly towards the location of the gas source and that the algorithm is rather accurate when localizing gas sources.

Regarding sensor planning for remote devices, we use an adaptive sensor planning algorithm for temperature estimation using a thermal camera. The sensors planning algorithm first computes a set of POI for exploration (e.g. a given sensing coverage) and then, modifies the original plan as data is being collected in order to maximize a given performance metric (i.e. improving the temperature estimates by decreasing the effects of the emissivity). The algorithm presented in this report is flexible enough to be used with different remote sensors beyond thermal cameras since it depends on generic parameters such such as field of view ($\varphi$) and sensor range (r). Parameters such as $\mathbf{n}_{cov}$ (Equation C.1) denote the desired coverage (exploration) and it is not bound to the particular task of temperature estimation. Similarly, $\mathbf{n}_{tomo}$ (Equation C.2) is a parameter that denotes a desired performance level.

To conclude, we addressed sensor planning for the task of communication coverage. For this particular task, we implemented an ad-hoc algorithm that models the coverage of the individual repeaters as range sensors and considers the interference of obstacles in order to determine the overall coverage of the network. The proposed algorithm considers the estimated coverage and the next robot position (provided by the motion planning algorithm) in order to determine whether or not the robot is going out of range. If it is determined that the robot is moving outside the area of coverage, a repeater is dropped. While our ad-hoc algorithm diverges from the POI approach, it allows nevertheless for future developments. For example, it would be possible to provide the information about the network coverage to a specialized path planner that could then take this information into account and plan the route so it tries to stay in range of the WiFi.

## References


| | |
|---|---|
| **[1]** | M. Schmuker, V. Bahr, and R. Huerta, "Exploiting plume structure to decode gas source distance using metal-oxide gas sensors", Sensor and Actuator B-Chemical, vol. 235, pp. 636-646, 2016 |
| **[2]** | M. Reggente and A. J. Lilienthal, "Using Local Wind Information for Gas Distribution Mapping in Outdoor Environments with a Mobile Robot", in Proc. IEEE Sensors, pp. 1715-1720, 2009. |
| **[3]** | Roberts, P., and Webster, "Turbulent Diffusion. Environmental Fluid Mechanics Theories and Application", Reston, VA: ASCE Press,2002. |
| **[4]** | M. Demetriou, "Numerical investigation on optimal actuator/sensor location of parabolic PDEs," Proceedings of the 1999 American Control Conference, vol. 3, no. June, pp. 1722–1726, 1999. |
| **[5]** | F. Pukelsheim, Optimal Design of Experiments. John Wiley & Sons, Inc., 1993. |
| **[6]** | A. Alexanderian, N. Petra, G. Stadler, and O. Ghattas, "A-optimal design of experiments for infinite-dimensional Bayesian linear inverse problems with regularized l0-sparsification," SIAM Journal on Scientific Computing, vol. 36, no. 5, p. 27, 2013. |
| **[7]** | K. Kunisch, K. Pieper, and B. Vexler, "Measure Valued Directional Sparsity for Parabolic Optimal Control Problems," SIAM Journal on Control and Optimization, vol. 52, no. 5, pp. 3078–3108, jan 2014. |
| **[8]** | Strikwerda, J.C.: Finite Difference Schemes and Partial Differential Equations, 2 edn. Society for Industrial and Applied Mathematics, Philadelphia (2004) |
| **[9]** | M. E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," Journal of Machine Learning Research, 2001. |
| **[10]** | D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," Signal Processing, IEEE Transactions on, vol. 52, no. 8, pp. 2153–2164, 2004. |
| **[11]** | M. A. Arain, M. Trincavelli, M. Cirillo, E. Schaffernicht and A. J. Lilienthal. Global coverage measurement planning strategies for mobile robots equipped with a remote gas sensor. Sensors, 15(3):6845-6871, 2015 |
| **[12]** | T. R. S.Y. Seidel, „A ray tracing technique to predict path loss and delay spread inside buildings," 1992. |
| **[13]** | R. Valenzuela, „A ray tracing approach to predicting indoor wireless transmission," 1993. |
| | |

## Appendixes

# Exploration and Localization of a Gas Source with MOX Gas Sensors on a Mobile Robot - A Gaussian Regression Bout Amplitude Approach

Mikel Vuka[1*], Erik Schaffernicht[2], Michael Schmuker[3], Victor Hernandez Bennetts[2], Francesco Amigoni[1] and Achim J. Lilienthal[2]

***Abstract*— Mobile robot olfaction systems combine gas sensors with mobility provided by robots. They relief humans of dull, dirty and dangerous tasks in applications such as search & rescue or environmental monitoring. We address gas source localization and especially the problem of minimizing exploration time of the robot, which is a key issue due to energy constraints. We propose an active search approach for robots equipped with MOX gas sensors and an anemometer, given an occupancy map. Events of rapid change in the MOX sensor signal ("bouts") are used to estimate the distance to a gas source. The wind direction guides a Gaussian regression, which interpolates distance estimates. The contributions of this paper are two-fold. First, we extend previous work on gas source distance estimation with MOX sensors and propose a modification to cope better with turbulent conditions. Second, we introduce a novel active search gas source localization algorithm and validate it in a real-world environment.**

## I. Introduction

Mobile Robot Olfaction (MRO) studies the combination of mobile robots with gas sensors to solve practical problems related to gas sensing. Among others, MRO systems perform gas discrimination, gas source localization (GSL), and gas distribution mapping. GSL can be of great importance for applications such as search and rescue missions, or environmental monitoring. Robotic solutions are especially favourable in dull, dirty or dangerous scenarios. When the gas of interest is harmful to humans, for example, it is indispensible to localize gas sources with a robot. The length of robot missions, however, is typically considerably limited by the available energy, both for ground and airborne robots. The challenges in GSL thus include importantly to find efficient navigation strategies that minimize the amount of time required for searching for a gas source.

In this paper, we present an active search GSL algorithm for a mobile robot, equipped with Metal-Oxide (MOX) gas sensors and a wind sensor. The robot searches a known environment (i.e., we assume that an occupancy map is given) for the gas source. The approach that we introduce estimates the gas source distance from the robot's position and aims to minimize it by exploiting a model of the wind flow and how it affects the gas distribution.

[1]Dipartitmento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133, Milano, Italy

[2]Mobile Robotics & Olfaction Lab, Center of Applied Autonomous Sensor Systems (AASS), Örebro Universitet, SE-70182 Örebro, Sweden.

[3]University of Hertfordshire, School of Computer Science, College Lane, Hatfield, Herts AL10 9AB United Kingdom

*Corresponding author mikel.vuka@mail.polimi.it

This work was funded in part by H2020-ICT project SmokeBot (645101).

The problem of gas source localization has been studied in the past two decades and there are several methods to approach it. In general, plume-tracking algorithms make use of wind flows (anemotaxis) and/or gas concentration (chemotaxis) to find a direction for the robot to follow. A broad group of algorithms try to mimic odor source localization performed by insects in nature [3], [4]. In these approaches the robots follow the gas plume in the upwind direction by making use of both anemotaxis and chemotaxis principles, whereas in our work we include an explicit source distance estimation in the process. For this estimation we draw upon recent works, which showed that the information conveyed by MOX sensors can be used to estimate the distance to a gas source in wind tunnel conditions [1]. Li et al. [6], [7] approach the problem by estimating the source location inside an area and then try to minimize this area. Some more complex algorithms model the location of the gas source with a probability distribution and then try to reduce its entropy [5]. These algorithms model the source location probabilistically, while we search the environment and build a probabilistic model of the source distance.

The contributions of this paper are two-fold. First, we introduce and validate a novel active search GSL approach. Second, we extend the work on gas source distance estimation, which was carried out in a wind tunnel [1]. We show results of the source distance estimation technique presented in [1] in real-world environments and introduce a modified version better suited to the task.

## II. Problem Definition and Approach

We assume that the environment is known a priori and is not subject to change during a mission. The environment is represented as a Cartesian grid, thus obtaining a set $M$ of $N$ cells of identical size: $M = \{x_1, ..., x_N\}$. We also assume that only one source of gas is present in the environment.

The robot is equipped with an array of six in-situ sensors. In order to measure wind speed and direction, the robot is equipped with an ultrasonic anemometer.

