ISSN 0280-5316
ISRN LUTFD2/TFRT--5872--SE

Robotic Gas Source Localization in an Industrial Environment

Erik Persson

Department of Automatic Control
Lund University
October 2010

Lund University **Department of Automatic Control** **Box 118** **SE-221 00 Lund Sweden**	*Document name* MASTER THESIS *Date of issue* October 2010 *Document Number* ISRN LUTFD2/TFRT--5872--SE
Author(s) Erik Persson	*Supervisor* David Anisi ABB AS, Oslo Norway Anders Robertsson Automatic Control, Lund Sweden Rolf Johansson Automatic Control, Lund Sweden (Examiner) *Sponsoring organization*

Title and subtitle

Robotic Gas Source Localization in an Industrial Environment (Robotbaserad gasläckagelokalisering i industriell miljö)

Abstract

Gas leaks are an important safety issue in oil and gas production. For example, natural gas often contains large portions of hydrogen sulfide, a gas that is lethal to humans in concentrations as low as 0.1%. In addition natural gas itself is explosive. During the past fifteen years, a considerable number of studies have been made into how to detect and localize gas leaks. Equipped with sensors measuring the point concentration of specific substances, a variety of mobile robots and algorithms have been looking for gas sources indoors and outdoors, underground and under water, in airless conditions and in windy dittos. Due to the complexity of turbulence and the limitations of gas sensors, robotic gas source localization has turned out to be complicated and so far it has not made its way to large scale real world applications. This study is an attempt to bring robotic gas source localization a bit closer to that. Three algorithms, carefully chosen from the literature, are adapted to an industrial environment. In addition, two novel strategies are derived from the original ones through combination of them. A comparative study between the five algorithms is made where their performances are evaluated and compared.
This study has been conducted within a project of ABB in Oslo that investigates how industrial robots can be used in an oil and gas-context.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title 0280-5316		*ISBN*
Language English	*Number of pages* 66	*Recipient's notes*
Security classification		

http://www.control.lth.se/publications/

Acknowledgements

The work presented in this thesis was performed at the Strategic R&D Unit for Oil, Gas and Petrochemicals at ABB in Oslo, Norway between March and October 2010.

First and foremost, I would like to thank David Anisi, my advisor at ABB for his support, for his guidance to the company and the world of academia and for always having time for me. I would also like to thank my advisor at Lund University, Anders Robertsson, for his support and for providing the contact with ABB in the first place. Further gratitude goes to my colleagues in the project, especially Johan Gunnar and Clint Heyer, for helping me out with the robot control system. My project leader Charlotte Skourup and department manager John Pretlove have also been supportive and I am happy for the interest they have showed in my work. The rest of the department too deserves my gratitude for taking well care of me and treating me like a regular employee from the first day. I would also like to thank Niklas Jansson for his help with data transmission and circuit production. Finally, I would like to thank my girlfriend for her support and for her fantastic idea of going to Norway. She has also done a tremendous job in correcting many minor errors in this thesis.

Contents

Chapter 1

Introduction

1.1 Background

Gas source localization is the concept of locating the source of a chemical substance spreading in the environment. It is a vital skill of many animals and humans too are capable of following odor trails. During the past fifteen years, a considerable number of studies have been made into how this capability can be given to robots. Equipped with sensors measuring the point concentration of specific substances, a variety of mobile robots and algorithms have been looking for gas sources indoors and outdoors, underground and under water, in airless conditions and in windy dittos. Due to the complexity of turbulence and the limitations of gas sensors, robotic gas source localization has turned out to be complicated and so far it has not made its way to large scale real world applications. This study is an attempt to bring robotic gas source localization a bit closer to that. Three algorithms, carefully chosen from the literature, are adapted to an industrial environment. In addition, two novel strategies are derived from the original ones through combination of them. A comparative study between the five algorithms is made where their performances are evaluated and compared. In contrast to previous studies, an industrial robot is used as platform throughout this thesis.

1.2 Motivation

Gas leaks are an important safety issue in oil and gas production. For example, natural gas often contains large portions of hydrogen sulfide, a gas that is lethal to humans in concentrations as low as 0.1%. In addition natural gas itself is explosive. There are also environmental issues involved as many of the substances in question are strong green house gases. After the recent catastrophe in the Gulf of Mexico [1], matters such as these have come in an even greater focus.

At the same time, there is a movement within the oil and gas industry towards more extensive automation. As the most accessible resources are being depleted, future facilities are going to be constructed in more remote locations and in more hostile environments. Together with a wish of increasing the safety of the personnel, this creates an interest in solutions where as little human on-site interaction as possible is needed. This study has been conducted within a project of ABB in Oslo that investigates how industrial robots can be used in an oil and gas-context. If this project fulfills its goals, industrial robots will be present in future oil and gas facilities, available to conduct gas source localization and numerous other tasks.

1.3 Method

To fulfill the objectives of this study, division was made into several subtasks:

- Literature study and search for suitable algorithms.
- Construction and performance testing of a sensor circuit.
- Implementation and simulation of algorithms.
- Adaptation of the algorithms to the real laboratory infrastructure and environment.
- Testing and comparison of the algorithm performance in the real environment.

1.4 Thesis Outline

In Chapter 2, previous work in the area of robotic gas source localization is summarized and the selected algorithms are presented. Chapter 3 describes the hardware, software and environment in which the algorithms are evaluated. In Chapter 4, the implementation, simulation and adaptations of the algorithms are described. Chapter 5 contains the results of the different algorithms and a comparison between them. In Chapter 6, the conclusions of this study are presented.

Chapter 2

Previous Work

This chapter summarizes the current state-of-art for robot odor localization and briefly describes different algorithms from the literature. Three algorithms, representing widely different approaches, were identified and chosen for practical evaluation and comparison. For the purpose of better understanding later chapters in this thesis, these algorithms are described in greater depth than other algorithms.

During the last 15 years, robotic odor localization has been a prominent research area [2] with numerous practical studies. Even though different environments and platforms have been used, the typical practical study has been made on a mobile robot [3]. Intended scenarios have usually been to locate bombs, earthquake victims or other odor-emitting objects in unknown surroundings. However, there have been no practical studies in industrial environments and specifically not in process industry settings. These surroundings differ from those of earlier studies for example in that networks of pipes and other infrastructure can be expected to distort the airflow. Further, the positions of obstacles and possible gas sources, such as tanks and pipes containing the substance in question, are likely to be known. In addition, industrial robots have never before been used as platforms for gas source localization and it remains to be investigated how the algorithms developed for vehicle-like robots perform on industrial ones. Exact robot localization, strong processing power and capability of three dimensional movements is much more easily accessed on an industrial robot than on a traditional vehicle-platform. Thus other algorithms than usual may be attractive. Further, very few comparative studies between gas source localization algorithms have been made at all and there seem to be no consensus on what strategies to prefer, even on a mobile platform.

Most of the proposed odor localization techniques can be divided into purely reactive ones and those who collect data over time to estimate the source location with the help of an inner dissipation model. This chapter follows that classification and reactive strategies are described in Section 2.1 while estimation/model-based strategies are described in the following Section 2.2.

2.1 Reactive Strategies

2.1.1 Gradient Following/Hill Climbing Techniques

The basic idea of this technique is to compare the values of two sensors and then continue the search in the direction of the sensor that indicates the higher concentration. This simple approach works fine when diffusion is the dominating mean of gas transportation, such as in underground environments [2]. However, in free air, turbulence will almost always dominate over diffusion [4]. This means that a pure gradient follower will have a hard time tracking the gas-plume. The local gradient will often point away from the source when patches of higher concentrations are passing by [5]. The mean gradient usually points in a good direction but being an average, it takes a long time to measure. There is also a risk of ending up in a local concentration maximum, for example in a corner [6].

To improve results, most "hill climbing" robots have been fitted with weather vanes to tell the direction of the airflow [3]. As a detected odor can be assumed to originate from an upwind direction this information is useful. The wind direction can either be weighted together with the direction of the estimated gradient to give the desired heading of the robot, or other techniques can be used [7]. However, even with knowledge of the wind direction, the search remains difficult, especially when starting far away from the source where the gradient is weak. To reach some level of robustness, "hill climbing" techniques have been integrated into systems where fall-back search mechanisms kicks in when the robot fails to follow a good gradient.

2.1.2 Reactive Strategies Example: Transient-Based Reactive System by Ishida

In a paper from 2005 [8], odor source localization pioneer Ishida describes a transient-based system for odor plume-tracking. By use of transient analysis and fall-back-mechanisms that slows down and turns the robot if it is about to leave the plume, good performance is achieved [2], [3]. The platform used is a relatively small differential wheeled robot equipped with three semiconductor gas concentration sensors (left, centre, right) located along a line about 10-15[cm] from each other. The platform also has a device for determining the wind direction. The central gas sensor is used to detect the presence of an odor plume while the left and right ones are used to compare the concentration properties of the different sides when needed. The algorithm acts directly on the resistances R_L, R_C, R_R of the sensor elements, which decreases with an increased gas concentration, and the actual corresponding concentrations does not need to be calculated.

The behavior of the robot is, as depicted in Figure 2.1, divided into four states: In the first state, the plume is found (defined by a certain threshold), in the second one, an upwind tracking algorithm takes over. If the plume is lost during the second state, the third state, in which the robot turns back 90 degrees to try to recover the plume, is activated. If this fails, a transition is made to a fourth state and the robot performs a spiral search until it is inside the plume again. After a successful recovery of the gas plume, the second state is once again activated.

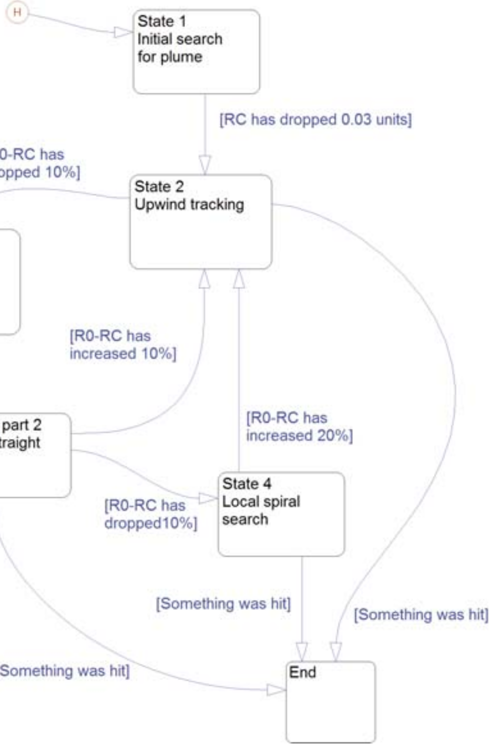


Figure 2.1: This state diagram shows the states and transitions of the gas localization algorithm described in [8]. The statements in brackets are conditions that have to be fulfilled to make the transitions in question. In these transition conditions, RC is referring to the resistance of the sensor element in the central sensor. R0 is the resistance of the sensor element when there is no detectable gas present.

Apart from the fall-back-mechanisms, the performance of the system is also high due to its transient-based nature. The third and fourth states are activated and deactivated not after R_C reaches certain absolute values corresponding to certain concentrations, but already when a given relative change in R_C is detected, *i.e.*, when the concentration starts to change. To some extent this compensates for the slow time constants of the sensors and enables the robot to move faster (10-20[cm/s]). In this way the mean search time may be decreased more than sixfold compared to earlier systems constructed by the same group [8]. At the same time, the robustness is improved [2].

The group also describes how a transient analysis approach is used in the upwind tracking (state two) to compare the output of the left and right gas sensors. The idea is to steer towards the sensor that has had the largest relative decrease in resistance since it last rose. Apart from this, the robot also steers towards the wind. This upwind tracking algorithm is described with pseudo-code in Table 2.1. The speed control algorithm too uses transients. Up to a maximum of 0.2[m/s], the speed is increased by 0.02[m/s] every time R_C-R_0 increased another 10 % and is reset to 0.1[m/s] if a drop of 2 % is detected. R_0 is the resistance of the sensor element when there is no detectable gas present. Pseudo code of the upwind steering algorithm can be seen in Table 2.2.

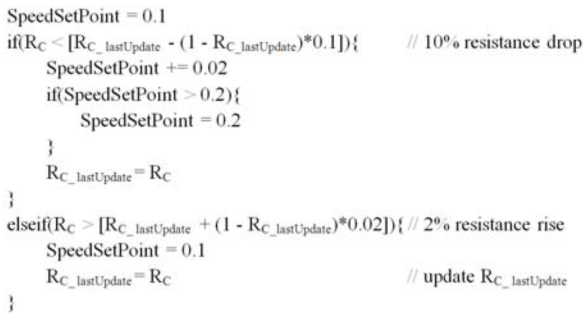
```
if($R_L$ < [$R_{L\_lastUpdate}$ - (1 - $R_{L\_lastUpdate}$)*0.1]){          // 10% resistance drop
    MotorBiasL += 0.2
    $R_{L\_lastUpdate}$ = $R_L$                                            // Update $R_{L\_lastUpdate}$
}
elseif($R_L$ > [$R_{L\_LastUpdate}$ + (1 - $R_{L\_LastUpdate}$)*0.02]){    // 2% resistance rise
      MotorBiasL = 0
      $R_{L\_lastUpdate}$ = $R_L$                                          // Update $R_{L\_lastUpdate}$
}
if($R_R$ < [$R_{R\_lastUpdate}$ - (1 - $R_{R\_lastUpdate}$)*0.1]){          // 10% resistance drop
    MotorBiasR += 0.2
    $R_{R\_lastUpdate}$ = $R_R$                                            // Update $R_{R\_lastUpdate}$
}
elseif($R_R$ > [$R_{R\_LastUpdate}$ + (1 - $R_{R\_LastUpdate}$)*0.02]){    // 2% resistance rise
      MotorBiasR = 0
      $R_{R\_lastUpdate}$ = $R_R$                                          // Update $R_{R\_lastUpdate}$
}
MotorBias = MotorBiasR - MotorBiasL
Bias = FlowBias + MotorBias                     // FlowBias ≈ DeviationFromWindDirection*Const
IncreaseLeftMotorPowerBy(Bias)
IncreaseLeftMotorPowerBy(Bias)
DecreaseRightMotorPowerBy(Bias)
```

Table 2.1: Pseudo code for the upwind steering, state 2, of Ishida's transient-based reactive system [8]. The resistances are normed so that R_0 is one.

2.1.3 Nature Inspired Algorithms

Nature proves a lot of examples on gas source localization. Just take the examples of a dog sniffing its way to a bomb or a mosquito finding its way in through your open window following the increased concentration of carbon dioxide. Many efforts have been made to mimic animal behavior with robots but as biological sensors still are superior to artificial ones [8], the task has turned out to be problematic. The two most studied bio-mimetic strategies seems to be those of a Dung beetle (*Geotrupesstercorarius*) and the silkworm moth (*Bombyxmori*) [3]. Both combine information about the current wind direction and concentration to navigate their way to the source. The Dung beetle algorithm, also called the zigzag algorithm, means that the robot zigzags its way upwind inside the plume, turning every time the concentration drops. A robot using the strategy of the silkworm moth alternates between upwind casts when an odor patch is found and consecutive searches for new patches.

However interesting, the nature inspired algorithms were rejected for standalone use. Even though components of them may be used in refined versions of other strategies, their bio-mimetic nature leaves little room for improvements.



```
SpeedSetPoint = 0.1
if($R_C$ < [$R_{C\_lastUpdate}$ - (1 - $R_{C\_lastUpdate}$)*0.1]){        // 10% resistance drop
    SpeedSetPoint += 0.02
    if(SpeedSetPoint > 0.2){
        SpeedSetPoint = 0.2
    }
    $R_{C\_lastUpdate}$ = $R_C$
}
elseif($R_C$ > [$R_{C\_lastUpdate}$ + (1 - $R_{C\_lastUpdate}$)*0.02]){ // 2% resistance rise
    SpeedSetPoint = 0.1
    $R_{C\_lastUpdate}$ = $R_C$                                // update $R_{C\_lastUpdate}$
}
```

Table 2.2: Pseudo code for the speed control in the upwind steering, state 2, of Ishida's Transient-Based Reactive System [8]. The resistances are normed so that R_0 is one.

2.1.4 Gas Source Declaration

Almost all reactive strategies for gas source localization, and all of the ones mentioned in this overview, works by in some way tracing the gas towards the source. To terminate the search when the target is reached, some kind of extra mechanism is required. Otherwise, the robot could keep running virtually forever, following the reactive rules iteration after iteration. This subtask of the gas source localization problem has been called *gas source declaration* [3]. Even though there have been more sophisticated methods proposed as well [3], the most common solution, used for example by Ishida's group [8], has been to let the search end when robot hits something, presumably the source. However, in general there is no guarantee that the robot will hit something at the location of the leak. For example, as long as the robot is only moving in two dimensions, the leak may be displaced in the third dimension, spilling from the floor or an object above. In other words, this is a field in need of more research.

2.2 Estimation/Model-Based Strategies

This section contains techniques that estimate the location of the gas source from multiple readings at different locations and a model of odor dissipation. As the performance of the algorithms is highly dependent on the accuracy of the models, good knowledge of environmental parameters and properties is essential. Compared to traditional gas source localization scenarios such as localization of earthquake victims or bombs, this should fit a process industry context well as it is possible to study the area in question beforehand. Further, data from additional stationary sensors might be available. All estimation/model-based strategies throughout the report assume that the airflow is known or measured with sensors on the robot.

2.2.1 Reasoning/Naïve Physics Techniques

These methods aim to find the odor source from a set of possible sources taking concentration samples at a few carefully chosen positions [2]. The airflow of the area is modeled using naïve physics ($\approx$ common sense) and a reasoning machine. Even though this can appear to be an attractive strategy for a robot working in a known environment, it has been rejected. The main reason for not evaluating this method is that modeling the airflow in a large three dimensional environment seems a too complex task for naïve physics. So far the technique has only been used in smaller areas with very low ceiling [2]. Also, the number of possible odor sources in a process industry environment might simply prove too many if not only joints but entire surfaces of tanks and pipes are considered to be possible sources of a leak.

