

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

**OPTIMAL SAMPLING STRATEGIES USING SPATIALLY
AVERAGED ADVECTION-DIFFUSION PARAMETER
ESTIMATION**

By

Andrew Philip White

THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Mechanical Engineering

2008

ABSTRACT

OPTIMAL SAMPLING STRATEGIES USING SPATIALLY AVERAGED ADVECTION-DIFFUSION PARAMETER ESTIMATION

By

Andrew Philip White

In this paper, optimal sensing strategies are presented which use spatial averaging of noisy measurements of an advection-diffusion process for parameter estimation to trace the maximum of a field of interest. Each strategy utilizes nonlinear least-square (NLS) optimization to estimate the parameters of a locally modeled advection-diffusion process. In the first strategy, nonholonomic sensing agents are steered in the direction that maximizes an objective function. Using ideas from the first strategy, a second strategy is designed to find the maximum of a complicated advection-diffusion process in a non-convex surveillance region. Each time the NLS optimization is run, the target position for the nonholonomic sensing agents is updated with the estimated maximum concentration position of the locally modeled advection-diffusion process. To ensure the nonholonomic sensing agents do not collide with boundaries in the non-convex region, an object avoidance algorithm is utilized.

ACKNOWLEDGMENTS

I would like to thank my advisors Dr. Jongeun Choi and Dr. L. Guy Raguin, for their guidance on this project. I would also like to thank Dr. Subir Biswas and Dr. Seungik Baek for being members on my committee.

TABLE OF CONTENTS

LIST OF TABLES	v
LIST OF FIGURES	vi
1 Introduction	1
2 Problem Statement	4
3 Averaged Process	6
4 Vehicle Formation Control for a Control Volume	9
5 Feedback Linearization of Nonholonomic Agents	13
5.1 Single-Integrator Compensated System	14
6 Control Strategies	16
6.1 Control strategy for Problem 1	16
6.2 Control Strategy for Problem 2	18
7 Simulation Results	24
7.1 Simulation Results from Problem 1	24
7.2 Simulation Results from Problem 2	26
8 Future Work	32
8.1 Numerical closed-form solution of the Local Process	32
8.2 Function Approximation	34
9 Conclusion	37
APPENDICES	38
BIBLIOGRAPHY	41

LIST OF TABLES

7.1	Parameters in the simulation	25
7.2	Estimated Parameters from each simulation	25

LIST OF FIGURES

Images in this thesis are presented in color.

4.1	Possible distribution of sensing agents in a control volume.	10
5.1	Schematic of the i^{th} Nonholonomic Agent	14
6.1	Flow Chart of the Optimal Sampling Strategy	17
6.2	Sampling Strategy Timeline	19
6.3	Contaminant Source with complex flow field	19
6.4	Sample grid of a non-convex region	22
6.5	Path found from (2,1) to (1,4) by A* search algorithm	22
6.6	Search graph to get from node (2,1) to node (1,4).	23
7.1	Nonholonomic Agents starting at $x = 20$ and $y = 20$	26
7.2	Nonholonomic Agents starting at $x = 20$ and $y = 120$	27
7.3	Nonholonomic Agents starting at $x = 120$ and $y = 20$	27
7.4	Comparison of the true plume and estimated plume after 150 iterations.	29
7.5	Comparison of performance with and without “anemotaxis”.	30
7.6	Sensing Agents attempting to trace a more complex concentration field.	31
8.1	Local and Global Coordinate Relationship	36

CHAPTER 1

Introduction

Many physical transport phenomena exist that are governed by advection-diffusion processes. Due to the harmful nature of some of these processes, researchers have been developing methods to track the source of such advection-diffusion processes [10, 5, 29, 14, 13, 8, 23, 28, 3]. The most common approach to this problem is biologically inspired “chemotaxis” [1, 5], in which a mobile sensing vehicle is driven according to a local gradient of the chemical plume concentration. However, with this approach, the convergence rate can be slow and the mobile robot may get stuck in the local maxima of chemical plume concentration. If strong ventilation is present, an appropriate technique is “anemotaxis” in which a robot moves in the direction against the wind for locating the odor source. For example, a robot equipped with gas sensors and anemometric sensors moves in the direction of upwind [14]. A switching mechanism between anemotaxis and chemotaxis according to the magnitude of measured odor concentration was designed in [13]. Before a sensing robot detects the plume concentration, popular biologically inspired exploratory movements of a robot include a biased random walk, a “run and tumble” flagellated bacteria like behavior, and “zigzag” and “spiral” trajectories.