The robot moves between the centers of the cells. After each movement it records measurements of gas concentration and wind information for a certain amount of time in order to compute the source distance estimation. We define a function $f : M \rightarrow \mathbb{R}^+$ to indicate the distance $f(x)$ from a cell $x \in M$. The function $f$ is unknown a priori and is updated from noisy measurements. From the measurements made in visited areas, the robot estimates values of the function $f$ in unvisited areas of the environment and uses this knowledge

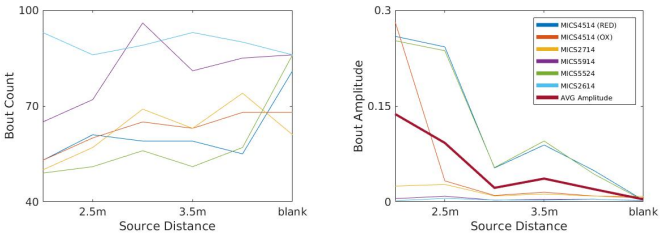



Fig. 1: Bout count and amplitude for each sensor vs. distance to the gas source at 0.1 m/s wind speed.

to decide where to navigate next in order to improve the estimation in an efficient way. The estimation of the source distance is thus performed online, as the robot visits the environment. The approach proposed in this paper consists in the repetition of the following steps: acquire sensor data from the current position, compute the source distance estimation, estimate $f(x)$, and finally move the robot to the next position selected by the search strategy. The two main modules of our system, namely source distance estimation and search strategy, are described in the following.

## III. Source Distance Estimation

One of the crucial steps of our approach is the detection of the *bouts* of the signal, i.e., portions of the filtered signal where the amplitude is rising. In [1], it is reported that there is a strong correlation between the number of bouts and the distance to the gas source: the higher the bout count, the closer the sensor to the gas source.

To detect the bouts, a cascaded filtering approach is used to detect fast transients in the sensor signal [1]. A low-pass filter is first used in order to remove high-frequency noise. This is done by applying a Gaussian convolution with $\sigma_{smooth} = 0.3s$. On the smoothed signal, a differential convolution is applied to show differences between pairs of samples and see the amplitude changes. Finally, the signal undergoes an exponentially-weighted moving average filter with a half life $\tau_{half} = 0.4s$. The operation that yields the filtered time series $y_t$ from the low-pass filtered $z_t$ can be expressed by the following equation:

$$y_t = (1 - \alpha) * y_{t-1} + \alpha * z_t \tag{1}$$

where $\alpha = 1 - \exp \frac{log(0.5)}{\tau_{half} \Delta t}$ and $\Delta t$ is the time step in the equation. Bouts of rising amplitude can be identified on the differential of the filtered signal ($y'_t$). The presence of a bout is characterized by $y'_t$ being equal to or greater than zero.

The bout method in [1] was evaluated only inside a wind tunnel, while we are considering open environments. The gas plume was generated through evaporation of propanol placed inside an open plastic container. A constant wind flow was generated with a fan placed near the gas source. The bout detection was tested with low wind speeds (0.1 m/s to 0.4 m/s). The propanol container was placed in between the robot and the fan at different distances from the robot in 6 different locations in the range from 2 to 4 meters. The sampling rate of the sensors was set to 74 Hz and the sensing time of the robot was 135 seconds, obtaining a gas concentration signal

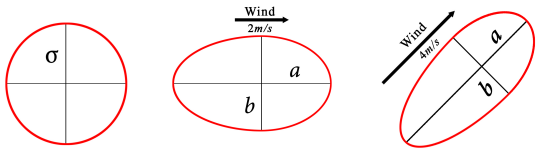



Fig. 2: Influence of the wind speed and direction on the shape of the kernel. The ellipse is rotated according to the wind direction. The semimajor axis $a$ is stretched in the upwind direction and shrunk in the downwind direction according to the wind speed. $\sigma$ represents the spatial scale of the kernel.

with a total of $10,000$ samples per location for the bout detection algorithm. Experiments were repeated 6 times.

Our experiments show that the bout count gives mixed results in this scenario and cannot be used to reliably estimate the distance to the gas source. In some cases it can be seen that the bout count even increases with distance. From the analysis of the gas concentration signal, we noted, however, that the amplitude of the bouts tends to decrease with distance, making it possible to estimate the source distance. The distance is thus estimated for each sensor as the mean value of the amplitudes of all the bouts. Our results (Fig. 1) actually show that the average bout amplitude is a good indicator of the distance to the gas source. In some cases, when moving away from the source by 0.5 m the average amplitude slightly increases. However, in most cases, the average bout amplitude decreases when moving away from the source for more than 1 m.

In the experiments performed in outdoor environments the wind flow conditions were not stable, resulting in the wind changing direction very often. Moreover, the wind speed was much higher than in indoor environments (about 1 m/s). Because of the high wind speed, the transients of the signal are too short and the sensors cannot resolve them, resulting in failure to detect the bouts. Overall, the sensors used in our experiments were able to reliably estimate the source distance in wind speeds up to 0.3 m/s.

## IV. Search Strategy

### A. Algorithm

We model the average bout amplitude function $f(x)$ (which is related to the distance from the robot to the gas source) as a Gaussian process with kernel function $k : M \times M \to \mathbb{R}^+$. Since, in realistic environments, advection dominates gas dispersal we use the upwind direction to direct $f(x)$ towards the gas source. This then favors exploration in upwind directions. We include wind information using a radial kernel, stretched according to the wind speed and rotated according to the wind direction as in [2], see Fig. 2:

$$k(x, x') = \exp -\sqrt{(x - x')^T \Sigma^{-1} (x - x')} \tag{2}$$

where $\Sigma$ is the 2D covariance matrix of the Gaussian.

The kernel in Eq. 2 corresponds to the assumption that positions in upwind direction have a bout amplitude similar

to the measurement point. It also expresses the exploitation component of the exploration strategy, which leads the robot to follow a gas plume. The bout amplitude estimation at an unvisited location $x_*$ and the a posteriori variance are computed respectively as:

$$\begin{aligned} \bar{f}_* &= k_*^T [K + \sigma_n^2 I]^{-1} y \\ V[f_*] &= k(x_*, x_*) - k_*^T [K + \sigma_n^2 I]^{-1} k_* \end{aligned} \tag{3}$$

where $k_*$ and $K$ are abbreviations respectively for $k(x_*, X)$ and $k(X, X)$ and $X$ is the set of visited cells.

The direction towards the next sensing position is computed as a trade-off between exploration of unvisited areas (following the variance gradient) and exploitation (following the direction to the highest bout amplitude estimate). The robot moves to the next position by following a direction $\theta$ for a step size $\rho$. The direction $\theta$ is sampled from the following Gaussian distribution:

$$p(\theta) = \begin{cases} \exp \frac{-(\theta - \theta_m(s))^2}{\sigma_m^2} & \text{if } R > \tau \\ \exp \frac{-(\theta - \theta_v(s))^2}{\sigma_v^2} & \text{otherwise} \end{cases} \tag{4}$$

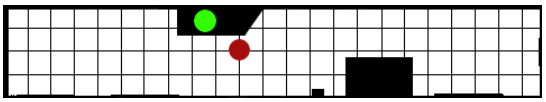
where $s$ is the iteration step, $\theta_m(s)$ is the direction to the highest bout amplitude estimation, $\theta_v(s)$ is the direction to the highest variance, $\tau$ is the trade-off parameter and $R \in [0, 1]$ is a random variable. At the beginning $\tau$ is set to 1. During the first steps of the mission the trade-off favors exploration and the robot moves towards unknown areas. After each step of the algorithm, the trade-off parameter decays to slightly lean more towards exploitation. If exploitation is favored, the robot follows the direction to the highest bout amplitude estimation. When the a posteriori variance is low enough (i.e., $\tau$ goes under a threshold), the algorithm terminates by declaring the position where the highest bout amplitude estimate is found as the final one.

### B. Experimental results

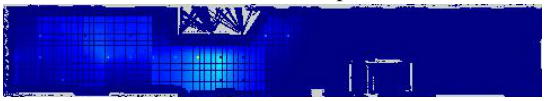
Experiments with a Clearpath Husky A200 robot were performed in a 22m x 4m indoor corridor.

The sensory function for Gaussian regression was derived from the bout amplitudes of the all signals of the sensor array. The estimation of the source distance is not always accurate if derived from a single sensor, especially the low resistance sensors were found to be not very reliable. The bout amplitude from the high resistance sensors, on the other hand, showed only small changes with distance. Therefore, the distance was estimated using all sensors. More specifically, for each sensing operation, the average bout amplitude for each sensor is calculated, then the six values are averaged together to have the final value. In this way, the effects of the sensors are balanced in order to get a good estimation. The occupied cells in the occupancy grid are considered to have a bout amplitude equal to zero.

We consider an experiment successful if the robot chooses as the final position the reachable cell nearest to where the gas source is placed, along the wind direction (Fig. 3). From a total of 12 complete experiments that were run, 8 were successful in identifying the proper final position, giving our method a 67% success rate. Reasons that could explain the failed experiments include the high wind speeds (which lead to bouts that cannot be resolved) and that the plane in which the robot sampled gas concentrations was at a substantially lower height than the gas source.



(a) Grid map of the environment: The black cells indicate obstacles. The green circle indicates the gas source and the red circle the final position of the robot.



(b) The mean estimate map: The dark blue regions map areas with low bout amplitude estimates, whereas the light blue indicates high mean values.

Fig. 3: Results of an experiment in which the wind was flowing towards the south-east direction.