2.2.2 Time Averaged Models

Until recently, most models for gas dissipation have been variations of Hinze's turbulent diffusion model [2]. This is a time averaged model of the gas concentration in the area around the source:

$$C(x,y) = \frac{q}{2\pi K} e^{-\frac{U}{2K}(r-\Delta x)} \tag{2.1}$$

where

$$r = \sqrt{(x_s - x)^2 + (y_s - y)^2}, \tag{2.2}$$

$$\Delta x = (x_s - x)\, cos\theta + (y_s - y)\, sin\theta. \tag{2.3}$$

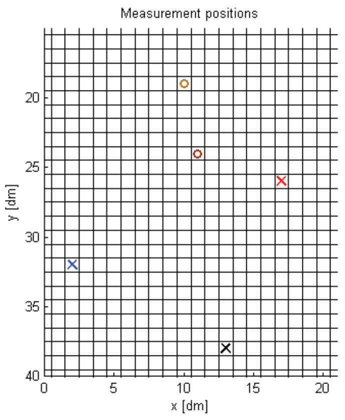
Here, *C(x,y)* is the concentration at point (x,y), q is the release rate of odor, K is the turbulent diffusion coefficient, U is the wind speed, (x_s, y_s) is the location of the odor source and θ is the angle of the upwind direction counter-clockwise from the x-axis. In addition to (x_s, y_s) that is what we are searching for, also q and to some extent K are in most cases unknown and have to be estimated.

2.2.3 Example of a Strategy Based on a Time Averaged Model

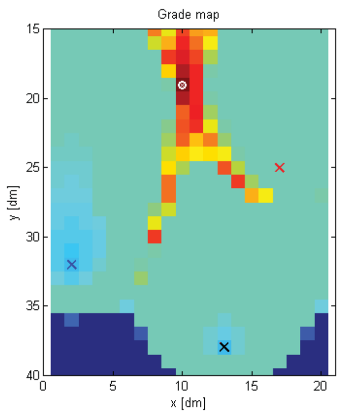
Several attempts have been made to use the model of (2.1) to estimate the location of the gas source [9, 10]. As when it came to gradient following, a paper from Ishida et al. [10], makes a good example.

The area in question is meshed in a grid structure. A robot takes a number of samples of the concentration in different grid cells, see Figure 2.2(a). For every measurement, (2.1) is used to calculate how strong the source would have been in every cell, had the cell been the location of the source, see Figure 2.2(c). For each cell, the would-be source strengths calculated for each measurement are compared. The more they correlate, the higher is the probability of the cell being the true position of the source, see Figures 2.2(d), 2.2(b).

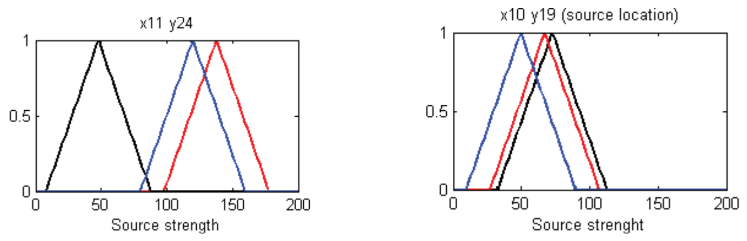
So far, the samples have been gathered in a "mow the lawn" fashion. It is possible that other strategies will make the search faster. The strategy has been tested, even outdoors with fairly good results [11]. However, a major problem is the estimation of the turbulent diffusion coefficient *K*. The model is quite sensitive, even to moderate errors in the parameter.



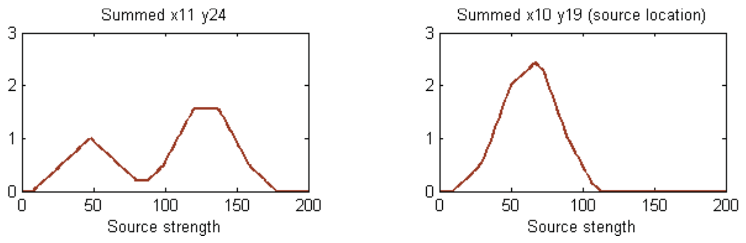
(a) Samples are taken in the positions of the three crosses. For every cell in the grid, it is calculated how strong the source would have had to be, was it situated in that particular cell, to generate the value of each sample. The two circles mark the particular cells for which these calculations are visualized in Figures 2.2(c) and 2.2(d). The upper circle happens to be in the position of the source. The airflow is directed from the upper part of the picture.



(b) A color map showing the summed correlation grades of the entire grid. The gas source is marked with a white circle. The dark blue areas in the bottom of the picture corresponding to correlation grade=0 are due to a strength-cap in the particular implementation. Sources in this downwind area would have had to be unrealistically strong to create the readings of the sensor.



(c) Triangular membership functions at two cells whereof the second happens to be the cell of the source. Each measurement is represented by a triangular membership function, centered on the concentration strength that the particular cell would have had, was it the location of the leak. Note that the functions calculated in the cell of the actual gas source show a significantly better correlation.



(d) The membership functions of Figure 2.2(c) summarized. A higher value is obviously reached at the location of the source where the membership functions correlates much better. The maximum value of each cells is translated to the "correlation grade" of the cell.

Figure 2.2: Illustration of Ishida's time averaged model-based estimation using data from a simulation of the strategy.

2.2.4 Probabilistic Models

Instead of modeling the gas distribution by average concentration, a particle approach may be used [12]. The gas leaking from the source is seen as small patches of gas that follow the airflow downwind. This model of movement is superimposed with the effects of turbulent fluctuations, for example modeled as random walk. The modeled patches may decay with a certain rate. All this means that the sensor no longer is primarily measuring the exact concentration but rather working in a binary way where for example measurements over certain thresholds are considered as "ones". Transients have also been used as triggers [13]. By combining the history of detections and non-detections, a probability-map can be created [14, 15].

With such an approach the sensors do not have to tune in to an exact value and the robot can move much faster. On the other hand, the sensors have to be very fast in the first place to be able to detect the small odor patches at all [13]. This is a practical problem as the gas sensors of today are quite slow with response times typically about 5 [s] [8].

2.2.5 Example of a Strategy Using Probabilistic Models: Infotaxis

Probabilistic methods for gas source localization has been an active field of research in recent years [13–16]. With the introduction of a novel strategy called infotaxis [15], they have been paired with a method for navigating the robot in a way that maximizes the rate with which new information is acquired. So far, infotaxis has not been used in a system with real gas detectors due to their slow response times. However, good results have been showed in simulations and with the use of a temperature sensor [16], [13]. Three dimensional implementations have also been proposed and simulated [16].

In infotaxis, the estimated probability distribution from a trace, T_t, of uncorrelated gas encounters is calculated in the following way:

$$P_t(\mathbf{r}_0) = \frac{\mathscr{L}_{\mathbf{r}_0}(T_t)}{\int \mathscr{L}_{\mathbf{x}}(T_t)\,d\mathbf{x}} = \frac{\exp\left[-\int_0^t R(\mathbf{r}(t')\,|\mathbf{r}_0)\,dt'\right]\prod_{i=1}^{H} R(\mathbf{r}(t_i)\,|\mathbf{r}_0)}{\int \exp\left[-\int_0^t R(\mathbf{r}(t')\,|\mathbf{x})\,dt'\right]\prod_{i=1}^{H} R(\mathbf{r}(t_i)\,|\mathbf{x})\,d\mathbf{x}}. \tag{2.4}$$

where $\mathscr{L}_{\mathbf{r}_0}(T_t)$ is the likelihood of observing the trace T_t of gas encounters/non-encounters for a source located at $\mathbf{r}_0$. Further, H is the number of encounters and t_i denotes the time of the i:th encounter. The function $R(\mathbf{r}|\mathbf{r}_0)$ denotes the mean rate of encounters at position $\mathbf{r}$ for a source located at $\mathbf{r}_0$. With the model outlined in Section 2.2.4, the resolution of $R(\mathbf{r}|\mathbf{r}_0)$ in two dimensions reads

$$R(\mathbf{r}|\mathbf{r}_0) = \frac{R}{ln\left(\frac{\lambda}{a}\right)} e^{\frac{(y_0-y)}{2D}} K_0\left(\frac{|\mathbf{r}-\mathbf{r}_0|}{\lambda}\right), \tag{2.5}$$

$$\lambda = \sqrt{\frac{D\tau}{1+\frac{V^2\tau}{4D}}} \tag{2.6}$$

where R is the release rate of detectable gas patches from the source and K_0 is the modified Bessel function of the second kind. The patches have a lifetime of τ, propagate with a diffusivity D and are advected by a mean wind V blowing in the negative y-direction. The parameter a denotes the size of the circular object detecting the point-sized gas patches (alternatively, the size of the patches detected by a point-sensor) at rate $R(\mathbf{r}|\mathbf{r}_0)$.

What separates infotaxis from other strategies using probabilistic models is the way in which the robot is controlled to optimize the acquisition of valuable information used as input to the probability estimator described above. The problem of balancing between exploiting currently known information and exploring for more information is a quite general one [15]. In the case of gas source location estimations, it translates to a trade-off between moving the robot towards the point that is estimated to have the highest probability of containing the source at the moment, and on the other hand, to first let more information be collected (like the "mow the lawn" sample pattern mentioned in Section 2.2.3 that only collected information according to a pre-defined scheme). For a range of theoretical cases, it has been shown that neither pure greedy exploitation nor pure exploration is effective [15]. The optimal algorithms are blends of both sides. The basic idea of infotaxis is to achieve such a blend by moving in the direction that minimizes the entropy of the search area. As the entropy decreases faster close to the source where gas patches, *i.e.*, information, arrives at a higher rate, the robot will be guided to the source. The robot will be guided in a direction that maximizes the information acquisition, thereof the name "infotaxis". The expected change of entropy from moving from one point to another is calculated as

$$\overline{\Delta S}(\mathbf{r} \to \mathbf{r}_j) = P_t(\mathbf{r}_j)[-S] + [1 - P_t(\mathbf{r}_j)][\rho_0(\mathbf{r}_j)\Delta S_0 + \rho_1(\mathbf{r}_j)\Delta S_1 + \ldots]. \qquad (2.7)$$

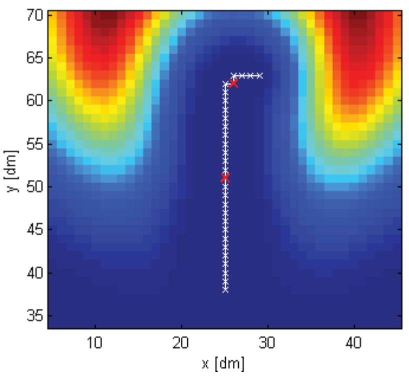
The first right-hand term corresponds to the case where the source is found whereas the second one represents the cases of 0,1,2... gas patch encounters not being at the location of the source. The symbols ΔS_k denotes the change of entropy between the fields $P_{t+1}(\mathbf{r}_0)$ and $P_t(\mathbf{r}_0)$ in case k encounters are made. The probability of encountering k patches during one time-step Δt is denoted by ρ_k which is the Poisson distribution

$$\rho_k = \frac{h^k e^{-h}}{k!} \qquad (2.8)$$

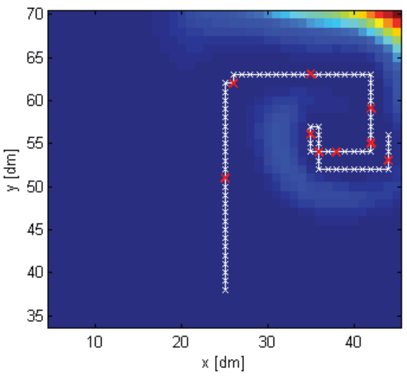
with the expected number of occurrences $h(\mathbf{r}_j)$ estimated as

$$h(\mathbf{r}_j) = \Delta t \int P_t(\mathbf{r}_0) R(\mathbf{r}_j|\mathbf{r}_0) d\mathbf{r}_0. \qquad (2.9)$$

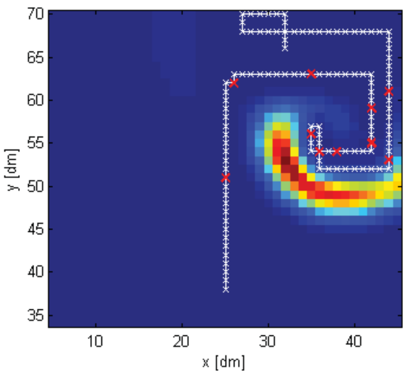
A simulated run using an infotactic algorithm can be seen in Figure 2.3.



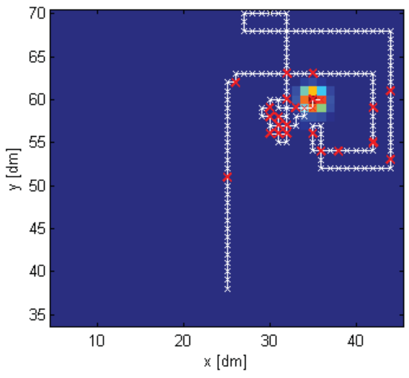
(a) After 30 iterations.



(b) After 80 iterations.



(c) After 120 iterations.



(d) At the source after 170 iterations.

Figure 2.3: Visualization of a simulated run of infotaxis. The colors shows the estimated probability of the source being in the cell in question, the color scales are individually adjusted to each figure. The red crosses mark the points of detection.

Chapter 3

System Properties

To implement the gas source localization strategies in a real environment, a system consisting of several hardware and software parts had to be constructed. This process was partly carried out in parallel with the simulation of the different algorithms and the properties of the system were incorporated in the simulations.

3.1 The Laboratory

The practical experiments were all conducted in the robot lab of ABB AS at Ole Deviks Vei 10, Oslo, Norway. The laboratory is used for research and development of robotics within Oil and Gas in the project mentioned in Section 1.2. The laboratory comprises two ABB IRB 4400 robots mounted on a common track, one ABB IRB 2400-16 mounted upside down on a three-axis Güdel Gantry and a full-scale separator process module. All interact with ABB's 800xA automation system. The robots are controlled by an in-house system managing pathplanning, workstation-robot communications and user interaction. The three robots and the process module are depicted in Figure 3.1.

3.2 Gas Release

For obvious reasons, it is not possible to release for example highly toxic H_2S, a likely pollutant in an oil and gas environment, for the purpose of the test. However, all gases obey the same laws of nature and spread in similar ways. To test algorithms for finding the source of an emission, almost any gas may be used. In strong wind and turbulence, effects such as those caused by differences in density are negligible. Even though carbon dioxide was considered as well, the choice fell on ethanol. The prime reasons for this were the availability of decent and inexpensive sensors, the ease of handling it, the ease of making it emit at a good rate (not depleting the source too fast) and ease of being compliant with ABB regulations concerning handling of chemicals. Also, ethanol has been the by far most common leak gas in earlier studies and with the same

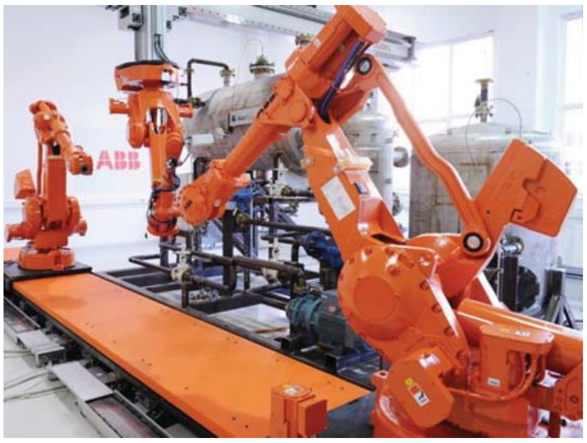


Figure 3.1: All three robots are controlled through an in house system managing path-planning, workstation-robot communications and user interaction.

gas and sensors we are able to compare our own result to those of these studies more fairly. The source of the ethanol vapor was vodka containing 60 % by volume ethanol. A device was built that let compressed air flow through the mixture of ethanol and water and eject through a small hose. In this way a release more similar to a real leak was created. As the ethanol should have evaporated faster than the water the share of ethanol likely decreased as soon as the device was activated. However, the alcohol content in the air was always large enough for the algorithms to work and the reservoir was repeatedly refilled so that the comparisons between the different algorithms would be fair. The pressurized air could be switched on and off through the 800xA system. The release device is depicted in Figure 3.2.

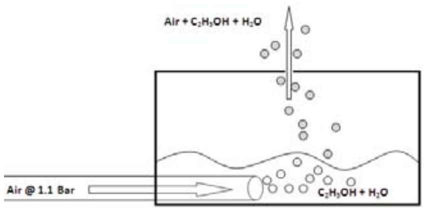


Figure 3.2: Sketch of the gas release device.

Wind has a huge influence on the spread of a released gas. To limit the scope of the thesis, only the situation where a wind blows in the environment was considered. This is a quite likely assumption in an oil and gas environment of Norwegian standards where the plants are located offshore or next to the sea. Also, this eliminates the need to consider inflection on the local wind field caused by the movements of the robot (even though its sheer presence will disturb the wind field). To create an artificially uniform wind field, a fan was placed about one [m] behind the gas source. To gear the setup towards an oil and gas context, the leak was located on the border of the process module while the fan was placed completely inside it. The wind created by the fan was thus obstructed by pipes and a crane and took unpredictable paths, just as can be expected in a real scenario. Turbulence in the wind field was also created by the fan itself and its limited capacity made the field less uniform than a real wind. As good anemometers are quite expensive, none were installed during this study. The general direction and speed of the wind was assumed to be known. It may be a good idea to incorporate an anemometer in a final device but information from plant anemometers or even weather forecasts may also prove sufficient. The setup is displayed in Figure 3.3.

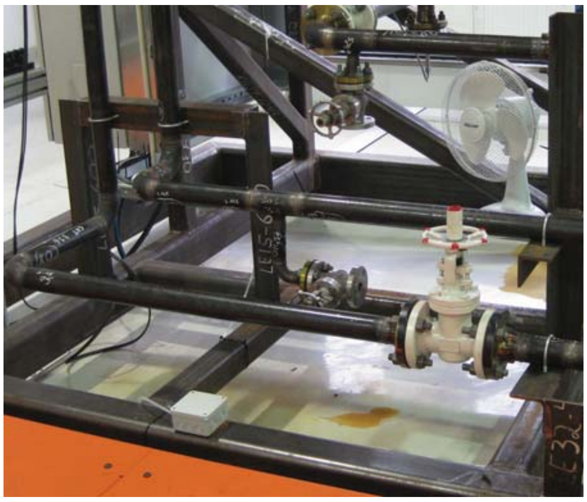


Figure 3.3: The fan was placed inside the process module to let the pipes create realistic turbulence. The white box in the bottom left corner is the release device from where ethanol fumes are ejected.

When the results of the real runs of the different algorithms (see Tables 5.1-5.5)were finally compared, it was obvious that the results of all algorithms tended to be more negative (left) than straight downwind of the source. An investigation in the matter showed that the gas concentration actually is higher on the negative side of this “wind line” than on the wind line itself on the border of the process module. The pipes and valves have likely steered the wind and gas in this direction. This shows some of the complexity of the gas source localization problem when the flow is disturbed. In some sense, it also indicates that the performance of the tested algorithms might have been better than the numbers show, as their results have been closer to the gas stream than to the source.