A methodology of identifying the source term for an instantaneous pollutant point source from measurements in different locations was presented in [15]. The parameters (such as chemical mass release, diffusivity, source height, location of the source

and the time of the pollutant release) for the solution of an advection-diffusion model were estimated through nonlinear least-squares regression. This was possible since an analytical solution was obtained from a simple diffusion model. The advantage of parametric estimation is that it allows sensing robots to be steered in order to minimize the uncertainty in the parameter estimation of a diffusion model. The control law of sensing robots for source localization was derived in [23, 28] by minimizing the error of the approximated maximum likelihood parameter estimation. The work [3] also described mobile sensing robots that estimate the parameters of the advection-diffusion equation. A major drawback of this approach is that the analytical solution of a chemical plume process is not available for most practical cases. For instance a gas release in an airport with a complex wind field (from ventilation and turbulence effects), it is impossible to pre-decide an analytical solution and its parameters that intelligent robots can learn about. With a wind field, dispersions of chemical plumes are more significant than diffusions. Given all boundary conditions and wind fields, the evolution of a plume can be simulated by the Large-Eddy Simulation (LES) code such as HIGRAD [4, 26] at the cost of high computational power. In addition, recently emerging distributed control strategies over mobile sensing networks have not been considered in the control algorithms for chemical plume tracing developed in [23, 28, 3].

In recent years, the cooperative control of mobile sensor networks has gained much attention from control engineers and scientists. The cooperative network of vehicles that performs adaptive gradient climbing in a distributed environment was presented in [11, 17]. The network can adapt its configuration in response to the sensed environment in order to optimize its gradient climb.

Due to the complexity of realistic advection-diffusion processes that are seen in practice, it is clear that obtaining an analytical solution is not only unpractical but in many cases is also not possible. Realistic advection-diffusion processes can be non-smooth, turbulent, and complex. However, if spatial averaging is used, measurements taken inside a control volume can be used to estimate the parameters of a locally

averaged model. The locally averaged model can be smooth and simple with well defined parameters. These parameters can then be utilized to drive the robots to trace the field of interest that is undergoing a complex transport phenomena. Hence, the objective of this research is to synthesize a mobile sensor network that is capable of finding the source of contaminant.

This paper is organized as follows. Chapter 2 describes the problems that are addressed in this research and a brief description of the methods used to solve them. Chapter 3 introduces the concept of an averaged process. Chapter 4 discusses the formation control consensus algorithms used to ensure that a consistent control volume is maintained. Chapter 5 shows how the nonholonomic constraints of the sensing agents are handled. Chapter 6 details the control strategies used to solve problems 1 and 2. Chapter 7 shows the results that were obtained from each control strategy. Chapter 8 introduces additional concepts needed to proceed further with this research. Finally, in Chapter 9, conclusions are stated.

CHAPTER 2

Problem Statement

Problem 1: Consider an idealized advection-diffusion process in a convex surveillance region. Using a network of nonholonomic sensing agents, estimate the parameters of the idealized advection-diffusion process. Moreover, coordinate the mobile sensor network to improve the quality of the estimates.

Problem 2: Consider a realistic advection-diffusion process in a complicated non-convex surveillance region. Using a network of nonholonomic sensing agents, locate the maximum of the advection-diffusion process.

The first problem is solved by considering a control volume over which spatial averaging takes place. To maintain a moving control volume, the robots are controlled such that they travel in a formation [27]. The mobile sensing agents take concentration measurements inside the control volume. Nonlinear least-squares regression is used to estimate the parameters of the locally averaged process model. To determine the uncertainty in the estimated parameters, the Fisher Information Matrix is used to find the Cramer-Rao Lower Bound [16]. Then to improve the quality of the estimated parameters, the determinant of the Fisher Information Matrix is maximized (D-Optimality) [7]. By taking the partial derivative of the determinant of the Fisher Information Matrix with respect to χ , control inputs for the formation control algorithm are generated which drive the formation in the direction that maximizes the determinant of the Fisher Information Matrix.

The second problem is solved by using the same spatial averaging and control volume approach that the first problem utilizes. However, to solve the second problem, instead of using D-Optimality to drive the sensing agents, the initial position of the estimated diffusion impulse response is set as a target position. Due to the non-convex geometry under consideration, an obstacle avoidance algorithm is used to track a safe, collision-free path from the start position to the goal position. The A* graph search [20] is used to plan the collision-free path.

CHAPTER 3

Averaged Process

In this section, the averaged process that is used to model the local, spactially-averaged, advection-diffusion process is introduced. The transport of a solute of concentration $C(\chi, t)$ in a medium is governed by the advection-diffusion equation

$$\frac{\partial C}{\partial t} = -\nabla \cdot q, \quad (3.1)$$

where q is the flux of the solute and is decomposed into an advective and a diffusive term

$$q = C u - D_s \nabla C. \quad (3.2)$$

u is the velocity vector, and D_s the molecular diffusion coefficient of the solute in the medium.