## V. Conclusions

In this paper we introduced an integrated GSL and exploration approach that uses a mobile robot equipped with MOX sensors and a wind sensor. The proposed solution exploits the bout amplitude of the concentration signal and drives the robot towards areas where its value is expected to be maximized. Experimental results show that the bout amplitude of the signal is a good estimator of the source distance. The proposed search strategy performed with a success rate of 67% in indoor environments, identifying reachable areas near the gas source.

A possible improvement of the work done in this paper would be to consider scenarios with multiple gas sources. The robot then estimates the number of sources and the distance to the closest one. Possible extensions would be to study the bout amplitude response of different gases and try to learn detection thresholds in order to declare the presence of a gas source.

## References


[1] M. Schmuker, V. Bahr, and R. Huerta, "Exploiting plume structure to decode gas source distance using metal-oxide gas sensors", *Sensor Actuat B-Chem*, vol. 235, pp. 636-646, 2016.

[2] M. Reggente and A. J. Lilienthal, "Using Local Wind Information for Gas Distribution Mapping in Outdoor Environments with a Mobile Robot", in *Proc. IEEE Sensors*, pp. 1715-1720, 2009.

[3] Y. Kuwana, I. Shimoyama, and H. Miura, "Steering control of a mobile robot using insect antennae," in *Proc. IROS*, pp. 530-535, 1995.

[4] Y. Kuwana and I. Shimoyana, "Pheromone-guided mobile robot that behaves like a silkworm moth with living antennae as pheromone sensors.," in *Proc. IROS*, pp. 924-933, 1998.

[5] M. Vergassola, E. Villermaux, B. I. Shraiman, "'Infotaxis' as a strategy for searching without gradients", *Nature*, vol. 445, pp. 406-409, 2007.

[6] J.-G. Li, Q.-H. Meng, Y. W., and M. Zeng, "Odor source localization using a mobile robot in outdoor airflow environments with a particle filter algorithm", *Auton Robot*, vol. 30, pp. 281-292, 2011.

[7] J.-G. Li, Q.-H. Meng, F. Li, M. Zeng and D. Popescu, "Mobile Robot based Odor Source Localization via Particle Filter" in *Proc. CDC/CCC* December 16-18, 2009.

# Probabilistic Modeling of Gas Diffusion with Partial Differential Equations for Multi-Robot Exploration and Gas Source Localization

Thomas Wiedemann[1], Christoph Manss[1], Dmitriy Shutin[1], Achim J. Lilienthal[2], Valentina Karolj[1], Alberto Viseras[1]

***Abstract*— Employing automated robots for sampling gas distributions and for localizing gas sources is beneficial since it avoids hazards for a human operator. This paper addresses the problem of exploring a gas diffusion process using a multi-agent system consisting of several mobile sensing robots. The diffusion process is modeled using a partial differential equation (PDE). It is assumed that the diffusion process is driven by only a few spatial sources at unknown locations with unknown intensity. The goal of the multi-robot exploration is thus to identify source parameters, in particular, their number, locations and magnitudes. Therefore, this paper develops a probabilistic approach towards PDE identification under sparsity constraint using factor graphs and a message passing algorithm. Moreover, the message passing schemes permits efficient distributed implementation of the algorithm. This brings significant advantages with respect to scalability, computational complexity and robustness of the proposed exploration algorithm. Based on the derived probabilistic model, an exploration strategy to guide the mobile agents in real time to more informative sampling locations is proposed. Hardware-in-the-loop experiments with real mobile robots show that the proposed exploration approach accelerates the identification of the source parameters and outperforms systematic sampling.**

***Index Terms* — multi-agent exploration, gas source localization, mobile robot olfaction partial differential equation, factor graph, sparse Bayesian learning, message passing**

## I. Introduction

Robotic platforms are particularly suitable for exploration of dynamic gas distributions. In our understanding and in the context of this paper, such an exploration task covers issues like mapping of gas distributions and identification of gas sources. Specifically, this paper examines the task of finding gas sources, e.g. leaks. While gas distribution maps appear as a byproduct of our approach, gas distribution mapping is not in the main focus of this paper. In searching for the sources we exploit measurements of the gas concentration in the environment. As an application one may think of a technical accident or disaster response scenario, where toxic material is leaking and the location has to be identified. Such a scenario implies threats for a human operator and indicates the benefits of an automated sampling of the gas concentration. Therefore, we consider a robotic platform equipped with a gas sensor for exploring gas distributions.

[1]Institute of Communications and Navigation of the German Aerospace Center (DLR), Oberpfaffenhofen, 82234 Wessling, Germany, thomas.wiedemann@dlr.de, christoph.manss@dlr.de, dmitriy.shutin@dlr.de, valentina.karolj@dlr.de, alberto.viserasruiz@dlr.de

[2]Mobile Robotics and Olfaction Lab, Orebro University, 70182 Orebro, Sweden, achim.lilienthal@oru.se

Employing robotic platforms for chemical sensing and gas source localization is an active and emerging research field (see [1] for a survey). Where a single robot would be able to only serially sample the gas concentration at different times, a multi-robot system is capable of taking measurements at different locations at the same time. This is of advantage for analyzing the dispersion process, since the gas dispersion is dynamic, and the time-variant nature of gas dispersion is an important property, that should be taken into account [2]. Besides, a multi-robot system has additional advantages: i) multiple robots can achieve the exploration task faster; ii) a multi-robot system is more robust, since it possesses natural redundancies; and finally iii) the individual robots can make use of synergies, e.g. share the computational costs of algorithms. Even though it seems reasonable to employ a multi-agent system, an open issue is the design of an automated exploration strategy. This exploration strategy should guide a single or a swarm of robots to the sampling locations in an efficient way and should avoid unnecessary measurements. The development and analysis of such an intelligent distributed exploration algorithm to identify the sources of the gas dispersion is the main focus of this paper.

The exploration or sampling problem is closely related to optimal sensor placement techniques. In literature, some approaches consider this as an observer design problem. In those cases an observer performance is optimized by adapting the sensor location, e.g. in [3] to estimate a distributed process described by a partial differential equation. Another approach is a probabilistic treatment of the sampling process. In this context there are different criteria to measure the informativeness of the samples [4]. In literature, this is also referred to as an optimal experimental design problem [5]. More details on these topics can be found in [6], where the author gives an introduction to optimal sensor location and experimental design problems. A probabilistic or statistical view is also a contemporary and relevant research topic in the context of gas distribution mapping [7], [8]. Further, the benefit of using statistical properties of the gas distribution in an adaptive sampling strategy has already been shown in first real-world experiments [9].

The contribution of this paper is twofold: First, we propose a new adaptive model-based exploration strategy for multiple mobile robots. Second, we present an approach to evaluate this exploration strategy within a hardware-in-the-loop experiment. For the exploration we are following the ideas of an uncertainty or entropy driven exploration [10]. This means that the sampling scheme of the robots prefers regions

with high uncertainty. In other words, if the system has currently little knowledge about a region in the environment, it will guide robots to this location in order to carry out the next measurements. The approach is similar to our previous work [11]. However, in this paper robots possess a global view of the whole environment in contrast to the greedy algorithm in [11] where the robots only consider their direct neighborhood for a new measurement, Still, it is necessary to be able to quantify the uncertainty of our model parameter in different regions in the considered environment. To this end, we propose to make use of a model of the gas diffusion in the environment. In particular, since the gas diffusion is time and space variant, we use a partial differential equation (PDE) as a mathematical model. For the evaluation of the proposed multi-robot exploration, we developed a hardware-in-the-loop experimental setup. In general, it is difficult to evaluate gas distribution exploration in realistic experiments due to the difficulty in measuring ground truth gas concentrations. In addition, interesting gas distribution for realistic application may be of toxic nature and dangerous to handle. Therefore, we simulate the gas dispersion in the robots environment. In contrast to the gas dispersion, the robots themselves are not simulated. We employ a real multi-robot system to test our system and algorithm with "real world" constraints. In this way we are able to analyze how our distributed algorithm cope with realistic communication constraints like data rate, package delay or jitter. Further, limitation arising from the robotic platform, e.g. speed limits, collision avoidance and low-power on board computers, can be studied.

The outline of the paper is as follows: First, in Section II we explain the exploration strategy in detail and how it is implemented in a distributed fashion. In Section III we describe the experimental setup of the hardware-in-the-loop experiments and evaluates the performance of the proposed exploration strategy.

## II. Exploration Strategy

This chapter describes the design of the exploration strategy. We first introduce the gas diffusion model, which is used in a second step to quantify the uncertainty in different regions.