3.3 Sensor Circuit

The choice of gas sensors was made along with the choice of gas. The availability of the Figaro TGS 2620 semiconductor gas sensor (see Appendix A), which was eventually chosen, made an impact on the choice of ethanol as the released gas. The sensor is durable, inexpensive, sensitive to concentrations as low as 50 ppm and has a fair response time. Practical investigation as well as literature [8] indicates a response time of about 5[s]. On the downside, the recovery time after a peak in gas concentration is very long, the sources indicate about one minute [8]. Further, the sensor has to be warmed up for about an hour to be accurate [17].

To enable simultaneous sampling at different locations with the prospect of stereo capabilities, two sensors were used. A circuit was produced to supply them with the correct voltage and to take care of their respective output. A simplified diagram of the circuit serving one sensor is displayed in Figure 3.4 whereas a complete circuit diagram can be seen in Figure B.1. Figure 3.5(a) shows a picture of the real circuit. To supply the sensors with the required 5[V] DC, a L7805 Voltage regulator is used. A capacitor stabilizes the input voltage to the voltage regulator. For each of the two sensors, identical parts are used to take care of their outputs. The heat element R_H and the sensor element R_S are both parts of the sensor itself. R_H heats the sensor element to its service temperature and R_S responds to the ethanol concentration by adapting its resistance, thereby performing the actual sensing. The value of R_S control the voltage over the resistor R_L. This voltage may be sampled as the output signal of the sensor, but as a voltage modulated signal is sensitive to disturbance, it is converted to a 4-20[mA] current modulated signal using a XTR110KP U/I converter. The XTR110KP is configured to an input of 1-5[V]. The IRF9610 MOSFET transistor is also part of the U/I conversion circuit. Apart from these more or less necessary parts, the circuit also comprises a light-emitting diode and a corresponding resistor to indicate whether the circuit is powered or not.

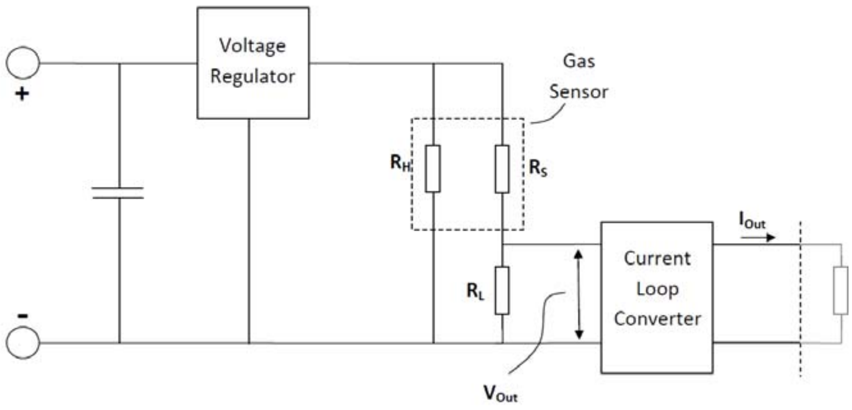
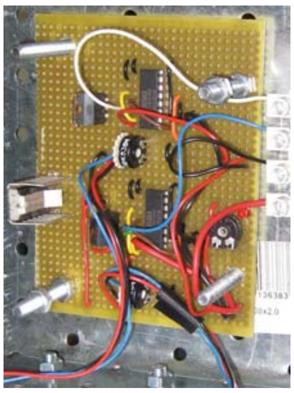


Figure 3.4: A simplified diagram of the circuit serving one of the sensors. For a complete circuit diagram, see Figure B.1.

In the practical implementation, all components are soldered onto the same board except the two sensors that each have their own small board, wired to the main board. This gives the possibility to distance the two sensors from each other to take simultaneous samples at different locations. In practice, the two sensors were separated by about 15[cm]. The final mounting of the sensor circuit is displayed in Figure 3.5.


(a) The sensor circuit board.

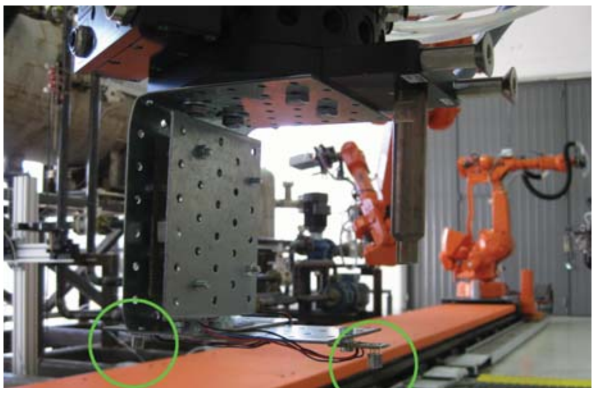

(b) The sensor circuit and the sensors (highlighted by the green circles) as mounted on the robot tool.

Figure 3.5: The final circuit board and its enclosure.

3.3.1 Calibration

To convert the resistances of the sensor elements to gas concentrations, calibration is needed. Besides, a number of nonlinearities are introduced by the sensor element itself as well as the parts of the circuit converting resistances to voltages and voltages to current. From the sensitivity characteristics graph of the sensor datasheet in Appendix A, it can be found that the relation between the concentration C and the resistance of the sensor element R_S in the sensors working area is

$$R_S = l \cdot C^k \tag{3.1}$$

where l is a constant determining the absolute concentration level that has to be calibrated with a known concentration for the specific sensor and k shows the relative relationship between the gas concentration and the resistance of the sensor element. From the sensor datasheet in Appendix A K was found to be -0.63 for concentrations between 100 and 5000 ppm.

The correlation between the resistance of R_S and the voltage U_L over R_L is governed by the following equation

$$U_L = U_d \frac{R_L}{R_L + R_S} \tag{3.2}$$

where U_d is the reference voltage of 5[V].

The ABB AI810 unit that receives the 4-20[mA] current modulated signals A/D converts them to digital values between 0 and 100. Because of this and the fact that the input intervals of the U/I converters are set to 1-5[V], the correlation between the voltage U_L and the output signal S is

$$U_L = \frac{S}{25} + 1 \tag{3.3}$$

Together, the above equations render the following relationship between output signal and ethanol concentration

$$C = \left[\left(\frac{25U_d}{S+25} - 1 \right) \frac{R_L}{l} \right]^{\frac{1}{k}} \tag{3.4}$$

which with known constants inserted becomes

$$C = \left[\left(\frac{100-S}{S+25} \right) \frac{21000}{l} \right]^{-\frac{1}{0.63}} \tag{3.5}$$

With

$$\Psi = \left(\frac{l}{21000} \right)^{\frac{1}{0.63}} \tag{3.6}$$

Equation (3.5) becomes

$$C = \Psi \cdot \left(\frac{S+25}{100-S} \right)^{\frac{1}{0.63}} \tag{3.7}$$

As no known concentration was available, l was set to 50 after a rough estimation. The absolute concentration plays only a very small role in the reviewed algorithms, which almost entirely depends on relative measurements. The value of k is not valid for ethanol concentrations above 5000 ppm. With an l of 50 this corresponds to an output level of 92, far above what is measured during the runs. However, the k-value is not accurate for concentrations below 100 ppm either. With l=50, 100 ppm corresponds to an output value of 45 which is far more than the lowest practical measurements. Thus the measurements at very low concentrations cannot be fully trusted with respect to exact values. However, the manufacturer claims the sensors to be working from zero concentration and they seem to give predictable response even at extremely low concentrations, even though the resistance of R_S cannot be mapped to an exact concentration level.

The sensors were not fully identical and further calibration was needed to balance them. This was not done within the initial calibration mentioned above but afterwards. The two sensors were placed very close to each other and exposed to a series of concentrations of ethanol vapor. After the collection of data, linear regression was performed on the difference between the two sensors. The polynom obtained was then applied to the measurements from one of the sensors so that both sensors showed the same result when at the same location.

3.4 Robot

The choice of robot platform was an early and important decision. Initially, a small autonomous wheel-driven robot, e-puck[1], was considered, but instead a full-size track-mounted industrial robot, an ABB IRB 4400 present in laboratory, was chosen. The advantages with this platform include exact positioning, a high maximum payload and thus a negligible need for miniaturization, easier connection to workstation/network and a three dimensional workspace. Thanks to the track, the work area was sufficiently big for the work in this thesis. Similar robot systems are intended for use within the future oil and gas industry, performing a wide range of tasks. Those robots may be fitted with gas sensors and it thus makes sense to use such a robot system for this study. It puts the study in the right setting and it enables an exploration of the advantages and limitations of using industrial robots for gas source localization. The sensor circuit is placed on a tool changer so that the robot can pick it up and release it when needed.

3.5 System

In the Robot Lab, an in house system is used for managing pathplanning, workstation-robot communications and user interaction. The gas source localization applications controlled the robot by providing coordinates and orientations of the tool. A pathplanner described in Section 3.5.1 then calculated the path for the robot to take and this path was communicated to the robot controller through an auto generated RAPID-program. To make implementation, simulation and visualization easy, the algorithms were all implemented in MATLAB. Since the in house robot control system is implemented in C#, data had to be transferred from MATLAB to C#. This was realized by using the class MLApp which lets MATLAB commands and functions be called from C#. In order not to have to handle inner values of the algorithm in the C# environment, the values used by MATLAB were all declared global and saved in the MATLAB workspace. This made the transition from simulations to real experiments smoother.

3.5.1 Pathplanner

During the work on this thesis, the control of the three robots in the laboratory underwent an upgrade. Among other things, an all new pathplanner was created. It enables on the fly calculation of collision free paths between any two reachable positions and orientations in the laboratory. As the system was entirely new at the time of the practical experiments and because of its general purpose nature, some of its properties did not fit the gas source localization algorithms very well. For example, as the pathplanner solves the entire pathplanning task, no control is left to the calling program over what trajectory the robot, and more specifically, the tool should take to the next position. Sometimes the robot will not move linearly between two points but pass waypoints some 50-100[cm] away from it. Further, the system is designed for moving between points distant from each other where the motion itself takes a lot of time. It is however

[1]http://www.e-puck.org/

not optimized for short and simple movements and it takes 4-10[s] to perform the calculation and actuation of even the smallest move. To speed up pathplanning, old paths and waypoints are saved in a database. To enable the reuse of old paths, a robot is made to move to a stored point if there exists one within 5[cm] of an intended target and as long as no robot joint calculated for the new target deviates more than 0.17[rad] from the angles of the robot joints connected to the stored point. If not taken into account, this feature will obviously hamper exact positioning of the tool as the robot might move to stored a point a few [cm] away from the intended one.

One adaptation of the pathplanner was made to make it fit the gas source localization algorithms better. To make the robot disturb the gas flow as little as possible, its base was always placed in a position downwind of the tool's orthogonal projection onto the track.

3.5.2 Signal routing

To get the readings from the sensors into the relevant program at the right workstation turned out to require quite a few steps. As mentioned in Section 3.3, the sensors respond to the present ethanol concentration by adapting the resistance over the sensor elements. In the circuit depicted in Figures 3.4 and B.1, this corresponds to a voltage above R_S. The signals are routed tens of meters from the robot tip before they are sampled. Since a voltage modulated signal is sensitive to disturbance, the signals are converted to 4-20[mA] current modulated signals using XTR110KP U/I converters. Close to the robot controller, these signals are received by an ABB AI810 analog input module connected to an ABB 800xA system over a PROFIBUS network. It should be pointed out that this control system, present in the lab to control a process model, is not used for controlling the robots in the lab. The 800xA system includes an OPC server connected to a TCP/IP Ethernet network. With Cogent OPC DataHub, the sensor data is tunneled from the computer where the OPC server resides to the workstation where the gas source localization algorithms are running. There it is written into a database from where a MATLAB program reads the values into the actual control software. A more effective solution would have been to read the data directly from the OPC server that is created in the receiving computer. However, the database approach turned out to be easier to implement and is good enough at the slow timescale concerned.

The connection socket of the OPC server that is used to read the sensor values is designed for system visualization, not system control. The control software is supposed to reside within the 800xA system itself and not in an external computer. Because of this, the read signal has a very low priority and is only updated once every second. Since the sensors are slow, this does not have a very big impact but it is still possible that an update rate faster than the 1[Hz] could have been useful.

Chapter 4

Algorithm Description and Setup

After the literature study, three particularly interesting algorithms with three very different approaches were selected to be tested in practice. The three algorithms were Ishida's Transient-Based System ([8] and Section 2.1), Ishida's Time Averaged Model based Gas-Source Estimation ([10] and Section 2.2.3) and infotaxis ([15] and Section 2.2.5). All three algorithms were first run in a simulated environment so that the necessary systems around them could be implemented, debugged and have their basic functionality verified. Some algorithms obtained from the literature study are not a complete system that covers everything from detection of a leak to moving strategy and final definition of the source. The simulations gave an opportunity to add and test extra functionality to the algorithms so that they all cover the entire process of gas source localization. Already at this stage, some small changes were made in the algorithms to adapt them to the properties of the real hardware. In some cases further algorithms were derived from the original ones and tested.

After the simulation of a particular algorithm was finished, real world experiments were run in the robot lab. Adjustments were made to the algorithms with regard to limitations of the real system. Parameters were tuned and the algorithms were finally run with their behaviors and results logged for comparisons.

4.1 Transient-Based Reactive System by Ishida

The transient-based system of Ishida [8] was the first to be implemented in simulation as well as in the real environment. This was the only purely reactive algorithm to be tested in the laboratory. However, as the exact scope of the algorithm comparison was not decided at the point of the initial simulations, some variations of Ishida's original algorithm, including a novel method for gas source localization, were developed and simulated in an attempt to investigate if the performance and reliability of Ishida's algorithm could be improved. These variations are not part of the main comparison of this study but the findings made during their development may nevertheless be of interest and are described later in this section.

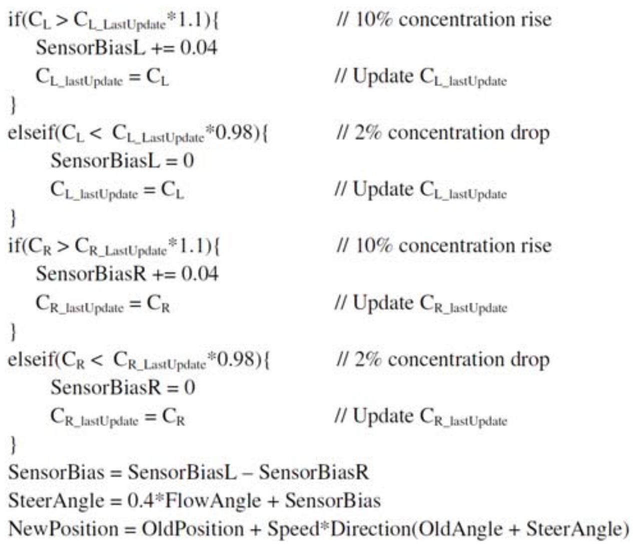
4.1.1 Simulation

Simulation Environment

The simulated test area was made up of a grid with 10[cm] spacing. It had a steady uniform airflow and was free of obstacles. For simulation of the algorithm of [8] and its variations, the time averaged model for turbulent diffusion, (2.1) was used. To get a plume similar in shape and concentration to the one measured by Ishida's group in their study [8], the following parameters were used: $q = 2.5 \cdot 10^{-6}$, $K = 0.02$ and $U = 0.3$. Moreover, the dimensions of the test area were set to be identical to those of the room in that study. Since (2.1) only describes the mean concentration, normal distributed noise with standard the deviation being 40 % of the calculated concentration was applied to the concentration of every tile and updated after every measurement to simulate temporary variations. Whenever the noise rendered a negative concentration, another noise was applied instead to avoid the unnatural situation if negative concentration. This obviously increased the expectation-value, but only by 0.7 % and that change was thus neglected. Atop of this, a normal distributed noise independent of concentration level with a standard deviation of 2 ppm was applied to simulate the uncertainty of the sensors. The sensor response to changes in concentration was modeled as exponential decay towards the new value. After observations of the real sensor, the half-life of the decay was set to 8[s] for concentration decreases and to 0.55[s] for increases.

Simulated Algorithm Truthful to the Original Paper

The plume-tracking algorithm of [8] was run in the simulation environment described above. The simulated algorithm was mostly truthful to the original system but some changes were however made to adapt it to the hardware limitations that were known at this point. In opposite to Ishida's real-world experiment, the movement of the simulated robot was made stepwise (1[Hz]) and the sensors were not simulated as measuring continuously but updated after every movement. The calculations of the steering algorithm were too given an update rate of 1[Hz]. Ishida's article does not state the cycle time of their system, *i.e.*, how often steering commands were updated, but deeming from steering plots in [8] the system was considerably faster than 1[Hz]. To make the simulated sensor array more equal to the two-sensor array of the hardware, the mean of the right and left sensors represented the reading from the third, central sensor, used in the original study. Further, Ishida's group used the raw resistances of the sensor elements as sensor signals, and their algorithm was working directly based on those. In the simulations as well as in the real tests, our algorithms were however working with concentrations to make them more sensor independent. In practice this is almost the same thing but some nonlinearity in the sensors and their circuitry, see Section 3.3.1, is eliminated. The properties of the motors and the wind sensor of the robot used in the original study were not fully known and some steering parameters of state 2 had to be re-tuned to fit the actuating principles of the simulation environment. Pseudo code of the upwind steering algorithm, *i.e.*, state 2, can be seen in Table 4.1. The same gas source declaration technique was used as in the original article, *i.e.*, the robot stops when it hits something (see Section 2.1.4). The criteria for success was thus to hit a box of similar size (20[cm] $\times$ 30[cm]) of the one containing the source in the original experiments without first hitting the walls or the fan.



```
if($C_L > C_{L\_LastUpdate}$*1.1){           // 10% concentration rise
    SensorBiasL += 0.04
    $C_{L\_lastUpdate} = C_L$                  // Update $C_{L\_lastUpdate}$
}
elseif($C_L < C_{L\_LastUpdate}$*0.98){      // 2% concentration drop
      SensorBiasL = 0
      $C_{L\_lastUpdate} = C_L$                // Update $C_{L\_lastUpdate}$
}
if($C_R > C_{R\_LastUpdate}$*1.1){           // 10% concentration rise
    SensorBiasR += 0.04
    $C_{R\_lastUpdate} = C_R$                  // Update $C_{R\_lastUpdate}$
}
elseif($C_R < C_{R\_LastUpdate}$*0.98){      // 2% concentration drop
      SensorBiasR = 0
      $C_{R\_lastUpdate} = C_R$                // Update $C_{R\_lastUpdate}$
}
SensorBias = SensorBiasL – SensorBiasR
SteerAngle = 0.4*FlowAngle + SensorBias
NewPosition = OldPosition + Speed*Direction(OldAngle + SteerAngle)
```

Table 4.1: Pseudo code for the upwind steering, state 2, of the simulation of Ishida's transient-based reactive system [8] most faithful to the original article. Compared to the pseudo code of the same state in the original article, Table 2.1, changes have been made in the code to make it act on concentration instead of direct sensor resistance and to adapt it to new ways of gaining wind information and actuating the steering

Using the algorithm described above, results similar to those of Ishida's group were obtained: Out of 13 runs 12 were successful with a mean completion time of 27.25[s]. In Ishida's case all 13 trials were successful with the mean time 32[s]. The trajectory of one successful run is pictured in Figure 4.1.