Assuming incompressible flow ($\nabla \cdot u$), with spatially averaged velocity field V , the governing equation can be rewritten in its volume-averaged form [22]

$$\frac{\partial \bar{C}}{\partial t} + V \cdot \nabla \bar{C} = D \nabla^2 \bar{C}, \quad (3.3)$$

where D is the dispersion tensor that contains both advective and diffusion terms, as well as effects of the geometry, and may be time-dependent [2]. D is symmetric and usually contains cross-terms.

In three-dimensional space, $V = (V_x, V_y, V_z)^T$ and

$$D = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{bmatrix}, \quad (3.4)$$

therefore the governing equation for $\bar{C}(x, y, z, t)$ can be written as

$$\begin{aligned} \frac{\partial \bar{C}}{\partial t} + V_x \frac{\partial \bar{C}}{\partial x} + V_y \frac{\partial \bar{C}}{\partial y} + V_z \frac{\partial \bar{C}}{\partial z} \\ = D_{xx} \frac{\partial^2 \bar{C}}{\partial x^2} + D_{yy} \frac{\partial^2 \bar{C}}{\partial y^2} + D_{zz} \frac{\partial^2 \bar{C}}{\partial z^2} + 2D_{xy} \frac{\partial^2 \bar{C}}{\partial x \partial y} + 2D_{yz} \frac{\partial^2 \bar{C}}{\partial y \partial z} + 2D_{xz} \frac{\partial^2 \bar{C}}{\partial x \partial z}. \end{aligned} \quad (3.5)$$

In an infinite medium, the response to an impulse function, $\bar{C}(x, y, z, t_0) = C_0 \delta(x_0, y_0, z_0, t_0)$, located at an unknown position $\chi_0 = (x_0, y_0, z_0)$ and unknown time t_0 is then

$$\begin{aligned} \bar{C}(\chi, t) = & \frac{C_0}{\sqrt{(4\pi)^3 (t - t_0)^3 \det D}} \\ & \times \exp \left[-\frac{(\chi - \chi_0 - V(t - t_0))^T D^{-1} (\chi - \chi_0 - V(t - t_0))}{4(t - t_0)} \right]. \end{aligned} \quad (3.6)$$

At this point, the scope of consideration is reduced to the two-dimensional process in the (x, y) -plane. The simplified response to the impulse function, $C(x, y, t_0) = C_0 \delta(x_0, y_0, t_0)$, is

$$\begin{aligned} \bar{C}(x, y, t) = & \frac{C_0}{\sqrt{(4\pi)^3 (t - t_0)^3 (D_{xx} D_{yy} - D_{xy}^2)}} \\ & \times \exp \left\{ -\frac{1}{4(D_{xx} D_{yy} - D_{xy}^2) (t - t_0)} \right. \\ & \times [(\chi - \chi_0 - V(t - t_0))]^T \begin{bmatrix} D_{yy} & -D_{xy} \\ -D_{xy} & D_{xx} \end{bmatrix} [(\chi - \chi_0 - V(t - t_0))] \}. \end{aligned} \quad (3.7)$$

The goal is then to recover the following parameters: D_{xx} , D_{yy} , D_{xy} , C_0 , x_0 , y_0 , and t_0 , based on measurements of $C(x, y, t)$ at different locations and times. The sensing agents are assumed to be able to measure wind velocity, such that V_x and V_y are considered to be known.

The concentration function is rewritten as

$$\bar{C}(\mathcal{Q}, \theta), \quad (3.8)$$

where θ is the parameter defined by

$$\theta := \begin{bmatrix} D_{xx} & D_{yy} & D_{xy} & C_0 & x_0 & y_0 & t_0 \end{bmatrix}^T \in \mathbb{R}^7, \quad (3.9)$$

and the configuration \mathcal{Q} is

$$\mathcal{Q} := \begin{bmatrix} x & y & t \end{bmatrix}^T \in \mathbb{R}^3. \quad (3.10)$$

CHAPTER 4

Vehicle Formation Control for a Control Volume

To describe the information flow between each of the sensing agents some graph notation is now introduced. A directed graph consists of a pair $(\mathcal{N}, \mathcal{E})$ where \mathcal{N} is a set of nodes and \mathcal{E} is a set of ordered node pairs, also known as an edge (see [24]). For the formation control [27] used in this research, a graph known as a directed spanning tree is required. A directed spanning tree is formed by graph edges which connect all of the nodes of the graph. Examples of a directed spanning tree are illustrated in Fig. 4.1.