### A. Environment Model

As a mathematical model for the gas dispersion we use the 2D diffusion equation which can be described with the linear parabolic partial differential equation (PDE) [12]:

$$\frac{\partial f(\boldsymbol{x},t)}{\partial t} - \kappa \Delta f(\boldsymbol{x},t) = u(\boldsymbol{x},t). \tag{1}$$

This equation models the dynamic behavior of gas concentrations $f$ at a location $\boldsymbol{x}$ and time $t$ given the diffusion coefficient $\kappa$. The right hand side $u(\boldsymbol{x},t)$ represents the inflow of material, i.e. the spatial source strength distribution we want to identify. In a first step the PDE is numerically approximated via the Finite Difference Method (FDM) [13]. Therefore, we discretize space and time by dividing our region into grid cells and considering temporal evolution of gas concentration in each cell at discrete time steps $n$. Since we used the FDM, our numerical approximation results in a linear equation $r_c$ for each grid cell $c$:

$$r_c(\boldsymbol{f}[n], \boldsymbol{f}[n-1], u_c[n]) = 0. \tag{2}$$

The FDM treats the concentration and source strength across a cell as constant. The 1D vectors $\boldsymbol{f}[n]$ and $\boldsymbol{u}[n]$ aggregate the 2D concentration field and 2D source strength distribution for all grid cells at time stamp $n$. Both are unknown and have to be estimated based on measurements. This is the purpose of the exploration. A measurement done by a robot will be modeled as:

$$y_k[n] = \boldsymbol{M}[n]\boldsymbol{f}[n] + \xi, \tag{3}$$

with $\boldsymbol{M}[n]$ being a measurement matrix selecting the cell from $\boldsymbol{f}[n]$ that is currently visited by the robot. Additionally, $\xi$ represents a measurement noise mostly defined by the sensor characteristics.

### B. Uncertainty Quantification

For the uncertainty driven exploration procedure, we need a way to distinguish two cells with respect to their uncertainty. In order to quantify the uncertainty, we transform our model to a probabilistic representation. In the probabilistic framework all variables are treated as random variables and could be described by probability density functions (PDF). This permits us computing second order moments of the PDFs, e.g. the variance of a Gaussian distribution. The second order moments can be interpreted as measure of entropy or informativeness of a parameter. It is the second order moment of the source strength we propose to use as an uncertainty quantification of a cell.

For the probabilistic formulation, the linear equations of (2) are relaxed. Thereby we assume that the equations hold only with a certain precision $\tau_s$, with some random deviations from zero. We assume that those derivations or residuals in all grid cells are spatially and temporally white and statistically independent from each other. The precision $\tau_s$ reflects some uncertainty of our model assumptions. In this way the conditional probabilistic density function for all concentrations and sources can be formulated as follows:

$$p(\boldsymbol{f}[n]|\boldsymbol{f}[n-1], \boldsymbol{u}[n]) \propto \prod_{c=1}^{C} \mathrm{e}^{-\frac{\tau_s}{2}(r_c(\boldsymbol{f}_c[n], \boldsymbol{f}_c[n-1], u_c[n]))^2}. \tag{4}$$

The measurement model (3) could be transformed to

$$p(\boldsymbol{y}[n]|\boldsymbol{f}[n]) \propto \prod_{k=1}^{K} \mathrm{e}^{-\frac{\tau_m}{2}(\boldsymbol{M}[n]\boldsymbol{f}[n] - y_k[n])^2}, \tag{5}$$

where the sensor noise is modeled as white with a variance $1/\tau_m$.

In our approach we do not need any assumption regarding the exact number of sources, their position or strength. We only utilize the prior knowledge that sources are sparsely distributed in the considered environment. In other words, we do not know the exact number of sources, but there are

only few of them and their number is of the same order of magnitude as the number of robots. In the probabilistic setting we are able to introduce prior knowledge according to the Bayes theorem. Here, we make use of a sparse prior and Sparse Bayesian Learning techniques [14] (SBL). SBL applies a hierarchical prior, where the actual prior itself is parametrized by an additionally introduced hyper-parameter $\boldsymbol{\gamma}[n]$ as

$$p(u_c[n]\big|\gamma_c[n]) = N(u_c[n]|0, \gamma_c^{-1}[n]). \tag{6}$$

By the combination of this prior and a hyper-prior $p(\gamma_c[n]) \propto 1/\gamma_c[n]$ the Automatic Relevance Determination (ARD) version of SBL [15] leads to emergence of sparsity in the Bayesian context of $\boldsymbol{u}[n]$. For a more descriptive interpretation, the prior can be compared to a regularization method. Whenever no information is available in a grid cell, the zero centered Gaussian prior will drive the source strength to zero.

Putting all together according to the Bayes Theorem the posterior PDF of our problem becomes:

$$\begin{aligned} &p(\boldsymbol{f}[0]...\boldsymbol{f}[N], \boldsymbol{u}[0]...\boldsymbol{u}[N], \boldsymbol{\gamma}[0]...\boldsymbol{\gamma}[N]|\boldsymbol{y}[0]...\boldsymbol{y}[N]) = \\ &\prod_{n=0}^{N} p(\boldsymbol{y}[n]|\boldsymbol{f}[n]) p(\boldsymbol{f}[n]|\boldsymbol{f}[n-1], \boldsymbol{u}[n]) \\ &\prod_{c=1}^{C} p(u_c[n]\big|\gamma_c[n]) \prod_{c=1}^{C} p(\gamma_c[n]). \end{aligned} \tag{7}$$

In order to evaluate the source strength in a single grid cell, it is possible to calculate the marginal PDF based on the posterior. Generally, the marginal results from integrating over all other variables and parameters. In contrast, in this paper we use a distributed algorithm that enables the multi-robot system to cooperatively calculate all marginal distributions. The marginal PDF for all $u_c[n]$ can be approximated by a Gaussian distribution. So we can use the variance of the PDF $p(u_c[n])$, i.e. the second order central moment, as a gauge for the uncertainty of cell $c$. More precise: the higher the variance the higher the uncertainty regarding the estimated source strength in a cell.

### C. Distributed Implementation

Let us consider how to calculate the marginal PDF of all cells based on the probabilistic formulation of the posterior, since these marginals are the foundation of the uncertainty quantification. Further, we want to calculate them in a distributed fashion in order to take advantage of the multi-agent system. Therefore, in a first step we introduce a graphical representation of our posterior PDF. This representation is used in a second step, where we apply a message passing algorithm to calculate the marginals in a distributed fashion.

*1) Factor Graph Representation:* An interesting property of our posterior PDF (7) is the fact that the function is nicely factorized. There are three factors for each cell and one factor for each measurement. A factorized function can be graphically represented with a Factor Graph (FG) [16]. A FG is an undirected bipartite Bayesian network being composed of value nodes, which represent random variables, and factor nodes, which model functional dependencies between them. (See [17] for a more detailed introduction to FG.) In our case the FG models the relationships between the different random variables $f_c[n]$, $u_c[n]$ and $\gamma_c[n]$. Figure 1 shows the part of the overall graph for one cell and one time instance.

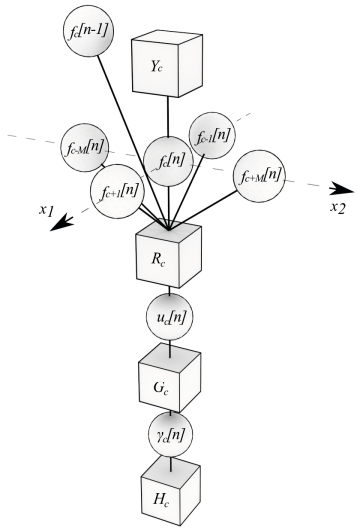



Fig. 1: Factor Graph: This graph represents the part of the posterior PDF associated with a single grid cell. It models the relations between variable nodes (spheres) by factor nodes (cubes).

The node $Y_c$ represents a measurement and it only exists if a measurement was taken in the corresponding cell. The node $R_c$ puts the neighboring concentration values of the grid cell into a relation with the source strength. This relationship arises from our PDE and the numerical approximation. The nodes $G_c$ and $H_c$ model the factors of the posterior PDF that correspond to the parametric prior $p(u_c[n]\big|\gamma_c[n])$ and, respectively, the hyper-prior $p(\gamma_c[n])$.

*2) Message Passing Algorithm:* The FG is in general the foundation of message passing algorithms. In the probabilistic context, these algorithms are a powerful tool to calculate marginal distributions. Mostly, it is used in coding theory for error detection and correction. In this paper we make use of them to calculate the marginal distribution of our posterior PDF (7). More precisely, we make use of the sum product algorithm [18] (also called loopy belief propagation) and variational message passing [19]. Messages are sent between nodes of the factor graph along the edges. There are two possible types of messages: messages from factor nodes to variable nodes and vice versa. The sum product algorithm provides update rules according to which the messages are calculated. In general the outgoing messages of a node are functions of incoming messages (For the specific calculation rules see [17], [19]). By iteratively exchanging messages between nodes, the outgoing messages of variable nodes converge to the marginal distribution of the corresponding variable or parameter. For messages corresponding to our hierarchical prior in the graph, analytical tractability is not given. However, for this part of the graph we use variational

message passing (VMP) [19] to circumvent the issue by analytical approximation techniques. The messages themselves represent beliefs, i.e. probabilistic distributions. Since we designed our posterior PDF properly and thanks to the VMP, all messages stay in the same class of distribution. E.g. the outgoing message of a node $u_c[n]$ is always Gaussian distributed and therefore could be parametrized by only two values: mean and variance. We stress that the overall factor graph contains a lot of edges and a lot of message-updates have to be calculated. But all updates can be calculated in closed-form and correspond to relatively simple calculations.