Simulation of Gradient Following Variation

A variation of Ishida's system towards a more classical gradient following strategy was also tested. The system was identical to the one depicted above but with another algorithm for upwind steering (state two): Instead of comparing transients, the robot simply steered towards the side with the highest absolute concentration measurements, just like in a standard gradient following algorithm [7]. The wind direction still affected the steering and the fall-back mechanisms, the transitions between states 2, 3 and 4, were still governed by transients. The pseudo code of this version of state 2 can

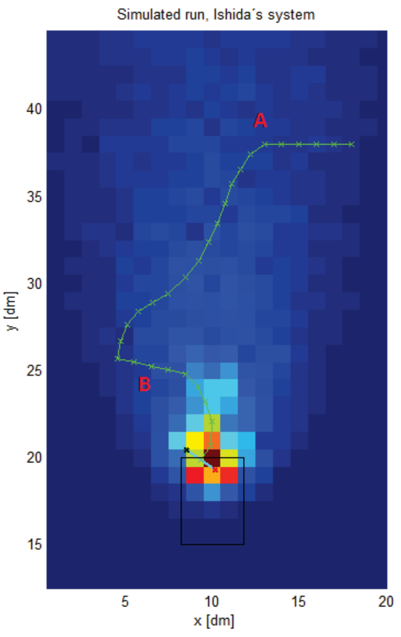


Figure 4.1: A plot of the robot movements during a simulated run of Ishida's Transient-Based Reactive System [8]. The color map shows the gas plume and the black box the target around the gas source. The path of the robot is highlighted in green. Note the transition from state one to two after five steps at A and how and state three is active around steps 20 to 25 at B.

be seen in Table 4.2. With this setup, 12 out of 13 trials were successful with a 26.5[s] mean completion time.

Simulation of Gradient Following Variation with Extrapolation

Another way of using transients to compensate for the slowness of the sensors was evaluated. The system was identical to the gradient following algorithm described above but with another variation in the upwind steering state. Instead of using the absolute values of the sensors to estimate the gradient, these values were linearly extrapolated using the last two readings for each sensor. The sensor values which were predicted two steps (or [s]) into the future were then used for gradient calculation. As the sensors react far slower when the concentration is sinking compared to when it is rising, the extrapolation was only applied when the measured concentration was falling. Pseudo code for this version of state 2 can be seen in Table 4.3. Out of 13 runs with this algorithm, only 10 were successful with a mean completion time of 28.5[s].

```
SensorBias = ($C_L$ - $C_R$)/mean($C_L$,$C_R$)
SteerAngle = 0.12*FlowAngle + 0.24*SensorBias
NewPosition = OldPosition + Speed*(OldAngle+SteerAngle)
```

Table 4.2: Pseudo code for the upwind steering, state 2, of the simulation of the gradient following variation of Ishida's transient-based reactive system [8]. The robot simply steers towards the wind and towards the side with the highest gas concentration.

```
DeltaL = $C_L$ - $C_{L\_LastCycle}$              //Trend calculation
if(DeltaL < 0) {
    $C_{L\_Eff}$ = $C_L$ + DeltaL*2              //Extrapolation 2 cycles
} else {
    $C_{L\_Eff}$ = $C_L$                         //No extrapolation
}
DeltaR = $C_R$ - $C_{R\_LastCycle}$              //Trend calculation
if(DeltaR < 0) {
    $C_{R\_Eff}$ = $C_R$ + DeltaR*2              //Extrapolation 2 cycles
} else {
    $C_{R\_Eff}$ = $C_R$                         //No extrapolation
}
$C_{L\_LastCycle}$ = $C_L$                       //Updated $C_{L\_LastCycle}$
$C_{R\_LastCycle}$ = $C_R$                       //Updated $C_{R\_LastCycle}$

SensorBias = ($C_{L\_Eff}$ - $C_{R\_Eff}$)/mean($C_{L\_Eff}$,$C_{R\_Eff}$)
SteerAngle = 0.12*FlowAngle + 0.24*SensorBias
NewPosition = OldPosition + Speed*Direction(OldAngle + SteerAngle)
```

Table 4.3: Pseudo code for the upwind steering, state 2, of the simulation of the gradient following variation with extrapolation of Ishida's transient-based reactive system [8]. The robot steers towards the wind and the side predicted to have the highest gas concentration if the current process continues for two steps.

Simulation of Novel Gas Source Declaration Technique

Several new solutions to the gas source declaration problem were considered for use with Ishida's transient-based reactive system to make it robust to the problems associated with the original declaration technique when the source is displaced in a third dimension, see Section 2.1.4. Among the considered solutions was the creation of a fifth state that performs a local search when the concentration has reached a threshold. However, the problem with this method and many others are that they presume knowledge about the intensity of the source, which is a serious drawback. Besides, as local peaks in gas concentration may have been created [6], a fifth state risks confining the search to the wrong place. A partly new approach to the problem was developed and tested in simulation. The idea is to keep the normal behavior of the robot and analyze it. If the robot passes a leak that is not represented by an obstacle in the plane of the robot, the gas concentration will suddenly drop, as it always will when the robot leaves the plume. The robot then enters state three and as there is no plume to recover behind the source, state four is soon entered as well. The robot starts a spiral search and finally recovers the plume close to the source. Of course, the robot soon loses the plume again and the process repeats. It is then possible to estimate the location of the source by taking a simple average of the positions where the plume is lost, or even better, a few steps before. During the earlier parts of the search, the plume may be lost at other positions as well. However it is very rare that it is lost more than once at the same position and by including only clustered positions in the estimation the method is made more robust. The more times the robot gets to recover and lose the plume, the more accurate and reliable the estimation becomes. When a cluster contains enough positions, the search is complete. The technique showed good promise during simulations but no data was recorded at this stage. One exemplifying run can be seen in Figure 4.2. As the real gas source was later placed inside the process module, this expansion of the original gas source declaration technique was not useful in the real experiments as the robot cannot even reach all the way to the source in the first place. However, the technique may still be useful in other applications.

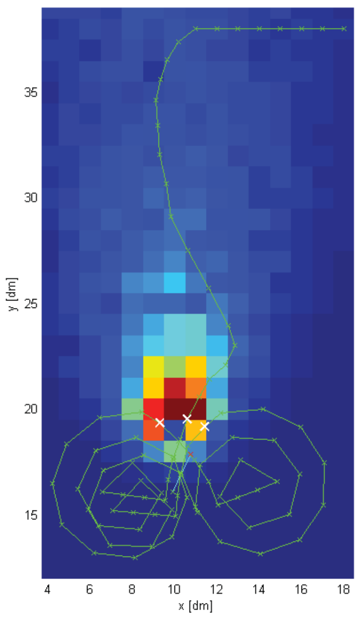


Figure 4.2: A simulation of Ishida's transient based reactive system where information gained from transitions from state two to three of the algorithm are used for gas source declaration (establishing the location of the source exact enough and deciding when the run is finished). The white crosses marks the positions of the robot two iterations before the sensor values started to decrease before the transitions from state two to three occurred. The marks are made two iterations prior to the state change to compensate for the slowness of the sensors.

4.1.2 Real World Tests

After the simulations were finished, the algorithm was tried out on the real robot. The only reactive algorithm to be implemented on a real robot was the one described in Section 4.1.1, which aimed at being as similar to the original algorithm as possible. The real environment was however not identical to the simulated one and the difference forced some modifications of the algorithm.

Difference Between Conditions of the Real and Simulated Runs

Even though the performance of the hardware had been studied in parallel with the simulations, the simulation environment never got fully truthful to the real world. For example, the sensor response to a new concentration was as mentioned in Section 4.1.1 modeled as an exponential decay with half-lives based on tests of the real sensors. However, the strong noise introduced in the simulations to really stress the algorithm afterwards turned out to not only have made the sensors less accurate, but also faster. The real sensors were thus slower but less noisy than the simulated ones. Another concern with the sensors was the trouble of getting them to work together. The behavior of the two sensors drifted unevenly over time and they had to be recalibrated as described in the end of Section 3.3.1 on a regular basis to ensure that both sensors returned the same result while measuring identical concentrations.

Limitations inflicted by the robot control system as described in Section 3.5.1 was another concern. As every move of the robot took 4-10[s], it was not possible to run the system at 1[Hz] as in the simulations. Further, the pathplanners way of sometimes not calculating linear trajectories introduced large errors in the measurements. As the sensors were continuously active, the concentration along the whole trajectory influenced them. The signals were odd in the cases when the tool had not taken the shortest path from the last point as (just about) presumed in the original algorithm. In Section 3.5.1 it was also described how the robot might end up in a point up to 5[cm] away from where it was originally commanded due to rounding to stored positions. This was not accounted for in the implementation of Ishida's reactive system.

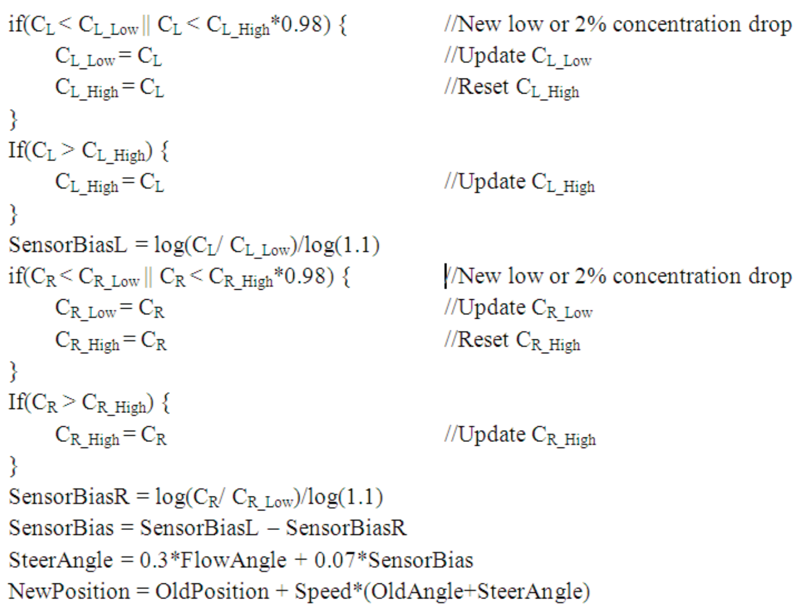
One deliberate change of the environment was the introduction of extra turbulence originating from the placement of the leak and fan within the process module. The pressurized leak surely too introduced a kind of turbulence that was not included in the simulation and the robot may have introduced some disturbance itself. However, the area in which a wind was created was not very wide. To mimic real conditions, it would have been better with a more uniform wind hitting the process module. Finally, the parameters for rate of gas release, diffusion and wind strength of the simulations were obviously not exact estimates of the real parameters.

4.1.3 Final Algorithm

To adapt the algorithm to the real conditions, some changes had to be made. First of all, most parameters had to be adjusted to fit new environmental parameters and timings. The parameters of the algorithm used for the real robot can be seen in Figure 4.4 and Table 4.4. To adapt to the long cycle time of the system, where every move takes more than 4[s], the influence of the gas sensors on the steering was calculated in a new manner. If it would still have been made in the same way as in the simulations, see Table 4.1, where the biases of the different sensors were increased step by step after that the concentration passed certain levels, much information would had been lost. For example, over a cycle 5[s] long, the response of a sensor may have passed multiple levels, thus deserving a strong increase of the corresponding bias. However, in the previous implementations the bias could only rise by a fix value, and would have done so already when the sensor response passed the first level, thus no difference between a modest and a large sensor response would had been made. To avoid such effects, the calculation of gas sensor bias is done with the continuous equation (4.1)

$$SensorBias = log\left(\frac{C}{C_{Low}}\right)/log\left(Inc\right) \tag{4.1}$$

where C is the measured concentration, C_{Low} is the lowest concentration measured since the last drop and Inc is the increase rate. In the original study and in the simulations, the stepwise increase was 10 %, corresponding to an Inc of 1.1 if effects of discretisations are omitted. Pseudo code for state 2 of the real experiment, showing the implementation of this, is displayed in Table 4.4.



```
if($C_L < C_{L\_Low}$ || $C_L < C_{L\_High}$*0.98) {    //New low or 2% concentration drop
    $C_{L\_Low} = C_L$                                   //Update $C_{L\_Low}$
    $C_{L\_High} = C_L$                                  //Reset $C_{L\_High}$
}
If($C_L > C_{L\_High}$) {
    $C_{L\_High} = C_L$                                  //Update $C_{L\_High}$
}
SensorBiasL = log($C_L$/ $C_{L\_Low}$)/log(1.1)
if($C_R < C_{R\_Low}$ || $C_R < C_{R\_High}$*0.98) {    //New low or 2% concentration drop
    $C_{R\_Low} = C_R$                                   //Update $C_{R\_Low}$
    $C_{R\_High} = C_R$                                  //Reset $C_{R\_High}$
}
If($C_R > C_{R\_High}$) {
    $C_{R\_High} = C_R$                                  //Update $C_{R\_High}$
}
SensorBiasR = log($C_R$/ $C_{R\_Low}$)/log(1.1)
SensorBias = SensorBiasL – SensorBiasR
SteerAngle = 0.3*FlowAngle + 0.07*SensorBias
NewPosition = OldPosition + Speed*(OldAngle+SteerAngle)
```

Table 4.4: Pseudo code for the upwind steering, state 2, of the 13 first real runs of Ishida's transient-based reactive system presented in Section 5.1. The algorithm of the following 12 runs presented in Sections 5.1, 5.2.2 and 5.4 was identical but SensorBias was multiplied by the parameter 0.11 instead of 0.07. To adapt the algorithm to the long cycle time of the system, the influence of the of the gas sensors is calculated with a continuous algorithm.

One further change of the algorithm was imposed by the sample rate. As explained in Section 3.5.2, the sensor responses were sampled at 1Hz. This sample rate was fixed and thus not synchronized with the 4-10[s] cycles of the system. This meant that the algorithm would either have to use values up to one second old, or wait up to one second for new ones. To not slow down the robot even more, the last measurement was used even though it was likely to not have been taken at the exact robot position but on the way there. In some simple comparisons, this technique showed better result than the alternative thanks to its higher speed.

Gas Source Declaration in the Real Runs

In the robot lab, the gas source declaration technique from the original article was used. That is, the robot was considered to have reached the source as soon as it had hit something, or rather, had calculated a position that was not reachable without hitting an obstacle. The unreachable position was returned as the result of the search. In the original study, the box containing the leak measured about 20[cm] $\times$ 30[cm] in the horizontal plane. The corresponding "box" in the robot lab, the process module, measures 5[m] $\times$ 2.5[m] and the source is placed on the long side, about 1.5[m] from the nearest corner. Consequently, the robot may be far away from the source even if the "box" was hit. The mission of the robot is thus not just to hit the process module but to hit it as close to the leak as possible. As the robot virtually never loses the plume so bad that it hits the process module from the side, the process module can be seen as a wall rather than a box from the robots perspective. This limits the ways in which the robot can reach its goal to one direction, it cannot miss the target and rely on fail-safe mechanisms to steer it in from the side. Further, as the wind is not perpendicular to the rim of the process model, there is a risk that the robot will hit the model too early while still being in front of the leak, see Figure 4.3.

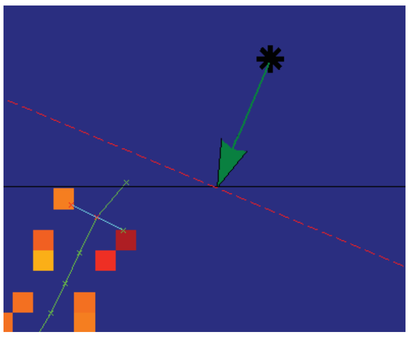
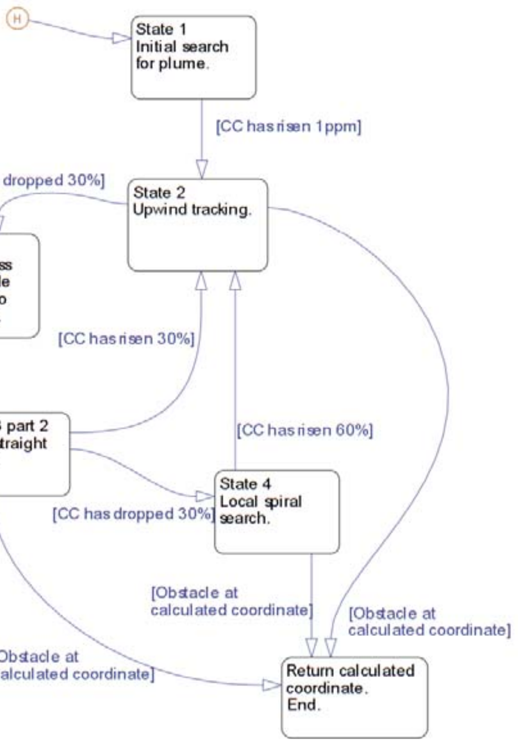


Figure 4.3: This figure shows how the robot, coming in from the lower left corner, hits the process module (black line) at an early stage as it is approaching the source (black star) from the left side. This is due to the fact that the rim of the process module is not perpendicular to the wind direction (arrow), as the illustrating dashed line is. As Figure 4.1



H
State 1 Initial search for plume.
[CC has risen 1ppm]
[CC has dropped 30%]
State 2 Upwind tracking.
State 3 part 1 Turn 90 degress towards the side with lowest R to reenter plume.
[CC has risen 30%]
[CC has risen 60%]
State 3 part 2 Move straight ahead.
State 4 Local spiral search.
[CC has dropped 30%]
[Obstacle at calculated coordinate]
[Obstacle at calculated coordinate]
[Obstacle at calculated coordinate]
Return calculated coordinate. End.

Figure 4.4: This state diagram shows the states and transitions of the real implementation for industrial robots of the gas localization algorithm described in [8]. The statements in brackets are conditions that have to be fulfilled to make the transitions in question. In these transition conditions, CC is referring to the mean of the concentrations measured by the left and right sensor.

4.2 Strategy based on Time Averaged Model

The second algorithm to be implemented was the time averaged model-based source location estimator of Ishida [10]. The algorithm, which has been explained in Section 2.2.3, describes how the location of a gas source can be estimated using information about wind and diffusivity together with measurements of gas concentration. However, the paper does not evaluate where the measurement should best be taken, *i.e.*, how the robot should be steered. The robot is only moved in a “mow the lawn”-pattern and it is possible that improvements may be made in this area.

```
SpeedSetPoint = 0.1
if($C_C < (C_{C\_lastUpdate}$ *1.01)){                // 1% concentration rise
    SpeedSetPoint += 0.016
    if(SpeedSetPoint > 0.16){
        SpeedSetPoint = 0.16
    }
    $C_{C\_lastUpdate} = C_C$
}
elseif($C_C > (C_{C\_lastUpdate}$*0.99)){             // 1% concentration drop
    SpeedSetPoint = 0.08
    $C_{C\_lastUpdate} = C_C$                         // update $C_{C\_lastUpdate}$
}
```

Table 4.5: Pseudo code for the speed control in the upwind steering, state 2, of the real implementation for industrial robots of Ishida's Transient-Based Reactive System [8].

4.2.1 Simulation

Simulation Environment

The same simulation environment as in Section 4.1.1 was used. This means that both the simulation environment and the algorithm itself were based on the same equation (2.1). Even though strong noise was included, the fact that both the environment and the algorithm used the same simplified picture of the world meant that the simulations were likely to provide unrealistically good results. For this reason, the simulations were mainly used as a way of testing the basic functionality of the implementation and for testing moving strategies.

Simulated Algorithm

The simulations were successful in recreating the algorithm of the original article [10] and estimation maps were generated. The result of such estimations can be seen in Figure 4.5. Apart from the noise of the simulated measurements described in Section 4.1.1, all parameters of the environment such as the turbulent diffusion constant and the wind parameters were usually available to the algorithm. The width of the membership functions was set to $4 \cdot 10^{-6}$[m^3/s].

Sample Patterns

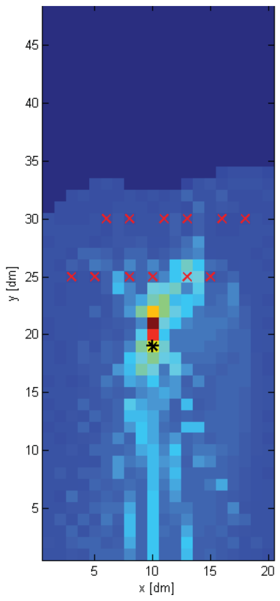
In the original paper [10] as well in a study of the algorithm where measurements were taken outdoors [11], samples were taken in a "mow the lawn" pattern. The robots were moved in lines perpendicular to the wind direction and made measurements at points along them as in Figures 4.5(a) and 4.5(b). To explore possible improvements in this area, a number of new sample patterns and strategies were tested. These are displayed in Figure 4.5. To stress the strategies, the turbulent diffusion constant was set 1.3 times higher in the estimation algorithm than in the simulation environment used. In all cases except for the combination with Ishida's transient based reactive algorithm, the robot stayed at each sample point until the simulated sensor value had stabilized, which often took up to a minute.

“Mow The Lawn” In Figure 4.5(a) and 4.5(b), samples are taken in the same pattern as in previous studies. As the value of the turbulent diffusion constant is exaggerated in the estimation algorithm, the samples taken far away from the source, see Figure 4.5(b), fails to make a good estimation of the distance to it. In this example, the distance from the source to the grid with the highest correlation grade is 0.5[m] for this pattern whereas it is 0.2[m] in the case where the samples are taken closer to the source, see Figure 4.5(a). Thus it are advantageous to “mow t1he lawn” close to the area where the source can be expected to be.

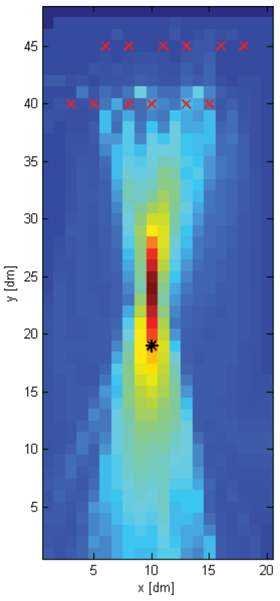
Scattered Samples Figure 4.5(c) shows the collective estimation from 5x2 samples scattered in the search area (note that this is 1x2 samples fewer than was used in the exemplifying estimations using the “mow the lawn” and “zigzag” patterns). In this example, the distance from the source to the cell with the highest correlation grade is 0.3[m].

Zigzag Pattern A strategy was invented where the sensors are moved a slightly randomized distance perpendicular to the wind towards the side downwind of the grid with the highest estimated correlation grade so far. In the same time, the tool is moved upwind by a fixed distance, see Figure 4.5(d). In this example, the distance from the source to the grid with the highest correlation grade is 0.1[m]. As seen, the strategy did very well in this test. However it was helped by the facts that a suitable number of samples fitted between the starting point and the source and that the last sample was taken very close to the source.

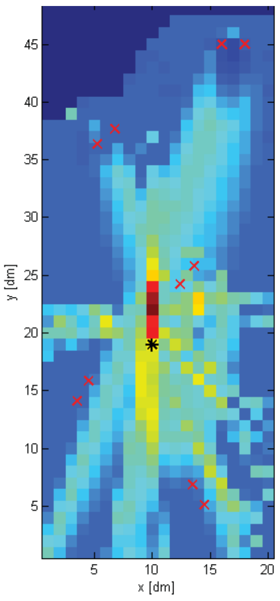
Combination With Ishida’s Transient Based Reactive Algorithm An attempt was made to combine the estimation algorithm of this section, Ishida’s strategy based on an time averaged model, with Ishida’s transient based reactive algorithm, simulated in Section 4.1.1. The robot was made to move according to the reactive algorithm and the samples taken during each cycle were used for making the estimation of the source’s location. The idea was that the user in this way would get a forecast of the location of the source before the reactive algorithm finished and also get an additional result in the end of the run and thus a more robust system. Estimations made at different stages of such a run are to be seen in Figure 4.6. As it turned out, the reactive algorithm was making the robot move far too fast for the estimations to be good. The measurements never stabilized on the true concentration of the grids, which is needed for the estimation algorithm. In general, the final estimation was however still fairly good thanks to the massive number of samples and because the last measurements usually were taken close to the source. However, as is illustrated by Figure 4.6(b), early estimation cannot be used as forecasts.



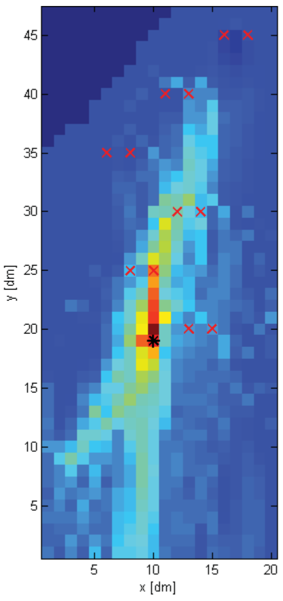
(a) Samples collected close to the source in a "mow the lawn" pattern.



(b) Samples collected far away from the source in a "mow the lawn" pattern.



(c) Only 5x2 samples collected in a scattered pattern.



(d) Samples in a zig-zag pattern.

Figure 4.5: Simulated location estimations of the gas source made with samples taken in four different patterns. The red crosses represent the sample points and the black stars marks the actual locations of the gas source. The wind is directed from the bottom to the top in all pictures.

4.2.2 Real World Tests

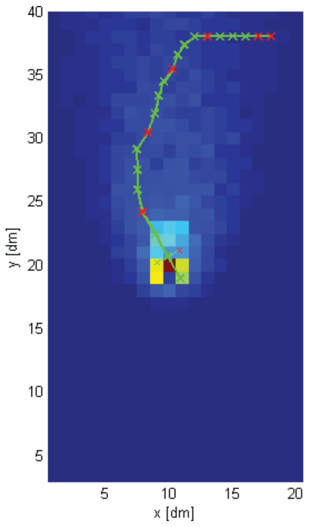
The core algorithm of the time average model-based strategy discussed in this section, the source-location estimator described in the original paper [10], is not reactive. Thus for most of the patterns described in Section 4.2.1, there is no need to keep the estimator online with the robot but the estimations may be made offline when the robot has finished its task and all samples have been collected. For this reason, the strategy was never fully integrated with the robot system during this study. Instead, a large number of samples were taken in the area in front of the source to provide data for parameter adjustments and preliminary evaluation of the strategy's performance in a real environment. In the end, a number of additional sample series were collected for the final evaluation. Sampling was also made during runs of Ishida's transient based reactive algorithm for testing it in combination with the estimator.

Difference Between Conditions of the Real and Simulated Tests

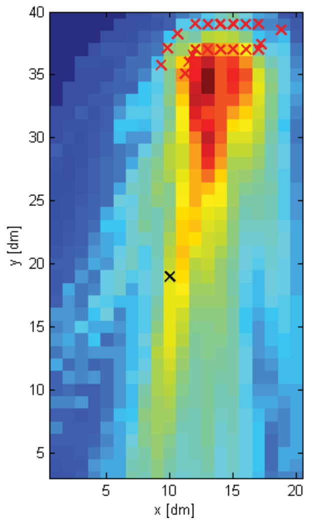
As both the simulation environment and the laboratory used in the evaluation of this algorithm are identical to the ones used in investigation of Ishida's transient-based reactive strategy, a description of the differences between the two environments can be found in Section 4.1.2. One difference of particular importance to the source-location estimation algorithm is the non-uniform properties of the artificial wind in the laboratory. As it is created by a single fan, its strength and direction vary as one moves perpendicular to the fan's direction. The wind strength also varies right in front of the fan, depending on the distance from it. Further, as the wind is concentrated to the area around the source and the ventilation is not very strong, released gas will spread in the laboratory, creating a background concentration.

Sample Collection

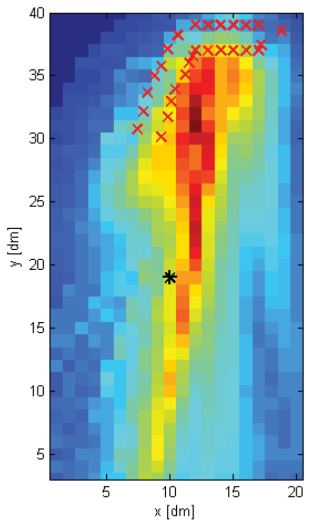
The samples used for the initial offline analysis were collected in an area about 2[m] $\times$ 2[m] big just downwind of the source. In addition to the 74 samples taken there, another 16 were collected further away from the source to collect information about the background gas concentration. At each point, samples were taken over a time of 90[s] to let the sensors stabilize their outputs. The mean of the last 30 samples at each point were saved as the measured concentration of that point. The samples were collected at a time when the gas leak had been active for some time so that the concentration levels in the room where reasonably stable. In addition to these samples, 13x10 samples were taken at a later occasion in the "mow the lawn"-fashion to try out estimations using that sample pattern for real. The time spent at each sample point remained the same. All of the sampling mentioned above was done with exact information about the location of the tool, *i.e.*, without the localization-problems connected to the path planner described in Section 3.5.1.



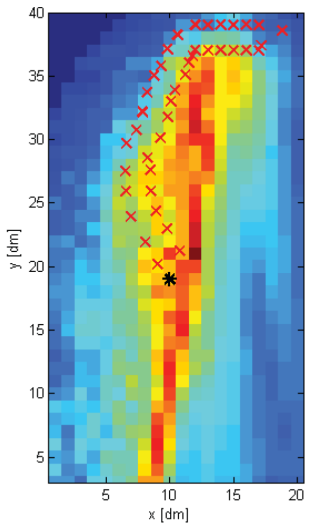
(a) The path taken by the robot controlled by Ishida's trasient-based reactive algorithm.



(b) Map showing the result of source-location estimation from samples taken during the first 10 iterations.



(c) Map showing the result of source-location estimation from samples taken during the first 14 iterations.



(d) Map showing the result of source-location estimation from samples taken during all 20 iterations.

Figure 4.6: Simulated location estimations of the gas source made with samples taken during the execution of Ishida's transient-based reactive algorithm. The three correlation grade maps show estimations made at different stages of the run. The red crosses represent the sample points and the black stars marks the actual locations of the gas source. In all pictures, the wind is directed from the bottom to the top.

The real implementation of Ishida's transient-based algorithm, see Section 4.1.3, was run 13 times with one sample taken during each iteration of the algorithm. The algorithm used the higher sensor dependency of Table 4.4 as the second last line read in that table read "SteerAngle = 0.3 × FlowAngle + 0.11 × SensorBias". Due to technical obstacles, exact localization of these sample points was not as easily achieved as for the other sample patterns and the localization uncertainty of about 5[cm] described in 3.5.1 was not compensated for.

Parameter Estimation and Tuning

As mentioned in Section 2.2.3, a correct value of the turbulent diffusion coefficient K is of the essence if good gas source location estimations should be achieved. In [11], two techniques to estimate K are proposed. One of these were used to estimate the parameters of the laboratory. Not only K was estimated but also the wind speed U, which in the lack of an anemometer could not be measured. The parameter estimation is done by running gas source location estimations with different parameters. For each parameter value, the root-mean-square distance, σ, to the source, is calculated for the grids with a correlation grade above 80 % of the highest correlation grade as described in (4.2),

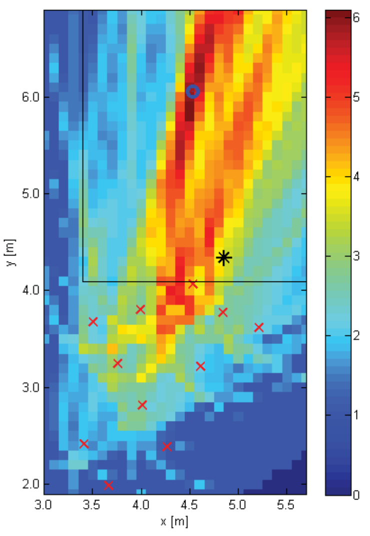
$$\sigma = \sqrt{\frac{1}{N}\sum_{i}^{N}(d_i)^2} \tag{4.2}$$

where d_i is the distance of grid i from the actual source location and N is the number of cells with correlation grades over 80 % of maximum. This gives a value, σ, of how close to the source the source location estimation got, and thereby, how applicable the environment parameters in question are. Because of long processing times, it was not possible to use the entire set of samples for the parameter estimations. Instead, two different subsets were used, one where the sample points were spread over the entire area and one where they were ordered in two lines perpendicular to the wind direction. As an outcome of this procedure, K was estimated to 0.108 and U to 0.8[m/s].

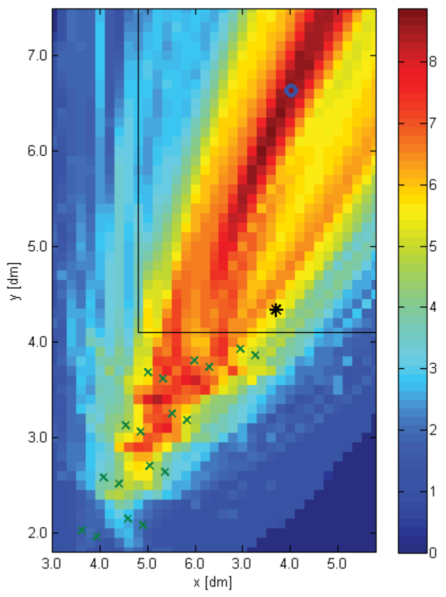
The background gas concentration in the laboratory was measured to 15 ppm at the time of sampling. An offset of -15 ppm were therefore applied to the samples used by the source location estimation algorithm.

4.2.3 Final Algorithm

The final source-location estimator algorithm only differed from the one described in the original paper [10] in terms of the parameters described under the previous paragraph. Not parts of the algorithm itself, new sample patterns were also introduced. The different sample patterns simulated as described in Section 4.2.1 were tested in combination with the initial real measurements. The outputs of estimations using the scattered pattern, see Figure 4.7(a), and the zigzag technique, see Figure 4.7(b), were deemed deficient at this stage. However, the "mow the lawn"-pattern and the combination with Ishida's transient-based reactive algorithm showed better promise and were used in the final tests.



(a) Scattered sample pattern.



(b) Zig-zag sample pattern created as described in Section 4.2.1, however without randomization of the sideway moving distance.

Figure 4.7: Plots of correlation grades from an estimation using samples from the large, pre-collected, sample set, gathered for the development of the strategy based on the time averaged model. The sample points used are marked with red crosses and the black line shows the borders of the process module. The blue circles show what in Section 5.2 is called center of the area with high correlation grades. The black stars mark the actual locations of the source.

4.3 Infotaxis

From the start of this work, successful real world implementation of infotaxis was far more unsure than it had been for the two previous strategies. The algorithm had only once before been tested for real [13] and never with gas sensors. In the previous practical study [13], heat sensors were used to make quick response times available and the authors claimed to have been unable to find suiting gas sensors despite extensive testing. They are correct that the response time of gas sensors such as those used in this study are well below the level required for resolving small gas filaments passing the sensors at the speed of the fan generated wind. Further, the sampling rate of 1[Hz] also puts a limit to the time resolution. Because of these limitations, a lot of the work with infotaxis consisted of finding ways to extract sufficient information from the slow sensors. This process is explained in Section 4.3.2, first however, the pure simulations of infotaxis are described.

4.3.1 Simulation

Simulation Environment

As the input to infotaxis is binary cues and not concentration levels, the simulation environment used for the two previous strategies was not suitable and therefore a entirely different one was used. A field of detections and non-detections was created by using the same model as infotaxis itself, the one described by (2.5). The grid-size was also identical to that of the simulated algorithm, 80 times 90 cells. The function of (2.5) was applied to all cells, calculating the mean rate of encounter. This determined the probability of the robot encountering a “hit” in each cell. For every iteration of the infotaxis algorithm, an encounter or non-encounter was randomly generated based on this calculated probability.

Simulated Algorithm

The simulated algorithm made no lapse from the one that was described in Section 2.2.5. These simulations were mainly used as first tests of the implementation to ensure it was working correctly. At this point, some of the sensor studies had already taken place and parameters were used to make the simulation reasonably truthful to the real conditions. The algorithm always had full access to the environmental parameters of the simulation environment, *i.e.*, everything but the location of the source and the resulting encounter rate map, to use for its inner model of the environment, which itself was identical to the one used to create the simulation environment. The algorithms of infotaxis and in particular those involved in entropy-calculations turned out to require a lot of processing and depending on grid size and number of elements in (2.7), the calculation times could be quite long. Each iteration usually took about 10[s] which is a long time as a full run often required more than 200 iterations.