Moreover, the main advantage of the message passing algorithms is the fact that it could be easily implemented in a distributed fashion. Even so many messages have to be calculated, they can be updated in random order or parallel. Thus, the framework is very suitable for a distributed implementation.

Actually, we divide the overall factor graph in different parts. A simplified version of the overall factor graph and the partitioning of this graph are shown in Fig. 2. The different parts correspond to different 2D region of our environment. Each region is assigned to one robot of our multi-agent system and each robot is able to calculate all messages of its own part of the graph. The robot only has to exchange the messages along the border of its partition (red arrows) with its neighbors. By iteratively calculating and exchanging messages the outgoing messages of $u_c[n]$ converges to the marginal PDF. Based on the variances each robot proposes a certain number of relevant points in the region it is responsible for to all other robots.

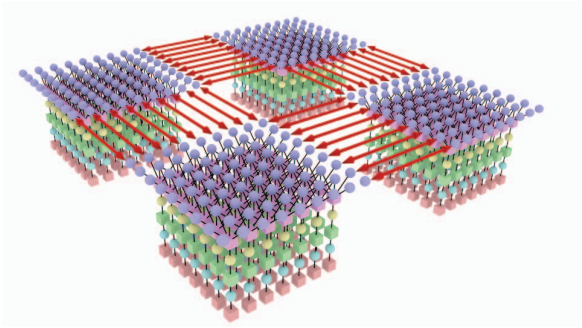


Fig. 2: Distributed Factor Graph: This figure illustrates the overall factor graph. This is a simplified version without measurement nodes and time dependencies. The graph is spatially split into four regions, where the big arrows represent messages sent between different regions.

### D. Exploration Procedure

Our exploration strategy utilizes the uncertainty quantification of the estimated source strength for each cell $c \in C$ with $C$ being the set of all cells in our environment. The cells are rated according to the inverse variance, i.e. precision $\tau_c[n]$, of the source strength marginal $p(u_c[n]) \propto N(\hat{u}_c[n], \tau_c[n])$. A set $P$ of cells with the lowest precisions - i.e. with the highest uncertainties - serves as a proposal for new way points for the robots. Therefore, each robot $a$ proposes $K$ cells from the region $C_a \subseteq C$ of the environment it is responsible for:

$$\tau_{c_1} \leq \tau_{c_2} \leq ... \leq \tau_{c_K} \leq \tau_{c_{(K+1)}} \leq ...; \tau_{c_i} \in C_a$$

$$P_a = \bigcup_{i=1}^{K} c_i \tag{8}$$

The $K$ most uncertain points of each region are combined to the set $P = \{c \in \bigcup P_a\}$ for the whole environment. From this set $P$ each robot selects one way point. We like to emphasize that therefore the movement of a robot is not limited to the region it is responsible for with respect to the calculations. Actually, the robot can choose from the whole set $P$. Their choice is based on two criteria: i) Each robot adds to the precision of a proposed way point a penalty based on its distance to the way point and selects the one with the lowest value. In this way the robots not only favor points with high uncertainty but also those which are close-by. ii) If another robot is already on its way to a point, this point is neglected. Therefore it is necessary that a robot communicates its decision to all other robots.

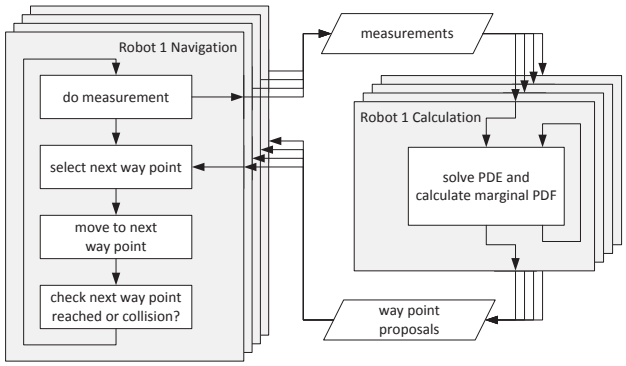



Fig. 3: Exploration Procedure: The exploration is implemented with two main loops. The right loop solves the PDE and produces new way point proposals, the left one controls the individual robots.

After the decision is made, a robot moves to the selected way point. To avoid collisions with other robots, we implemented a reactive collision avoidance mechanism. On its way, the robot monitors the distance to all other robots based on received position information. If the distance drops below a defined safety threshold, the robot stops and selects another way point from the proposals that would increase the critical distance. Finally, when a robot reaches its goal, it takes a measurement which is incorporated in the probabilistic solution of the PDE.

The overall procedure is depicted in Fig. 3. As can be seen, the robotic navigation and solving the PDE are actually two separate loops. The loops are connected by the data exchange of proposed way points and measurements. We would like to stress that in this way the navigation part of each robot could be realized in an asynchronous fashion, where no robot has to wait for results of others.

## III. Evaluation

In order to evaluate our approach we designed a hardware-in-the-loop experiment. This means that we use a real robotic system but we simulate the gas dispersion taking place in the environment of the robots.

### A. System Setup

We developed a small robot for our experiments. The main part is a Raspberry Pi 2, a low power single-board computer with Linux OS (900MHz quad-core ARM Cortex-A7 CPU, 1GB RAM). From this computer it is possible to send velocity commands to a micro-controller that implements a velocity controller for two motors driving the tracks of the robot. The experiments were done in a laboratory with the commercial optical tracking system. This system is able to track active infra-red LEDs on the robot and provides the current position and orientation of each robot with high accuracy. On the robot's computer a position controller is implemented that compares the actual pose from the tracking system and the nominal pose demanded by the exploration strategy. We employ five robots. Their computers are connected to an 802.11 wireless LAN communication system. By this, they get their own position data and are able to exchange messages among each other. For the software implementation and the inter-process communication we make use of the Robot Operating System (ROS) [1].

The gas diffusion is simulated for a two dimensional case. The data are generated according to equation (1). Whenever a measurement is demanded by the exploration procedure for one robot, this equation is evaluated at the current position of the robot. Additionally, we disturb the measurement by the additive white noise $\xi$. For the evaluation of the PDE a Finite Volume Method solver [20] is used. At the boundary we have chosen a Dirichlet boundary condition $f(x,t) = 0$, except for the right border, where we use a Neumann boundary condition $\frac{\partial f(x,t)}{\partial x} = 0$. For the virtual gas simulation we considered the concentration and source strength unit-less. The discrete grid size, the time difference between two discrete time stamps and the diffusion coefficient $\kappa$ are set to $1$ in the simulation. However, later on the concentration field is fitted to our laboratory with a scale of $6m$ times $2.4m$.

### B. Results

We evaluate the proposed exploration strategy in comparison against exploration with a predefined sweeping trajectory. The sweeping trajectories are generated by dividing the environment into five equal regions and fitting a meander into each region. In this way the measurements will fully cover the whole environment after a certain time, i.e. each grid cell is measured at least once. This strategy might be reasonable, if no prior knowledge or model assumptions are available. We compare the performance by means of speed and quality of the estimates. Regarding speed we consider the number of measurements needed to converge. In order to see how well the spatially distributed sources $\boldsymbol{u}[n]$ are identified,

[1] http://www.ros.org/

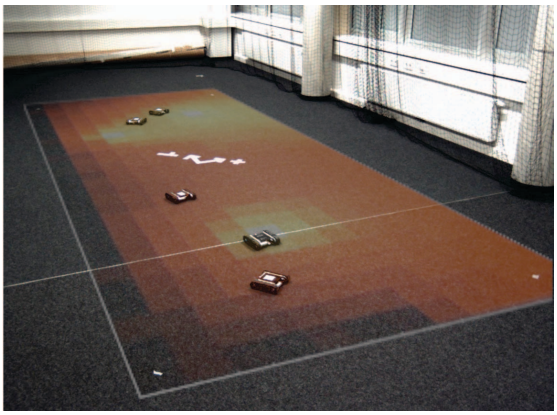


Fig. 4: Lab Environment: The picture shows our lab during an experiment. The simulated concentration field is projected to the ground in a post-processing step. A video of an experiment can be found here: https://youtu.be/UJYwdzrDTL4

we use the so-called earth mover's distance (EMD) measure, which is an analog of a Wasserstein metric for discrete distributions [21]. The EMD measures the effort to "displace" one distribution onto another one and is particularly useful for comparing sparse functions. Specifically, we compare the estimated vector $\boldsymbol{u}[n]$ with the ground truth vector $\hat{\boldsymbol{u}}$ with all elements set zero except for the three cells containing a source ($\hat{u}_c = 1.0$ at $x = -0.2m, y = -1.8m$; $\hat{u}_c = 1.0$ at $x = -0.6m, y = 1.0m$; $\hat{u}_c = 0.8$ at $x = 0.4m, y = 1.6m$).

The results are shown in Fig. 5 with the aid of an example experiment. Fig. 5a and 5b visualize the trajectories of the meander and the proposed exploration strategy. The proposed exploration strategy is adaptive. In other words it reacts to measurements. Therefore, the trajectory is not deterministic and not predictable. The trajectory in Fig. 5b is an example and will look different in another simulation run, because of different initial conditions and randomness caused by measurement noise. The trajectories are plotted as an overlay above the simulated concentration field at the time, when the sources are correctly identified. From the figure it gets obvious that this distribution is driven by three sources located at the concentration peaks. Fig. 5c depicts the performance of both strategies. The curves plot the difference between the estimated source distribution $\boldsymbol{u}[n]$ and the true source distribution $\hat{\boldsymbol{u}}$ in relation to the number of collected measurements.

With the meander trajectory the multi-agent system was able to identify the source distribution after approximately 340 measurements in our example. This indicates the step in the convergence curve towards zero in Fig. 5c. In general, the performance of the meander highly depends on the position of the sources. If they are already covered at the beginning of the trajectory, of course fewer measurements are needed. However, to be conservative the worst case has to be considered and this means a full coverage of the region. In our example, 360 measurements would be necessary for that. In contrast, the curve for the proposed exploration strategy converges with only 230 measurements in Fig. 5c. This indicates that robots were able to identify

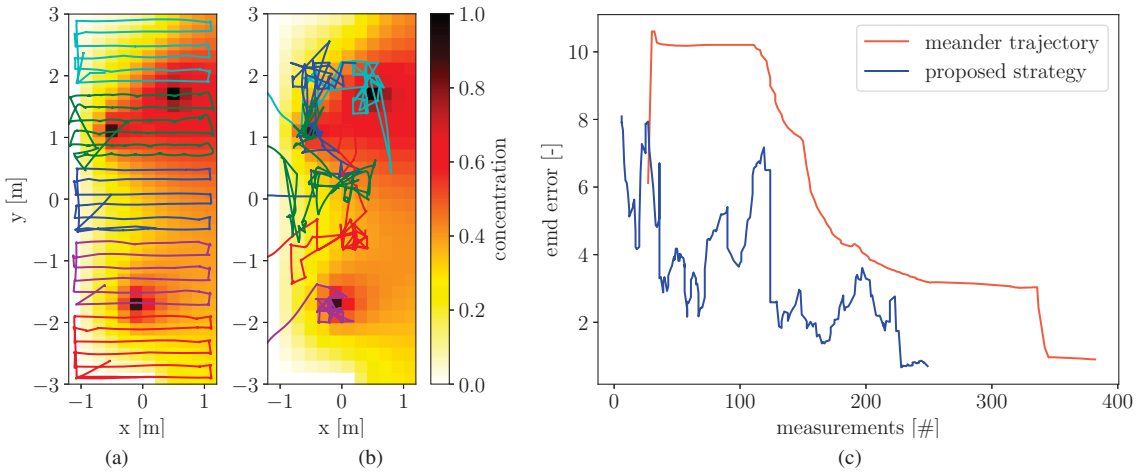



Fig. 5: Results: The figure compares the performance of the meander trajectory (a) and the proposed exploration strategy (b). The trajectories superimpose the simulated concentration field. In (c) the error regarding the estimated source distributions are plotted for both cases. The proposed exploration is able to identify the sources with less measurement than the meander.

the sources with fewer measurements. As can be seen from the trajectory in Fig. 5b the measurements are concentrated around the source locations. These measurements contain more information regarding the sources. This is the reason for a better performance.

Additionally, the hardware-in-the-loop experiments enable us to analyze other performance indicators of our algorithm and system properties. For example we can measure the gross data rates containing all overheads caused by ROS, OS, TCP etc. Concretely, in our case a data rate of less than $70kBytes/s$ is required for a communication link between two robots. This provides us with the information to create specifications for a communication system for future real-world experiments. Similarly, we can investigate the processor load of the on-board computers caused by the algorithm. In particular, the on-board computers were able to generate way point proposals with a update-rate of $1.0Hz$.

## IV. Discussion and Conclusion

The results of the experiments have shown that a model-based exploration is of advantage for sampling a gas diffusion process. By an intelligent exploration strategy, the number of required measurements to identify gas sources can be reduced. This property is favorable to applications, where a measurement is expensive or consumes a lot of time.

The potentials of the presented approaches arise from the uncertainty driven strategy for taking new measurements in combination with the assumption that the sources are sparsely distributed in the environment and their number is small. This assumption is encoded with a prior PDF that assumes a source to have zero amplitudes with unknown variance. Under this assumption each found source with non-zero posterior amplitude effectively "contradicts" this prior assumption; as such the uncertainties of the estimated sources in the corresponding regions grow. As a consequence, the robots concentrate their measurements on informative regions around the sources according to the proposed exploration strategy. Additionally, this implies a need to employ a multi-robot system with more agents as gas sources, since single robots may got stuck in a neighborhood of a source and do not discover new sources. Multiple rovers are able to effectively reduce the uncertainty around the sources by multiple simultaneous measurements, and individual robots can "escape" from the source location.

For future development of the method a key problem to be addressed is a mismatch between the used dispersion model and real gas dispersion under the influence of possible turbulence and advection mechanisms that are not explicitly represented with model (1). To account for these effects several tactics can be explored. In particular, advection-based diffusion can be incorporated directly into the PDE through a convection term. In the presented framework, this is easy to achieve, since only equation 1 and 2 have to be modified appropriately, with the rest of the approach left intact. Turbulence, on the other hand, can be treated as a random effect and accounted for in the probabilistic formulation with the model uncertainty $\tau_s$ in (4).

In summary, let us mention that realistic gas dispersion problems are quite complex dynamical processes. From a practical perspective having a very complex model that adequately represent reality might lead to computationally very complex inverse problems. Instead, simpler models, like the one used in this work, can be seen as a numerically feasible approximation, which can be estimated based on concentration measurements in an adequate (in the context of robotic exploration) amount of time.

## Acknowledgment

This work has partly been supported within H2020-ICT by the European Commission under grant agreement number 645101 (SmokeBot)

## REFERENCES


[1] H. Ishida, Y. Wada, and H. Matsukura, "Chemical sensing in robotic applications: A review," *IEEE Sensors Journal*, vol. 12, no. 11, pp. 3163–3173, Nov 2012.

[2] A. Marjovi and L. Marques, "Multi-robot odor distribution mapping in realistic time-variant conditions," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3720–3727, May 2014.

[3] M. A. Demetriou and D. Ucinski, "State Estimation of Spatially Distributed Processes Using Mobile Sensing Agents," *American Control Conference (ACC)*, pp. 1770–1776, 2011.

[4] D. J. C. MacKay, "Information-Based Objective Functions for Active Data Selection," *Neural Computation*, vol. 4, no. 4, pp. 590–604, 1992.

[5] F. Pukelsheim, *Optimal Design of Experiments*. John Wiley & Sons, Inc., 1993.

[6] D. Uciński, *Optimal Measurement Methods for Distributed Parameter System Identification*. CRC Press, 2004.

[7] A. J. Lilienthal, M. Reggente, M. Trincavelli, J. L. Blanco, and J. Gonzalez, "A statistical approach to gas distribution modelling with mobile robots-the kernel dm+ v algorithm," *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 570–576, 2009.

[8] J. G. Monroy, J.-L. Blanco, and J. Gonzalez-Jimenez, "Time-variant gas distribution mapping with obstacle information," *Auton. Robots*, vol. 40, no. 1, pp. 1–16, Jan. 2016.

[9] P. P. Neumann, S. Asadi, A. J. Lilienthal, M. Bartholmai, and J. H. Schiller, "Micro-Drone for Wind Vector Estimation and Gas Distribution Mapping," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 1, pp. 50–61, 2012.

[10] P. Whaite and F. Ferrie, "Autonomous exploration: driven by uncertainty," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 193–205, mar 1997.

[11] T. Wiedemann, C. Manss, and D. Shutin, "Multi-Agent Exploration of Spatial Dynamical Processes under Sparsity Constraints," *to appear Journal Autonomous Agents and Multi-Agent Systems*, 2017.

[12] J. Crank, *The Mathematics Of Diffusion*, 2nd ed. Oxford: Clarendon Press, 1975.

[13] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2004.

[14] M. E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *Journal of Machine Learning Research*, 2001.

[15] D. Wipf and S. Nagarajan, "A New View of Automatic Relevance Determination," *Advances in Neural Information Processing Systems 20*, pp. 1625–1632, 2008.