4.3.2 Real World Tests

To enable use of infotaxis together with a slow sensor system, a way to extract the necessary cues from the sensor data had to be found. As when it came to the strategy based on the time averaged model (see Section 4.2), samples collected beforehand could be used to make some test of infotaxis offline. These tests were however mostly limited to parameter configuration and try-outs of the probability estimator. The collected data was also used to solve the problem of limited information caused by the slow sensors and sampling system.

Available Sensor Data

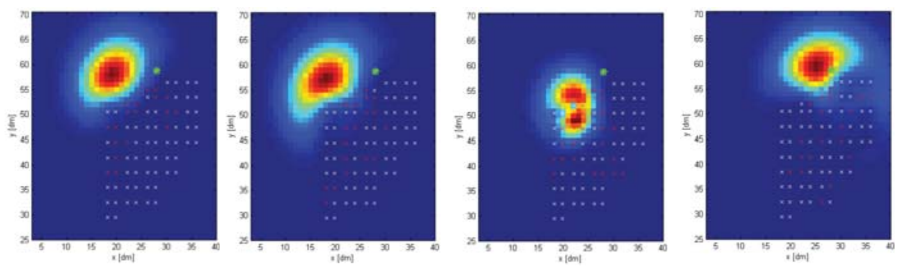
As the sensor system was so slow, an investigation had to be undertaken into how to extract useful data. The information had to have the right form, *i.e.*, binary cues that are more frequent in the vicinity of the source but independent of the amount of gas released. An attempt was made to decrease the sensor response time by removing the protecting cap that covers the sensor element. This did however only introduce a lot of noise. No additional information regarding the gas distribution could be acquired.

Instead data from standard sensors was used and the big sample set collected for the development of the system based on the time averaged model (see Section 4.2.2 could be used in the analysis.

Three different approaches for extracting binary cues from the sensor data were evaluated: The variance-method, the steepest slope-method and the counter-method. The input to each method was a number of consecutive samples taken at the same point, divided by their mean to be independent of the amount of gas released. The three methods either calculated the variance of these values, the mean of the few largest sample to sample increases (steepest upward slopes) or the number of samples exceeding the mean of the preceding few samples by a certain value. If these calculated values reached certain thresholds, detection was generated. The three different techniques were applied to the samples taken 21-102[s] after the sensors arrived at every point of the big sample set collected for the development of the system based on the time averaged model. The results were compared to what the model using (2.4) would predict. After some parameter tuning, the steepest slope- and highest variance-methods showed the best results. As the steepest slope-method was thought to be the one best suited for smaller sample sets, it was chosen to be used in the infotaxis implementations.

Simulations Using Real World Data

Fair estimations of the gas source location were extracted using the sample set collected for the development of the system based on the time averaged model and cues generated by the steepest slope-method. It was found that the results usually were as good using only ten samples as when using the entire set, thus enabling more frequent movements in the final application. However, the ten first samples at every point proved unusable as the sensors needed some time to tune in to the new general concentration level. Estimations created from samples taken different amounts of time after the sensor arrived at the sample positions are displayed in Figure 4.8.



(a) Samples 25-100 (b) Samples 12-21 (c) Samples 72-81 (d) Samples 82-91

Figure 4.8: Probability maps created from samples of the set collected for the development of the system based on the time averaged model. A different subset of the samples taken in every point is used for the different estimations. The white crosses mark the sample positions, the red crosses mark "hits" and the green star is the real position of the source. For purpose of calculations, the coordinate system is rotated compared to the one used in the laboratory so that the wind is blowing in the negative y-direction.

The large sample set was also used for simulated runs of the entire infotaxis algorithm, including navigation. At each point, a randomly chosen series of ten consecutive samples was used for the calculations by the steepest slope method. The sample series were taken sometime in the interval of 21-102[s] after the sensor reached the point in question. The grid of samples was too small for the runs to be considered entirely realistic but showed good promise. One of these runs is depicted in Figure 4.9.

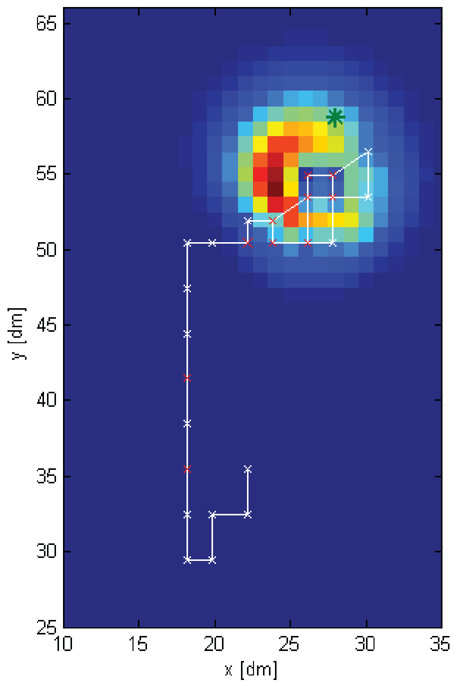


Figure 4.9: A simulated run of infotaxis using samples from the set collected for the development of the system based on the time averaged model. The white line shows the path of the robot, the red crosses mark “hits” and the green star is the real position of the source.

4.3.3 Final Algorithm

The laboratory implementation of infotaxis was based on the system developed during the simulations based on real sensor data (see Section 4.3.2). The parameters obtained during these studies were reused as well as the method of letting the highest steepest slope relative to the concentration during a ten second period determine if a cue was detected or not. In an effort not to let initial transients caused by change of location interfere, the sampling interval in the final tests reached from the twelfth to the twenty-first second after the robot reached a certain position.

Adaptations to the Environment As each run of the infotaxis algorithm took a considerable amount of time, usually more than twenty minutes, the conditions in the laboratory changed during the course of the many runs made for the comparative study. Parameters such as temperature and humidity in the laboratory naturally change over the course of a day and so does the level of accumulated ethanol in the air. As the algorithm extracting "detections" from the sensor data contains a division by the mean concentration level at the position in question, it is crucial that the concentration level is correct. To compensate for variations, the offset was adjusted ahead of almost every run. Small adjustments were also made to the release rate, R, of the algorithms inner model and to the slope steepness required for a detection. Even so, results varied greatly from run to run. The parameters used were, as described in Section 2.2.5, release rate R = 0.7-0.8[particles/s], lifetime τ = 1[s], diffusivity D = 3, mean wind V = 1.5[m/s] and detector- or patch-size a = 0.1[m]. Further, the increase made during the steepest upward slope of a sample set divided by the sets mean concentration had to exceed 0.06-0.08 to be counted as detection.

Space The grid was composed by 27x27 tiles each covering 25x25[cm]. The cell size was chosen to be that large to make its sides as long as the step size of the robot, which in turn had that size so that the search would not take too long even though each iteration needed 25-30[s]. Not all cells were accessible by the robot as some were occupied by obstacles (such as the process module or the tool stands) while some cells were simply out of reach of the track mounted robot. In the cases when the infotaxis algorithm tried to steer the robot to such a inaccessible location, *i.e.*, when the pathplanner refused to perform a move, the robot was instead moved in the direction that offered the second largest expected decrease in entropy. If that was not possible, a move was made towards the third lowest expected entropy and so on. The same was true if the robot reached the border of the grid.

Dual Sensors To make the conditions of the real infotaxis runs more equal to those of the other algorithms (see Sections 4.2.3 and 4.1.3), both sensors were used simultaneously, together taking two sets of samples during each iteration. The steering algorithm of infotaxis was however acting as if only one sensor existed but with a doubled chance of making detections: The entropy estimations according to (2.7) were calculated for the tool-coordinates, situated right between the two sensors, with the value of the release rate R doubled. The problem of uncertainty in sensor localization mentioned in Section 3.5.1 was not present during the runs of the infotaxis algorithm and localization was exact down to a few [mm].

Chapter 5

Results

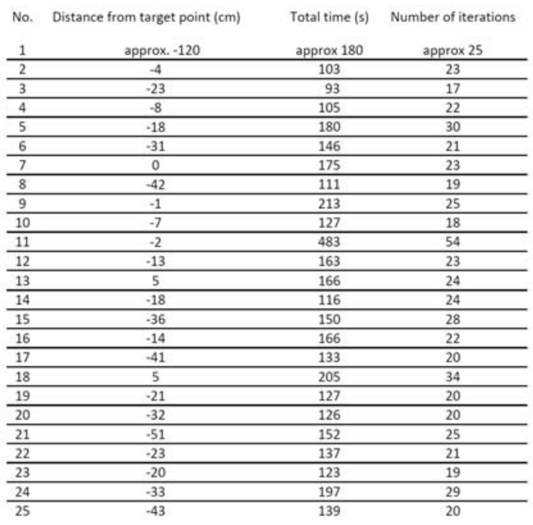
5.1 Ishida's Transient-Based Reactive Strategy

The final incarnation of the reactive algorithm of this study, described in Section 4.1.3, was run 25 times. A visualization of one of the runs is displayed in Figure 5.1. The quality of a run was judged by the distance from the target point to the point where the robot hit the process module, or rather, would had hit the process module, had the robot not been stopped. The target point is the point on the side of the process module straight downwind of the source. It is marked by an arrow in Figure 5.1(a). This is where the robot would had hit the process module, had it been in the middle of the plume heading straight for the source. The results of the 25 runs are displayed in Table 5.1.

Out of the 25 runs, 8 hit the process module within 10[cm] of the target point. Another 8 runs got within 25[cm] of that point, which corresponds to the combined width of the target and the robot in Ishida's original study. All of the remaining runs did hit the process module and only one was more than 51[cm] off the target point. The mean completion time was 163[s], the mean distance to the target 24.4[cm] and the average number of iterations 24.2.

5.2 Strategy Based on Time Averaged Model

The final source-location estimator was run in combination with two different sampling strategies. Both the two line "mow the lawn"-strategy of the original article [10] and a combination with Ishida's transient-based reactive algorithm, described in Section 4.1.3, were used. The quality of an estimation was judged by the distance from the source to the center of the area with high grades. This center point was defined as the mean location of cells with correlation grades higher than 93 % of the maximum correlation grade, weighted with the parts of the values exceeding 93 %. To enable a fair comparison with the results of Ishida's transient-based reactive algorithm, the distance corresponding to the double-arrow in Figure 5.4 was calculated for each run. This is the distance in the x-direction (parallel to the process module's side) between the center



No.	Distance from target point (cm)	Total time (s)	Number of iterations
1	approx. -120	approx 180	approx 25
2	-4	103	23
3	-23	93	17
4	-8	105	22
5	-18	180	30
6	-31	146	21
7	0	175	23
8	-42	111	19
9	-1	213	25
10	-7	127	18
11	-2	483	54
12	-13	163	23
13	5	166	24
14	-18	116	24
15	-36	150	28
16	-14	166	22
17	-41	133	20
18	5	205	34
19	-21	127	20
20	-32	126	20
21	-51	152	25
22	-23	137	21
23	-20	123	19
24	-33	197	29
25	-43	139	20

Table 5.1: Results from 25 runs of the final laboratory implementation of Ishida's transient-based reactive system [8] described in Section 4.1.3. A visualization of run number 12 is shown in Figure 5.1.

of the area with high grades and an imaginary line parallel to the wind direction that goes through the location of the source, hereafter called the "windline".

5.2.1 Samples Collected in a "Mow the Lawn"-Pattern

The results of the runs using the "mow the lawn"-pattern are displayed in Table 5.2. Run number 11 is visualized in Figure 5.2.

Of the 13 runs, one managed to place the center of the area with high grades within 10[cm] of the windline. In addition, 5 more high grade center points were within 25[cm]. The mean distance between the center of the area with high grades and the wind line in the x-direction was 29.8[cm]. The mean distance between the area with high grades and the *source* was 36[cm]. The same 760[s] or 8 iterations of sampling, using dual sensors, were required before every estimation. At each point, sampling was made for 90[s]. It should be noted that this time was not optimized and could probably be decreased without any impact on the results of the estimation. In addition to that time, some 1-2[s] per sample iteration were needed to make the estimations.

5.2.2 Samples Collected with Ishida's Transient-Based Reactive Algorithm

The results of the estimations using samples collected during runs of Ishida's transient-based reactive algorithm are displayed in Table 5.3. Estimation number 2 is visualized in Figure 5.4. As no sampling function initially was implemented, sampling was only

No.	Δx, wind line that crosses source to high grades center (cm)	Dist. perpendicular to wind dir., source to high grades center (cm)	Dist. in wind dir., source to high grades center (cm)	Abs. dist., source to high grades center (cm)
1	-16	-15	56	58
2	-19	-17	-6	18
3	-19	-18	-28	34
4	-12	-11	-40	42
5	-17	-15	72	74
6	-19	-18	-58	61
7	-19	-17	-39	42
8	-15	-14	-16	21
9	-24	-23	12	26
10	-22	-21	-15	26
11	-18	-16	-13	21
12	-29	-27	3	27
13	-20	-18	-6	19

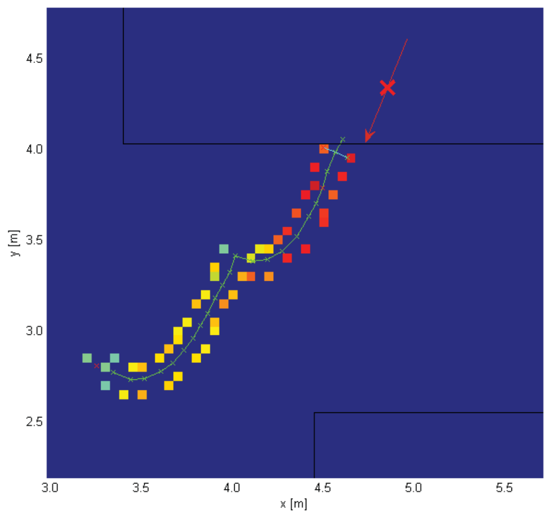
Table 5.2: Results from the 13 runs of the source-location estimator 4.2.3 using the "mow the lawn" sampling-pattern. A visualization of run number 11 is shown in Figure 5.2. An explanation of the first table header can be found in the beginning of Section 5.2

active during the last 12 runs of the reactive algorithm and estimations were only made based on these runs. As described in the caption of Table 4.4, these runs happened to be under a higher stronger influence of the gas sensors than the previous runs of Ishida's transient-based reactive algorithm.

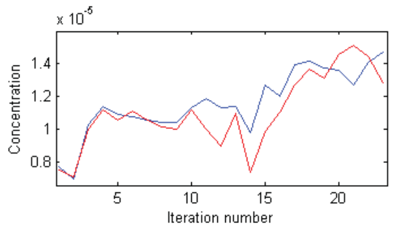
No.	Δx, wind line that crosses source to high grades center (cm)	Dist. perpendicular to wind dir., source to high grades center (cm)	Dist. in wind dir., source to high grades center (cm)	Abs. dist., source to high grades center (cm)	Total time (s)	Number of iterations
1	-63	-58	12	60	116	24
2	-40	-37	-24	44	150	28
3	-42	-38	-47	61	166	22
4	-50	-46	-12	47	133	20
5	-29	-27	-16	32	205	34
6	-34	-31	10	32	127	20
7	-59	-54	-26	60	126	20
8	-55	-51	-8	51	152	25
9	-29	-27	51	57	137	21
10	-50	-46	22	51	123	19
11	-50	-46	11	47	197	29
12	-2	-2	3	4	139	20

Table 5.3: Results from the source-location estimator 4.2.3 using the samples collected during 12 runs of Ishida's transient-based reactive algorithm. These runs correspond to run 14-25 in Table 5.1. A visualization of estimation number 2 is shown in Figure 5.2. An explanation of the first table header can be found in the beginning of Section 5.2

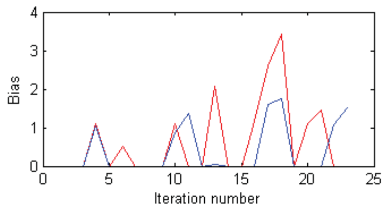
Of the 12 runs, one managed to place the center of the area with high grades within 10[cm] of the windline but none more high grade center points were within 25[cm]. The mean distance between the center of the area with high grades and the wind line in the x-direction was 41.8[cm]. The mean distance between the area with high grades and the *source* was 45.5[cm]. The mean completion time was 148[s] and the average number of iterations was 23.5.



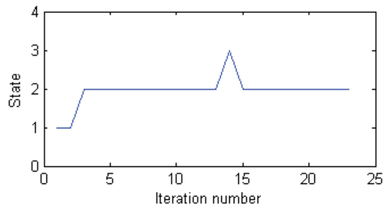
(a) The path of the tooltip and sensors. The color shows the strength of the measured concentration. The "X" shows the location of the gas source, the arrow the wind direction and the black boxes represent the obstacles namely the process module at the top and a tool stand at the bottom.



(b) Sensor readings of the run. The red and blue line represents the left and right sensor, respectively.



(c) The biases on the steering (calculated according to Table 4.4) from the left (red) and right (blue) sensor.



(d) Operating states of the algorithm.

Figure 5.1: Navigation data of a real run of Ishida's transient-based reactive strategy, number 12 in Table 5.1.

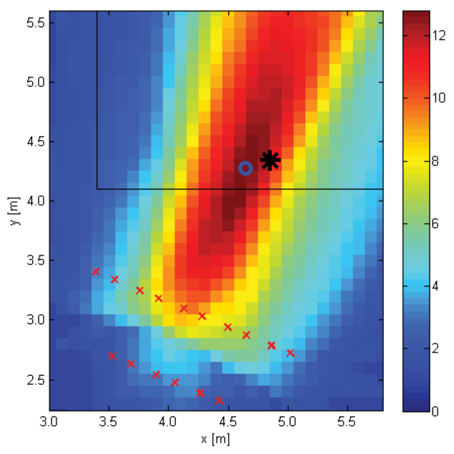


Figure 5.2: Plot of correlation grades from an estimation using samples collected in the “mow the lawn”-pattern. The result of this run is displayed as number 11 in Table 5.2. The sample points are marked with red crosses and the black line show the borders of the process module. The blue circle shows what in Section 5.2 is called center of the area with high grades. The black star marks the actual location of the source.

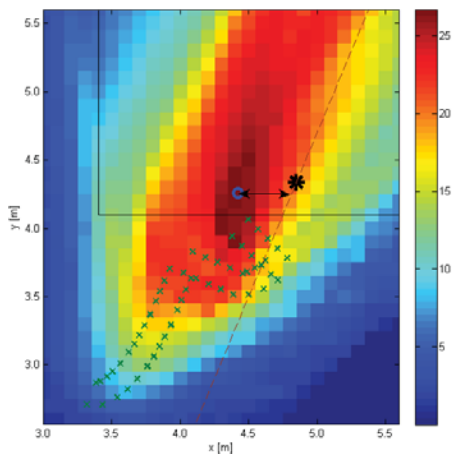


Figure 5.3: Plot of correlation grades from an estimation using samples collected during a run of Ishida’s transient-based reactive algorithm, see Section 4.1.3. The result of this estimation has the number 2 in Table 5.3. The sample points are marked with green crosses and the black line shows the borders of the process module. The blue circle shows what in Section 5.2 is called center of the area with high grades. The black star marks the actual location of the source. The line crossing the source is the windline, an imaginary line parallel to the wind direction that runs through the source.