[16] F. R. Kschischang, B. J. Frey, and H. a. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[17] H.-a. Loeliger, "An Introduction to Factor Graphs," *Signal Processing Magazine, IEEE*, vol. 21, no. 1, pp. 28 – 41, 2004.

[18] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Francisco: Kaufmann, 1988.

[19] J. Winn and C. M. Bishop, "Variational Message Passing," *Journal of Machine Learning Research*, vol. 6, pp. 661–694, 2005.

[20] A. Logg, K.-A. Mardal, and G. Wells, *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*. Springer Publishing Company, Incorporated, 2012.

[21] Y. Rubner, C. Tomasi, and L. J. Guibas, "A Metric for Distributions with Applications to Image Databases," *Computer Vision, 1998. Sixth International Conference on*, pp. 59–66, 1998.

# Bayesian Gas Source Localization and Exploration with a Multi-Robot System Using Partial Differential Equation Based Modeling

Author1 and Author2
Affiliation 1
Address 1

Author3, Author4 and Author5
Affiliation 2
Address 2

***Abstract*—Model based approaches, such as those that use partial differential equations (PDE), lend themselves to gas distribution mapping and gas source localization. Moreover, they also permit constructing intelligent sampling strategies. However, a realistic mathematical model of gas dispersion is complex and computationally expensive to solve. This is especially the case for inverse problems, where sources are estimated based on concentration measurements. In this paper, we propose a probabilistic model based on a diffusion PDE to approximate the complex behavior of gas dispersion. This model is used (i) to identify the sources, using ideas from Sparse Bayesian Learning, and (ii) to guide a multi-agent robotic system to measurement locations, which assists the source localization. The potential of the approach is shown in experiments, where laminar gas plumes are simulated using an open-source CFD-based filament gas dispersion simulator. The exploration is carried out using multiple real robots implementing the proposed algorithm.**

## I. Introduction

Deploying robotic platforms in hazardous environments for exploration or monitoring has significant practical advantages as it allows avoiding potential threats for human operators. This is particularly important in case of disaster scenarios when a gas concentration has to be mapped, or there is a need to identify unknown gas sources. In this paper we propose a model-based exploration strategy for a multi-robot system that permits detection and localization of unknown gas sources based on agent-mounted sensors. Unfortunately, physical mechanisms causing gas propagation are not trivial, and in case of turbulence can even exhibit non-deterministic and chaotic behavior [3], [9]. Nonetheless, for on-line mapping scenarios a simplified approximation of the physical phenomenon with low computational complexity might be of a great use. To this end in this paper we investigate the capability of a diffusion partial differential equation (PDE) to approximate spatial gas dynamics for the purpose of identifying sources that drive the gas propagation.

The addressed exploration task is closely related to a so called optimal sensor placement problem. Known methods use linear-quadratic control design techniques, which try to optimize the observer performance by adjusting the sensor location [2]. Other popular methods are based on optimal experimental design [8] and probabilistic [4] or Bayesian approaches [7]. In our work we build upon these ideas and propose an exploration strategy that minimizes the uncertainty of source localization following a criterion similar to an A-optimality [1]. Additionally, similar to [6] we exploit the assumption that the sources causing gas dispersion are sparsely distributed and use sparse Bayesian learning techniques to model this. By using a realistic gas dispersion simulator [5], we demonstrate the effectiveness and performance of the proposed method with a hardware-in-the-loop simulation with five mobile robots in a laboratory environment. The analysis of the experimental data demonstrates the effectiveness of the proposed modeling approach and of the exploration strategy in a multi-agent environment.

## II. Gas dispersion simulator

As previously stated, we validate our proposed algorithms with a state-of-the-art gas dispersion simulator[1], originally presented in [5]. The simulator is build upon the Robot Operating System (ROS) framework[2], allowing in this way to simulate gas dispersion, mobile robots and gas sensing technologies in a unified tool. The gas dispersion simulation engine is particle-based. This means that gas concentrations depend on the number of particles, which are affected by diffusion, turbulence, advection, and gravity. Wind information is integrated in the simulation engine as a series of time snapshots that can be computed with an external fluid dynamics tool. In addition, response models for gas sensing devices such as Metal Oxide (MOX) sensors can be integrated in the framework and simulated on board of stationary or mobile platforms. In this work, we neglect the airflow by setting the wind vector field to zero.

## III. Model based exploration

### A. Probabilistic model

From a practical perspective, approximating complex gas dynamics with simple models can be of an advantage. Here we use a 2D diffusion model that can be formally described with a linear parabolic partial differential equation (PDE):

$$\frac{\partial f(\boldsymbol{x},t)}{\partial t} - \kappa \Delta f(\boldsymbol{x},t) = u(\boldsymbol{x},t). \tag{1}$$

[1]The simulator is available at the following repository (link removed for the double-blind review)

[2]http://www.ros.org/

Essentially, (1) models the dynamic behavior of some gas concentration $f$ at a 2D position $\boldsymbol{x}$ with respect to time $t$ given the constant diffusion coefficient $\kappa$. The term $u(\boldsymbol{x}, t)$ on the right hand side represents a spatio-temporal inflow of material; in this work we aim to identify this process.

In order to solve a PDE, numerical approximation methods are often the instrument of choice. To this end we begin with a classical Finite Difference Method [10] which "transforms" (1) into a finite dimensional linear system by appropriately discretizing both space and time:

$$\boldsymbol{A}\left[\boldsymbol{f}[n], \boldsymbol{f}[n-1], \boldsymbol{u}[n]\right]^T = 0; \tag{2}$$

In other words, we divide our region of interest into a finite number of grid cells and consider temporal evolution of concentration values in each cell at discrete time steps $n$. Thus, the vector $\boldsymbol{f}[n]$ contains the concentration in all grid cells at time stamp $n$; likewise, $\boldsymbol{u}[n]$ aggregates the source strength for all grid cells. Both quantities are in general unknown and are estimated numerically using sample observations of the gas concentration at different locations and time instances. Yet we stress that the main interest for us is the estimation of the source distribution $\boldsymbol{u}[n]$ as this information guides the dynamic of the whole process.

To facilitate numerical estimation of the PDE parameters we relax (2). This reflects the fact that the real gas dispersion mechanisms are more complex and not always deterministic; thus, we assume that (2) holds only with a certain precision, with deviations from zero being random. This permits us taking a probabilistic approach toward PDE solution by treating all variables as random. This results in a probabilistic formulation, where we can define a conditional probabilistic density function (PDF) for concentrations $\boldsymbol{f}[n]$ as follows:

$$p(\boldsymbol{f}[n]|\boldsymbol{f}[n-1], \boldsymbol{u}[n]) \propto \mathrm{e}^{-\frac{\tau_s}{2}\left\|\boldsymbol{A}[\boldsymbol{f}[n], \boldsymbol{f}[n-1], \boldsymbol{u}[n]]^T\right\|^2} \tag{3}$$

which is a relaxation of (2) that allows it to deviate from 0 within the precision $\tau_s$. In practice the gas concentrations are not observed directly, but are rather measured with a sensor. The actual measurement can be represented as $\boldsymbol{y}[n] = \boldsymbol{M}[n]\boldsymbol{f}[n] + \boldsymbol{\epsilon}_m$, where $\boldsymbol{M}[n]$ selects grid cells visited by the robots, and $\boldsymbol{\epsilon}_m$ models a spatially and temporally white measurement noise with precision $\tau_m$. The corresponding concentration likelihood can then be expressed as

$$p(\boldsymbol{y}[n]|\boldsymbol{f}[n]) \propto \mathrm{e}^{-\frac{\tau_m}{2}\|\boldsymbol{M}[n]\boldsymbol{f}[n]-\boldsymbol{y}[n]\|^2}. \tag{4}$$

Now we assume that the sources $\boldsymbol{u}[n]$ are sparse in space, i.e., the gas distribution is only driven by a few unknown discrete sources; neither their number nor their location is known. These assumptions can be incorporated using Sparse Bayesian Learning (SBL) techniques (see [11], [12] for more details). SBL assumes a parametric prior $p(\boldsymbol{u}[n], \boldsymbol{\gamma}[n])$, with parameters $\boldsymbol{\gamma}[n]$ effectively determining if an element in $\boldsymbol{u}[n]$ is 0 or not.

Using Bayes theorem, the inference problem reduces to the calculation of the posterior PDF

$$p(\boldsymbol{F}, \boldsymbol{U}, \boldsymbol{\Gamma}|\boldsymbol{Y}) \propto \prod_{n=1}^{N} p(\boldsymbol{y}[n]|\boldsymbol{f}[n])p(\boldsymbol{f}[n]|\boldsymbol{f}[n-1], \boldsymbol{u})p(\boldsymbol{u}[n], \boldsymbol{\gamma}[n]) \tag{5}$$

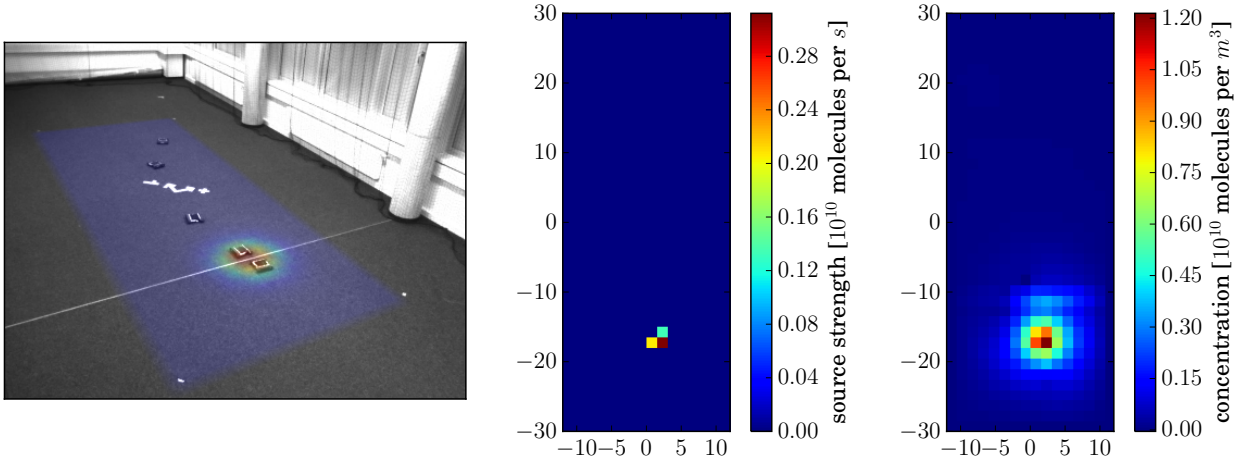
where, with some abuse of notation, we use the capital letters to define the aggregation of the corresponding variable over the whole observation time, i.e., $\boldsymbol{F} = [\boldsymbol{f}[0], \ldots, \boldsymbol{f}[N-1]]$. Effectively, (5) defines the most likely gas concentration $\boldsymbol{F}$ and source locations $\boldsymbol{U}$ given all available measurements. Note that SBL not only "sparsifies" vector $\boldsymbol{u}[n]$, but also regularizes the solution. This is important as in early stages, when the number of measurement is smaller then the number of unknowns – $\boldsymbol{f}[n]$, $\boldsymbol{u}[n]$ and $\boldsymbol{\gamma}[n]$ – the resulting optimization problem is ill-posed.

### *B. Exploration strategy*

While movements of the agents can be arbitrary, it is advantageous when their trajectories are selected to facilitate estimation of the sources $\boldsymbol{u}[n]$. This is the objective of the exploration strategy discussed below. Note that the probabilistic approach permits us to quantify the uncertainty of the source strength in each grid cell. In our approach we take into account the individual marginal PDF for each element of $\boldsymbol{u}[n]$. Specifically, that a high variance of the estimated source marginal PDFs implies a high uncertainty. This quantity can be used to rate all cells in the region of interest, and cells with highest uncertainties are provided to the multi-agent system as proposals for new measurement locations. Each robot selects a cell based on the uncertainty value and additionally weighted by the distance to it. In other words, the robots not only prefer cells with high uncertainty but also cells which are close to their current positions. When a target is reached, a measurement is taken, which maximally reduces the uncertainty of the corresponding grid cells. This can be shown to be equivalent to a so called A-optimality criteria. Then the model is updated by (i) adapting the measurement vector $\boldsymbol{y}[n]$ and matrix $\boldsymbol{M}[n]$, and (ii) performing inference of the current states $\boldsymbol{f}[n]$ and sources $\boldsymbol{u}[n]$. After that the whole scheme is repeated. By successively improving each individual cell, this exploration strategy minimizes the overall uncertainty of the source distribution. Note that practically some collision avoidance mechanism is needed. To this end in the discussed experiments a simple scheme to avoid collisions is realized on a lower control layer.

## IV. Results

For evaluation purposes, we used the gas dispersion simulator to generate a dynamic gas distribution driven by one source in an area of $20m$ times $60m$. This area is virtually scaled down by factor 10 to fit into our laboratory environment. Whenever a robot is triggered to collect a measurement, the simulated gas concentration at the robots position is used as a synthetic measurement for the exploration algorithm. The diffusion coefficient $\kappa$ of the PDE model is estimated in a




(a) (b) (c)

Fig. 1. Snapshot of the exploration procedure at a certain time stamp (marked in Fig 2): In (a) a picture of the multi-robot system is shown overlayed by the simulated gas distribution. Plot (b) displays the estimated source distribution and (c) the estimated gas concentration.

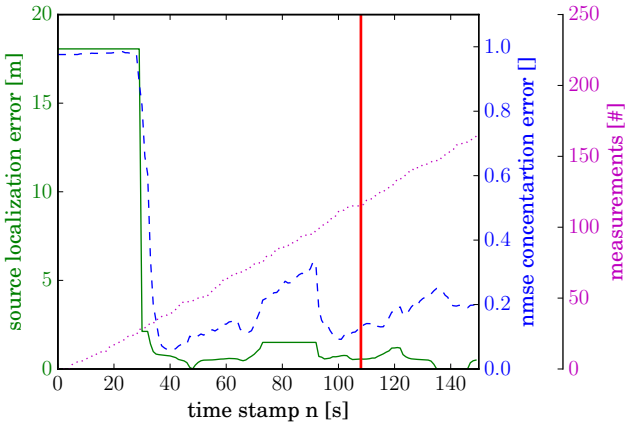



Fig. 2. Performance of the exploration: The source localization error is calculated by comparing the center of mass of the estimated source distribution with the actual position of the source used in the simulation. The concentration estimate is assessed by the normalized mean square error (nmse) with respect to the concentration given by the simulator.

pre-processing step to best approximate the gas dispersion. Fig. 1 visualizes the result of the exploration at a certain time step, whereas Fig. 2 depicts the performance of the estimates over time. Although the difference between the simulated gas concentration and the estimated concentration based on the simplified model is rather high due to the oversimplifying diffusion approximation, the estimated source strength distribution is quite accurate and the robots are able to localize the source in a reasonable time frame with good accuracy.

## V. Conclusion

The paper showed that it is possible, to some extend, to approximate a complex gas dispersion phenomenon by a simplified model. We exploited this model to design an exploration strategy for a multi-robot system that facilitates the localization of sources driving the phenomenon. In future work, we will extend the approach to better cope with advection mechanisms by introducing a convection term into the PDE.

## References


[1] A. Alexanderian, N. Petra, G. Stadler, and O. Ghattas, "A-optimal design of experiments for infinite-dimensional Bayesian linear inverse problems with regularized l0-sparsification," *SIAM Journal on Scientific Computing*, vol. 36, no. 5, p. 27, 2013.

[2] M. Demetriou, "Numerical investigation on optimal actuator/sensor location of parabolic PDEs," *Proceedings of the 1999 American Control Conference*, vol. 3, no. June, pp. 1722–1726, 1999.

[3] V. Hernandez, A. J. Lilienthal, P. P. Neumann and M. Trincavelli, "Mobile robots for localizing gas emission sources on landfill sites: is bio-inspiration the way to go?," Front. in Neuroeng., 4: 0, 2012.

[4] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, 1st ed. Springer-Verlag New York, 2005.

[5] A. Khaliq, S. Pashami, E. Schaffernicht, A. J. Lilienthal and V. Hernandez, "Bringing Artificial Olfaction and Mobile Robotics Closer Together - An itegrated 3D Gas Dispersion Simulation in ROS," Proc. of the 16th Intl. Symposium on Olfaction and E-Noses (ISOEN), 2015.

[6] K. Kunisch, K. Pieper, and B. Vexler, "Measure Valued Directional Sparsity for Parabolic Optimal Control Problems," *SIAM Journal on Control and Optimization*, vol. 52, no. 5, pp. 3078–3108, jan 2014.

[7] D. J. C. MacKay, "Information-Based Objective Functions for Active Data Selection," *Neural Computation*, vol. 4, no. 4, pp. 590–604, 1992.

[8] F. Pukelsheim, *Optimal Design of Experiments*. John Wiley & Sons, Inc., 1993.

[9] Roberts, P., and Webster, "Turbulent Diffusion. Environmental Fluid Mechanics Theories and Application", Reston, VA: ASCE Press,2002.

[10] Strikwerda, J.C.: Finite Difference Schemes and Partial Differential Equations, 2 edn. Society for Industrial and Applied Mathematics, Philadelphia (2004)

[11] M. E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *Journal of Machine Learning Research*, 2001.

[12] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *Signal Processing, IEEE Transactions on*, vol. 52, no. 8, pp. 2153–2164, 2004.