5.2.3 Full Combination with Ishida's Transient-Based Reactive Algorithm

The combination between the estimations based on the time averaged model and Ishida's transient-based reactive algorithm were taken one step further. The results of 5.2.2 and 5.1 were combined to equip the reactive system with the ability to estimate the source's location in a second dimension without risking to lose precision. The result of the reactive algorithm decided the position in the dimension perpendicular to the wind direction while the coordinate in the wind direction was the wind direction coordinate of the high grades center from the estimation based on the time averaged model. The distance between the calculated position and the "windline" mentioned earlier in Section 5.2 was thus not changed compared to the standard version of Ishida's transient-based reactive algorithm. In the same time a measure in the wind direction was added. The results are compiled in Table 5.4. The mean distance between the source and the position resulting from the combination was 35.9[cm].

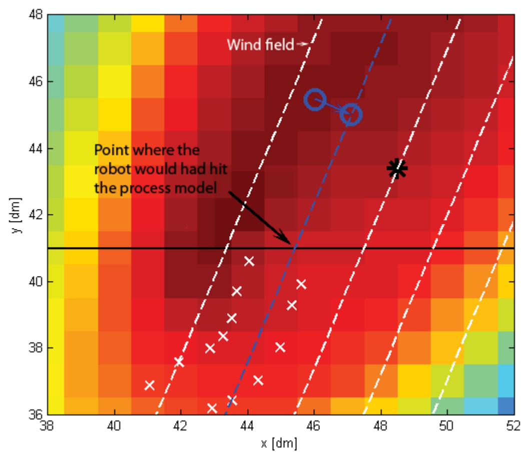
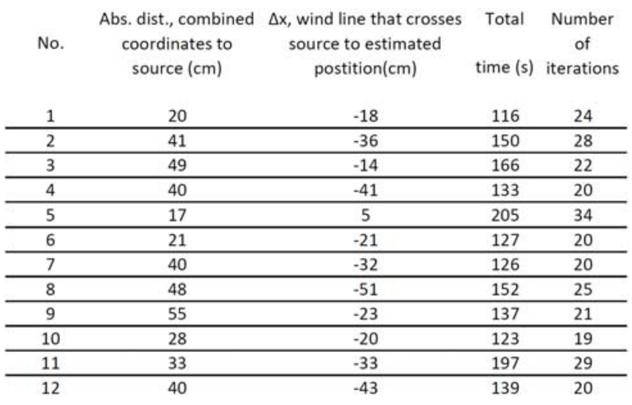


Figure 5.4: Plot showing how the results of Ishida's transient-based reactive algorithm from Section 5.1 and the results of time averaged model based estimation of Section 5.2.2 are combined in the "full combination" in Section 5.2.3. While the result of the model-based estimation is used in the wind direction, the result of the reactive algorithm is used in the perpendicular dimension. The combined result (blue circle) is thus projected onto the line (blue) parallel to the wind direction that crosses the result-coordinates of the reactive algorithm. The black star marks the real location of the source and the black line is the rim of the process model.



No.	Abs. dist., combined coordinates to source (cm)	Δx, wind line that crosses source to estimated postition(cm)	Total time (s)	Number of iterations
1	20	-18	116	24
2	41	-36	150	28
3	49	-14	166	22
4	40	-41	133	20
5	17	5	205	34
6	21	-21	127	20
7	40	-32	126	20
8	48	-51	152	25
9	55	-23	137	21
10	28	-20	123	19
11	33	-33	197	29
12	40	-43	139	20

Table 5.4: Combination according to 5.2.3 of the results Ishida's transient-based reactive system and the estimations based on the time averaged model using samples from the same reactive runs. The twelve combinations corresponds to all the runs of Table 5.3 and run number 14-25 of Table 5.1. The total time refers to the sampling time, in addition some 1-2[s] per sample iteration are needed for the estimator.

5.3 Infotaxis

The final infotaxis algorithm, described in Section 4.3.3, was run 12 times and the results are displayed in Table 5.5. The runs started in positions close to the starting position used in the real runs of Ishida's transient based reactive algorithm (see Section 4.1.3) and with the same heading as used by that algorithm. A few slightly different starting positions were used to create variation between the different runs which would otherwise have used identical paths until the first detection was made. Each run of the algorithm ended either when the robot had repeated a series of three consecutive moves, or when the algorithm had been running for 30 minutes. The quality of an estimation was judged by the distance from the source to the mean location of cells with an estimated probability of containing the source higher than 93 % of the highest probability of the grid. This location is hereafter called the "high probability center". This "high probability center" is thus a concept identical to the "high grades center" of Section 5.2 except that the algorithms they are used for present their results in different ways, in probabilities and in correlation grades. As for the results of the time averaged model based strategy (see Section 5.2 and double arrow in Figure 5.4), a distance parallel to the side process module was calculated to enable fair comparison with the results of Ishida's transient-based reactive algorithm. This was the distance from the high probability center to the "wind line", an artificial line stretching straight downwind from the source. A run was deemed to be a failure if the area estimated to have a high probability of containing the source was not clearly within the grid map used for the calculations (at least 1.5[m] away from the source). An example of a relatively successful run is displayed in Figure 5.5.

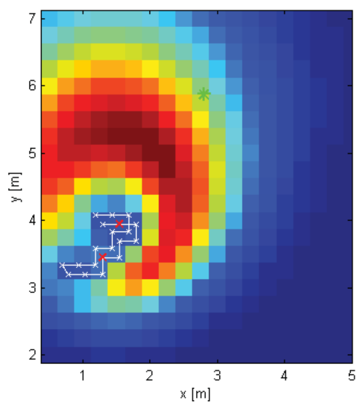
No.	Δx, wind line crossing source to high probability center(cm)	Dist. perpendicular to wind dir., source to high probability center (cm)	Dist. in wind dir., source to high probability center (cm)	Abs. dist., source to high probability center (cm)	Total time (s)	Number of iterations
1	-87	-80	38	88	960	32
2	49	45	-187	193	1680	57
3	-33	-30	-37	48	1440	42
4	-33	-30	-12	32	900	30
5	Failed					
6	Failed					
7	-73	-68	-125	142	1680	56
8	-5	-5	-37	38	1020	35
9	Failed					
10	Failed					
11	-19	-18	-50	53	1020	33
12	Failed					

Table 5.5: Results of the 12 runs of infotaxis (see Section 4.3.3). A visualization of run number 3 is displayed in Figure 5.5.

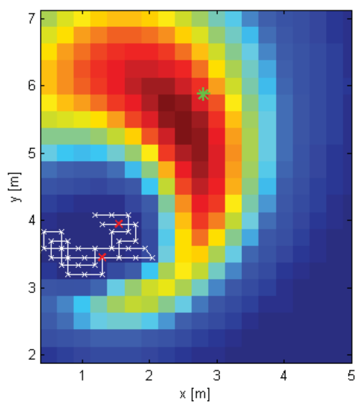
Only one of the twelve runs placed the probability center within 10[cm] of the wind line and another run got within 25[cm]. Five runs were deemed to have failed completely while the other runs had a mean completion time of 1243[s] and an average result 42.6[cm] away from the *windline*. The non-failure runs on average used 40.7 iterations to complete their task and the mean distance to the *source* from their resulting probability centers was 84.8[cm].

5.4 Ishida's Transient-Based Reactive System in an Obstacle Free Environment

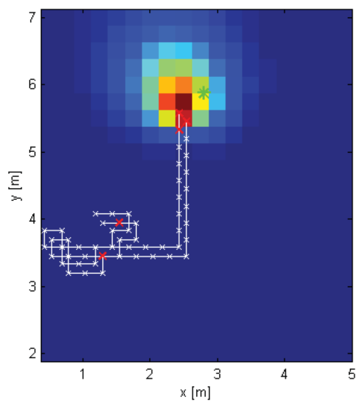
Outside the comparison of this study, the algorithm was run 10 times without the advanced pathplanner and collision detection activated. To avoid collisions, the robot was confined to an area that was known to be free of obstacles straight downwind of the gas source. All tool movements were linear. This lowered the cycle time to some 2.5-3.5[s] but the most important improvement was that the linear movements prevented the problems with discontinuity in the samples. The parameters were adapted to these changes. This implementation of Ishida's transient based reactive system was thus more faithful to the original algorithm and had a mean completion time of around 60[s]. Seven out of ten runs got within 10[cm] of the target point whereas nine were within 25[cm]. The tenth run too hit the process module but 45[cm] off the target. The mean distance to the target was 11[cm]. As these tests used simplified laboratory infrastructure, they are not a part of the comparative study. They do however show that the algorithm has further potential in case the laboratory control system can be improved and adapted to it.



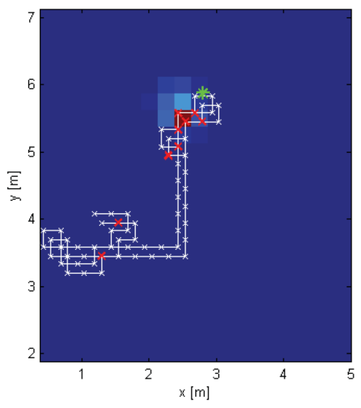
(a) After 10 iterations.



(b) After 20 iterations.



(c) After 30 iterations.



(d) After all 43 iterations.

Figure 5.5: Example of an successful run of Infotaxis corresponding to number 3 in Table 5.5. The colors shows the estimated probability of the source being in the cell in question, the color scales are individually adjusted to each figure. The actual source position is marked by the green star. The red crosses mark the points of detections. Compared to plots of the two previous algorithms, see for example Figures 5.1(a) and 5.2, the coordinate system is turned 23° so that the wind direction is the negative y-direction.

5.5 Comparison

For the sake of the distinctiveness of this section, the algorithms are numbered from one to five:

1. Ishisda's transient-based reactive system of Section 5.1.
2. The estimator based on the time averaged model using samples collected in a "mow the lawn"-pattern of Section 5.2.1.
3. Infotaxis of Section 5.3.
4. The estimator based on the time averaged model using samples collected during runs of algorithm 1, Section 5.2.2.
5. Combination of algorithm one and four, using number four in the wind direction and number one perpendicular to the wind according to Section 5.2.3.

Table

A compilation of the results of the algorithms tried out in this study is displayed in Table 5.6. When it comes to distance to target, two kinds of results are presented: Sideway distance to the "wind line", *i.e.*, the measure that was developed for fair comparisons between the algorithm one and the others, and absolute distance to the source. Algorithm one only provided one dimensional results in our setting, it did not produce any information regarding how deep into the process module the source was located. Because of this, it could not provide any absolute distance to the source and is disqualified from that part of the comparison.

Infotaxis

The first and most obvious conclusion that can be drawn from Table 5.6 is that algorithm number 3, infotaxis, is not yet ready for real use. Discarding the worst, "failed", runs, the results were still worse than those of any other algorithm in nearly every aspect. The estimations were further from the source and the runs took longer to perform, but even so, infotaxis showed some strength. The strategy was possible to integrate with the pathplanner to a higher degree than the other algorithms (see Section 4.3.3), which would have been advantageous in a more obstacle-filled environment. Further, a three dimensional implementation of infotaxis would be easy to create as this has already been described and simulated [16]. Infotaxis also solves the entire problem of detecting a leak and finding and declaring its source in a more complete way than the other algorithms. Even so, as the results clearly reveals significant improvements have to be made to gas sensors, especially in terms of speed, before infotaxis can be subject to real world gas source localization applications.

	Original algortihms			Combinations of algorithms	
	Algorithm 1 Ishida's transient-based reactive system (Section 5.1)	Algorithm 2 Estimation based on time averaged model. Sample collection: ``Mow the lawn" (Section 5.2.1)	Alg. 3 Infotaxis (Section 5.3)	Algorithm 4 Estimation based on time averaged model: Sample collection: Ishida's transient-based reactive system (Section 5.2.2)	Algorithm 5 Combination: Reactive in one dimension, time average model based estimation in the other (Section 5.2.3)
Number of runs	25	13	12	12	12 / 25
Distance to target/wind line in dir. parallell to module's side					
Mean distance (cm)	24.4	19.2	42.6[(1)]	41.8	24.4
Success rate: Within 10 cm	32 %	0 %	8 %	8 %	32 %
Success rate: Within 25 cm	64 %	92 %	17 %	8 %	64 %
Success rate: Within 50 cm	92 %	100 %	42 %	75 %	92 %
Absolute distance to source					
Mean distance (cm)	Not available	36.0	84.8[(1)]	45.5	35.9
Success rate: Within 25 cm	Not available	31 %	0 %	8 %	25 %
Success rate: Within 50 cm	Not available	77 %	25 %	50 %	92 %
Success rate: Within 75 cm	Not available	100 %	33 %	100 %	100 %
Mean completion time (s)	163	760[(2)(3)]	1243[(1)]	Time of Ishida's transient-based reactive system + estimation time[(3)]	Time of Ishida's transient-based reactive system + estimation time[(3)]
Mean number of iterations	24.2	8.0	40.7[(1)]	Same as Ishida's transient-based reactive system	Same as Ishida's transient-based reactive system
Possibility of adaptation to three dimensions	Hardly possible	Partly possible[(4)]	Easy	Hardly possible	Hardly possible

[(1)]Among the 7 runs not deemed as failures
[(2)]Not optimized
[(3)]The estimation adds some 1-2 seconds per iteration
[(4)]Much more complex model needed

Table 5.6: Compiliation of the results of the systems tested in practice.

Best Algorithm

Establishing which the best algorithm is, is not as easy as finding the worst one. Algorithm number five is clearly superior to the similar algorithm four. Thus it is a good idea to combine estimator results with reactive ones. In one dimension, algorithm two however reaches even higher precision while it in two dimensions performs even to algorithm five. On the other hand, algorithms one and five are considerably faster than algorithm number two. With the time factor taken into account, algorithm five is thus the most attractive overall choice of this comparison. This however depends on what factors are important. If time is not an issue, algorithm two seems to be better, and if only sideways results are of interest, the estimator-free algorithm one is quicker.

Potential of Further Improvements

As mentioned, the sampling times of algorithm two have not been optimized. It is however probably not possible to decrease them by more than 50 % without loss of precision, meaning the times will still be long. Further, as has been described in Section 5.4, it should be possible to double the precision and speed of algorithm one, most likely improving the performance of algorithm five as well, if only the control system is adapted to the task. Apart from this, the potential of further improvements of algorithm one, the reactive part of algorithm five, are limited. Algorithm one, with all its details and fall-back mechanisms, is clearly an optimization of reactive gas source localization in one plane. Thus the algorithm is already relatively mature and would have to be completely re-worked to function in three dimensions.

When it comes to the estimators of algorithms two, four and five, the model of gas dissipation (2.1) is a greatly simplified picture of the environment. If the estimator was to be improved, the time averaged model itself would probably have to be replaced with a more advanced model. The system of using fuzzy membership functions would however be possible to reuse. Depending on the model, such a change could also enable three dimensional estimations. Further, algorithm number five could be improved by a more sophisticated way of combining its sub-algorithms. For example, the estimator might be allowed to influence the sideway coordinate under certain conditions and the influence of different samples could be weighted differently in the estimation (the last improvement affecting algorithm four as well).

Chapter 6

Conclusions and Future Work

In our comparison of Section 5.5, a combination of Ishida's transient-based reactive system and simultaneous estimations according to Section 4.2.2 came out as the best current method for gas source localization in an industrial environment. Estimations using samples collected in a pre-defined symmetrical pattern did too provide good results. As a step towards a future industry implementation, both these algorithms should be further improved and adaptations for three dimensional searches should eventually be developed. The improved algorithms should be tested outdoors in realistic weather conditions and in larger sets. A wider range of source locations, wind directions and starting positions should be evaluated.

This study also showed how the absence of fast and reliable gas sensors is a major hindrance to commercialization of the evaluated techniques. Further sensor reviews and not least evaluation of industry standard sensors satisfying the ATEX-criteria[1] should form an important part of future work. In addition, this study has showed that the performance of the robot and communication systems in terms of movement and transmission rates is vital to some algorithms. This has to be taken into account when future systems indented for use in gas source localization applications are designed.

[1]The ATEX directive describes what equipment is allowed in environments with explosive atmosphere. The abbreviation derives from the French title: Appareils destinés à être utilisés en atmosphères explosibles.

Bibliography

[1] A. Bleicher, “The gulf spill’s lessons for robotics”, vol. 47, no. 8, pp. 9–11, 2010.

[2] G. Kowaldo and R.A. Russell, “Robot odor localization: A taxonomy and survey”, *The international Journal of Robotics Research*, vol. 27, no. 8, pp. 869–894, 2008.

[3] A.J. Lilienthal, A. Loutfi, and T. Ducket, “Airborne chemical sensing with mobile robots”, *Sensors*, vol. 6, no. 11, pp. 1616–1678, 2006.

[4] L. Marques, N. Almeida, and A.T. de Almeida, “Olfactory sensory system for odour-plume tracking and localization”, in *In Proceedings of IEEE Sensors*, 2003, vol. 1, pp. 418–423.

[5] O. Rochel D. Martinez and E. Hugues, “A biomimetic robot for tracking specific odors in turbulent plumes”, *Autonomous Robots*, vol. 20, no. 3, pp. 185–195, 2006.

[6] Q. Meng, F. Li, J.G. Li, S. Bai, and M. Zeng, “Multi-robot based chemical plume tracing with virtual odor-source-probability sensor”, in *In Proceedings of the Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, 2009, vol. 1, pp. 246–250.

[7] R. Deveza R.A. Russell, D. Thiel and A. Mackay-Sim, “A robotic system to locate hazardous chemical leaks”, in *In Proceedings of IEEE International Conference on Robotics and Automation*, 1995, vol. 1, pp. 556–561.

[8] H. Ishida, G. Nakayama, T. Nakamoto, and Moriizumi T., “Controlling a gas/odor plume-tracking robot based on transient responses of gas sensors”, *IEEE Sensors Journal*, vol. 5, no. 3, pp. 537–545, 2005.

[9] H. Ishida, T. Nakamoto, and T. Moriizumi, “Remote sensing and localization of gas/odor source and distribution using mobile sensing system”, in *IEEE International Conference on Solid State Sensors and Actuators, TRANSDUCERS*, 1997, vol. 1, pp. 559–562.

[10] T. Ushiku, N. Satoh, H. Ishida, and Toyama S., “Estimation of gas-source location using gas sensors and ultrasonic anemometer”, 2007.

[11] Y. Fukazawa and H. Ishida, “Estimating gas-source location in outdoor environment using mobile robot equipped with gas sensors and anemometer”, in *In Proceedings of IEEE Sensors*, 2009, pp. 1721–1724.

[12] J.A. Farrell, J. Murlis, X. Long, W. Li, and R.T. Cardé, “Filament-based atmospheric dispersion model to achieve short time-scale structure of odor plumes”, *Environmental Fluid Mechanics*, vol. 2, no. 1-2, pp. 143–169, 2002.

[13] E.M. Moraud and D. Martinez, “Effectiveness and robustness of robot infotaxis for searching in dilute conditions”, *Frontiers in Neurobotics*, 2010.

[14] S. Pang and J.A. Farrell, “Chemical plume source localization”, *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, vol. 36, no. 5, pp. 1068–1080, 2006.

[15] M. Vergassola, E. Villermaux, and B.I. Shraiman, “Infotaxis as a strategy for searching without gradients”, *Nature*, vol. 445, no. 7126, pp. 406–409, 2007.

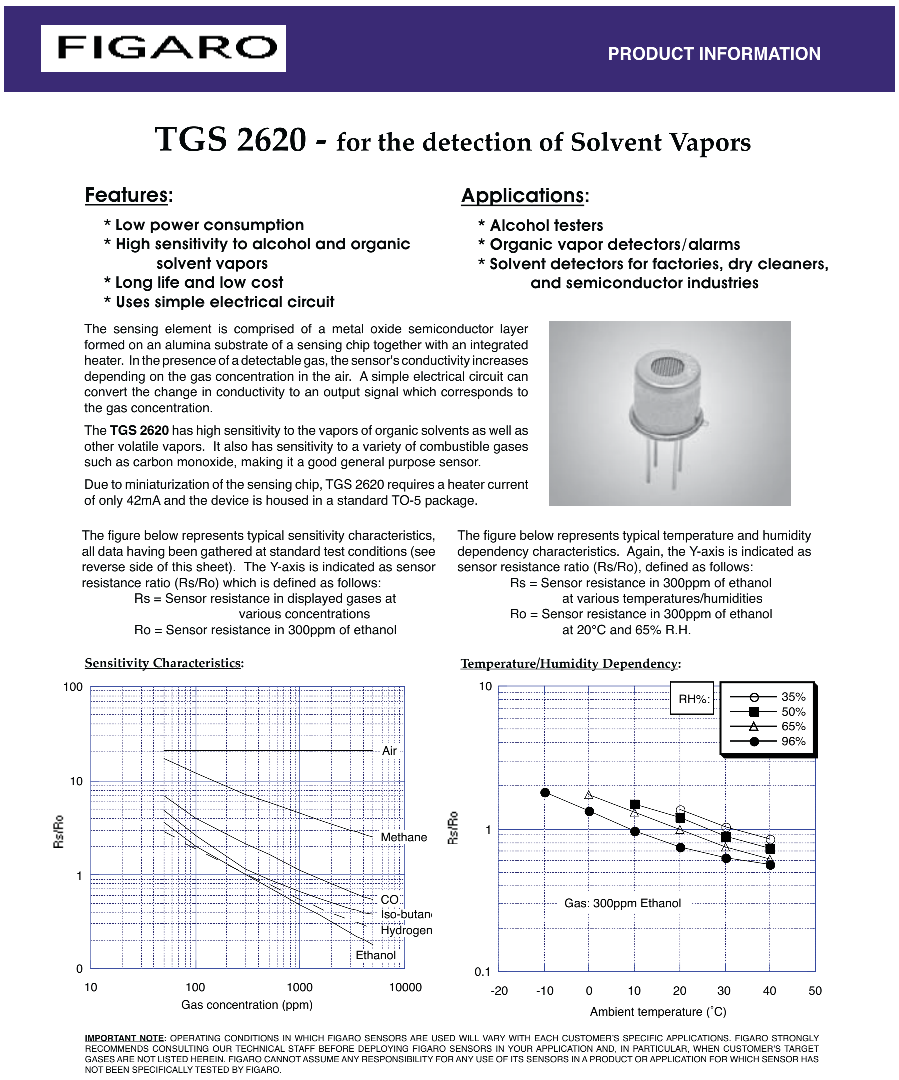
[16] J.B. Masson, M.B. Bechet, and M. Vergassola, “Chasing information to search in random environments.”, *Journal of Physics A: Mathematical and Theoretical*, vol. 42, no. 43, pp. 434009–434010, 2009.

[17] M. Wandel A. Lilienthal, A. Zell and U. Weimar, “Sensing odour sources in indoor environmentswithout a constant airflow by a mobile robot”, in *In Proceedings of ICRA, IEEE International Conference on Robotics and Automation*, 2001, vol. 4, pp. 4005 – 4010.

[18] Figaro Engineering Inc., “Product information, TGS 2620”, `http://www.figaro.co.jp/en/data/pdf/20091110165158_19.pdf`, Nov. 2009, Rev: 01/05.

Appendix A

Figaro TGS 2620



FIGARO

PRODUCT INFORMATION

TGS 2620 - for the detection of Solvent Vapors

Features:

* Low power consumption
* High sensitivity to alcohol and organic solvent vapors
* Long life and low cost
* Uses simple electrical circuit

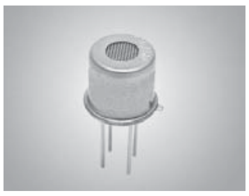
Applications:

* Alcohol testers
* Organic vapor detectors/alarms
* Solvent detectors for factories, dry cleaners, and semiconductor industries

The sensing element is comprised of a metal oxide semiconductor layer formed on an alumina substrate of a sensing chip together with an integrated heater. In the presence of a detectable gas, the sensor's conductivity increases depending on the gas concentration in the air. A simple electrical circuit can convert the change in conductivity to an output signal which corresponds to the gas concentration.

The **TGS 2620** has high sensitivity to the vapors of organic solvents as well as other volatile vapors. It also has sensitivity to a variety of combustible gases such as carbon monoxide, making it a good general purpose sensor.

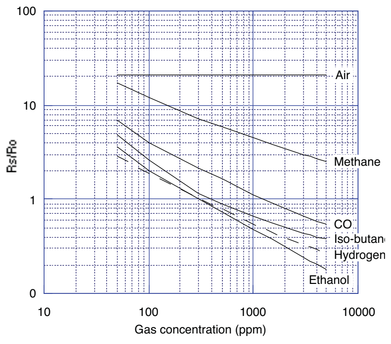
Due to miniaturization of the sensing chip, TGS 2620 requires a heater current of only 42mA and the device is housed in a standard TO-5 package.



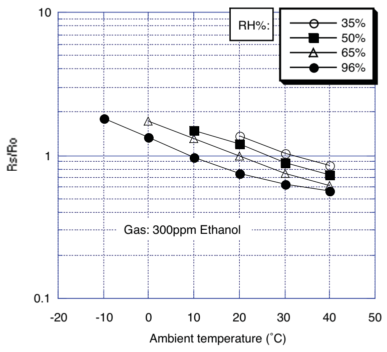
The figure below represents typical sensitivity characteristics, all data having been gathered at standard test conditions (see reverse side of this sheet). The Y-axis is indicated as sensor resistance ratio (Rs/Ro) which is defined as follows:
Rs = Sensor resistance in displayed gases at various concentrations
Ro = Sensor resistance in 300ppm of ethanol

The figure below represents typical temperature and humidity dependency characteristics. Again, the Y-axis is indicated as sensor resistance ratio (Rs/Ro), defined as follows:
Rs = Sensor resistance in 300ppm of ethanol at various temperatures/humidities
Ro = Sensor resistance in 300ppm of ethanol at 20°C and 65% R.H.

Sensitivity Characteristics:

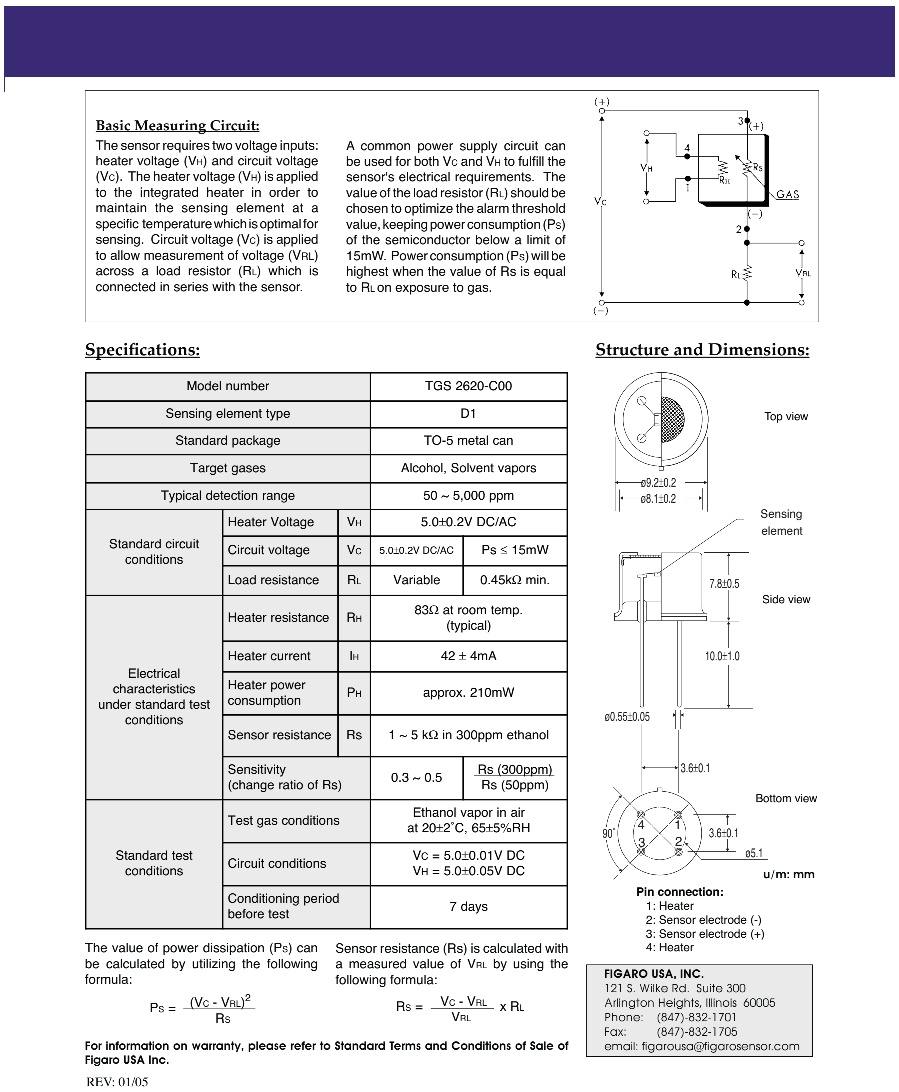


Temperature/Humidity Dependency:



IMPORTANT NOTE: OPERATING CONDITIONS IN WHICH FIGARO SENSORS ARE USED WILL VARY WITH EACH CUSTOMER'S SPECIFIC APPLICATIONS. FIGARO STRONGLY RECOMMENDS CONSULTING OUR TECHNICAL STAFF BEFORE DEPLOYING FIGARO SENSORS IN YOUR APPLICATION AND, IN PARTICULAR, WHEN CUSTOMER'S TARGET GASES ARE NOT LISTED HEREIN. FIGARO CANNOT ASSUME ANY RESPONSIBILITY FOR ANY USE OF ITS SENSORS IN A PRODUCT OR APPLICATION FOR WHICH SENSOR HAS NOT BEEN SPECIFICALLY TESTED BY FIGARO.

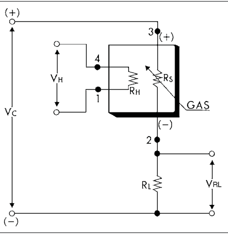
Figure A.1: Product information for Figaro TGS 2620 [18]. Published with permission from Sören Johansson at CN System AB on behalf of Figaro Engineering Inc.



Basic Measuring Circuit:

The sensor requires two voltage inputs: heater voltage (V_H) and circuit voltage (V_C). The heater voltage (V_H) is applied to the integrated heater in order to maintain the sensing element at a specific temperature which is optimal for sensing. Circuit voltage (V_C) is applied to allow measurement of voltage (V_{RL}) across a load resistor (R_L) which is connected in series with the sensor.

A common power supply circuit can be used for both V_C and V_H to fulfill the sensor's electrical requirements. The value of the load resistor (R_L) should be chosen to optimize the alarm threshold value, keeping power consumption (P_S) of the semiconductor below a limit of 15mW. Power consumption (P_S) will be highest when the value of Rs is equal to R_L on exposure to gas.



Specifications:

Model number			TGS 2620-C00	
Sensing element type			D1	
Standard package			TO-5 metal can	
Target gases			Alcohol, Solvent vapors	
Typical detection range			50 ~ 5,000 ppm	
Standard circuit conditions	Heater Voltage	V_H	5.0±0.2V DC/AC	
	Circuit voltage	V_C	5.0±0.2V DC/AC	$P_S \leq$ 15mW
	Load resistance	R_L	Variable	0.45kΩ min.
Electrical characteristics under standard test conditions	Heater resistance	R_H	83Ω at room temp. (typical)	
	Heater current	I_H	42 ± 4mA	
	Heater power consumption	P_H	approx. 210mW	
	Sensor resistance	Rs	1 ~ 5 kΩ in 300ppm ethanol	
	Sensitivity (change ratio of Rs)		0.3 ~ 0.5	$\frac{\text{Rs (300ppm)}}{\text{Rs (50ppm)}}$
Standard test conditions	Test gas conditions		Ethanol vapor in air at 20±2˚C, 65±5%RH	
	Circuit conditions		V_C = 5.0±0.01V DC V_H = 5.0±0.05V DC	
	Conditioning period before test		7 days	

The value of power dissipation (P_S) can be calculated by utilizing the following formula:

$$P_S = \frac{(V_C - V_{RL})^2}{R_S}$$

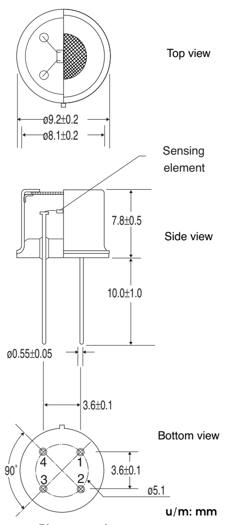
Sensor resistance (Rs) is calculated with a measured value of V_{RL} by using the following formula:

$$R_S = \frac{V_C - V_{RL}}{V_{RL}} \times R_L$$

For information on warranty, please refer to Standard Terms and Conditions of Sale of Figaro USA Inc.

REV: 01/05

Structure and Dimensions:



Pin connection:
1: Heater
2: Sensor electrode (-)
3: Sensor electrode (+)
4: Heater

FIGARO USA, INC.
121 S. Wilke Rd. Suite 300
Arlington Heights, Illinois 60005
Phone: (847)-832-1701
Fax: (847)-832-1705
email: figarousa@figarosensor.com

Figure A.2: Product information for Figaro TGS 2620 [18]. Published with permission from Sören Johansson at CN System AB on behalf of Figaro Engineering Inc.

Appendix B

Sensor Circuitry

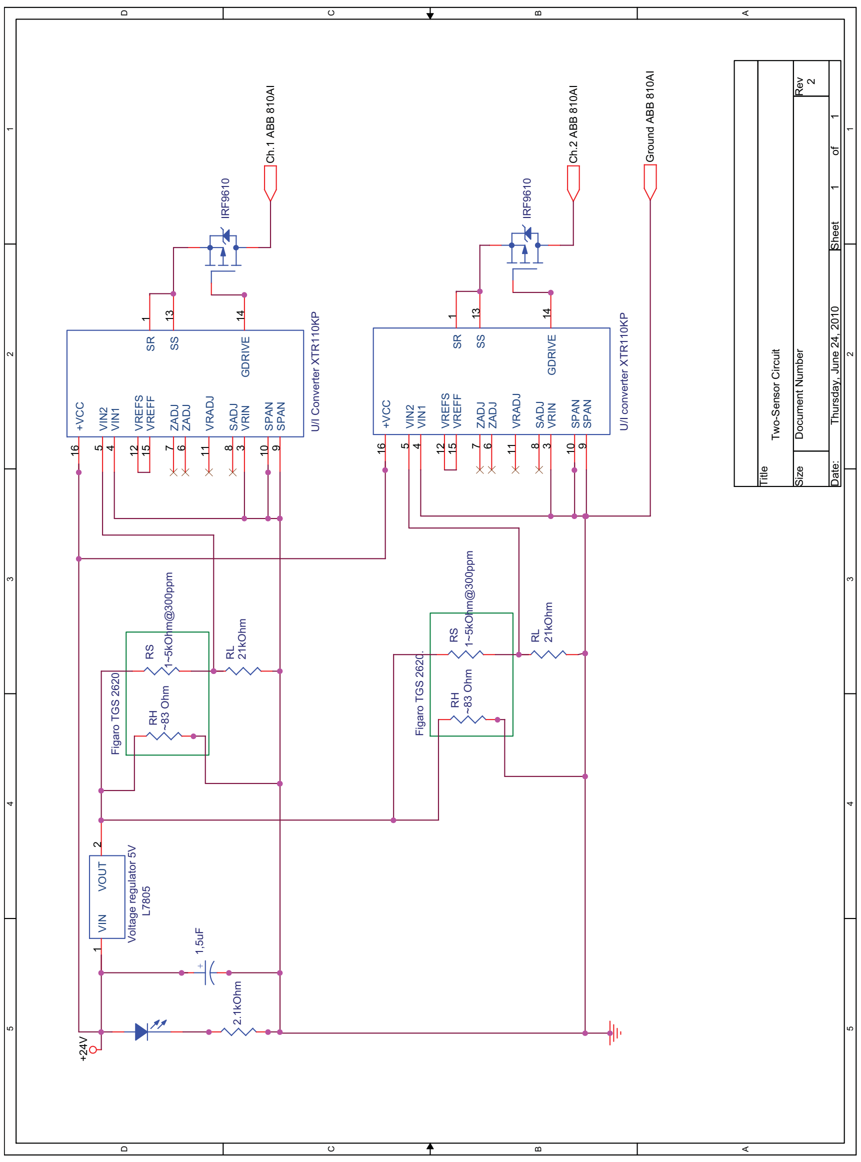


Figure B.1: Circuit