To maintain a control volume for the averaged process estimation, the sensing agents are arranged into a controlled formation. The reference point of the control volume is referred to as the virtual center (x^{vc}, y^{vc}) . The position of all sensing agents is determined with respect to the virtual center based on the shape, size, and the number of agents in the control volume.

Each agent has its own understanding of the location of the virtual center, which is defined as $q_i = [x_i^{vc} \ y_i^{vc} \ \theta_i^{vc}]^T$ for the i^{th} agent. The path the formation follows is defined as $q_d = [x_d^{vc} \ y_d^{vc} \ \theta_d^{vc}]^T$. According to [27], if the adjacency matrix $G^{vc} = [g_{ij}^{vc}]$ has a directed spanning tree, then $q_i \rightarrow q_d$ asymptotically using the consensus

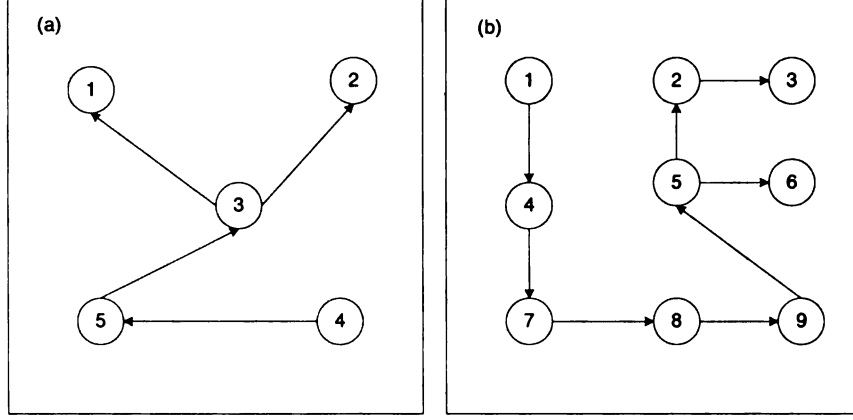


Figure 4.1. Possible distribution of sensing agents in a control volume.

algorithm

$$\dot{q}_i = \frac{\ell_i(\dot{q}_d - \gamma(q_i - q_d)) + \sum_{j=1}^n g_{ij}^{vc} [\dot{q}_j - \gamma(q_i - q_j)]}{\ell_i + \sum_{j=1}^n g_{ij}^{vc}}, \quad (4.1)$$

where $\ell_i = 1$ if the i^{th} agent is a leader and zero otherwise, $g_{ij}^{vc} = 1$ if agent j 's estimate is available to agent i and zero otherwise, and $\gamma > 0$. To demonstrate that $q_i \rightarrow q_d$ asymptotically, the error dynamics are now investigated. First the denominator of equation (4.1) is multiplied over to the left hand side of the expression

$$(\ell + \mathcal{G}^{vc}) \dot{q}_i = \ell [\dot{q}_d - \gamma(q_i - q_d)] + \sum_{j=1}^n g_{ij}^{vc} [\dot{q}_j - \gamma(q_i - q_j)] \quad (4.2)$$

where

$$\mathcal{G}^{vc} = \sum_{j=1}^n g_{ij}^{vc}.$$

Next, $\mathcal{G}^{vc} \dot{q}_d$ is subtracted from each side of the expression

$$(\ell + \mathcal{G}^{vc}) \dot{q}_i - \mathcal{G}^{vc} \dot{q}_d = \ell [\dot{q}_d - \gamma(q_i - q_d)] + \sum_{j=1}^n g_{ij}^{vc} [\dot{q}_j - \gamma(q_i - q_j)] - \mathcal{G}^{vc} \dot{q}_d. \quad (4.3)$$

After some more re-arranging, the expression becomes

$$(\ell + \mathcal{G}^{vc}) (\dot{q}_i - \dot{q}_d) = -\ell \gamma (q_i - q_d) + \sum_{j=1}^n g_{ij}^{vc} [(\dot{q}_j - \dot{q}_d) - \gamma((q_i - q_d) - (q_j - q_d))]. \quad (4.4)$$

Substituting $\tilde{q}_i = q_i - q_d$ into the expression, the error dynamics are

$$(\ell + \mathcal{G}^{vc}) \dot{\tilde{q}}_i = -\ell \gamma \tilde{q}_i + \sum_{j=1}^n g_{ij}^{vc} [\dot{\tilde{q}}_j - \gamma(\tilde{q}_i - \tilde{q}_j)], \quad (4.5)$$

which can be arranged into matrix form

$$\dot{\tilde{q}} = B\dot{\tilde{q}} + A\tilde{q} \quad (4.6)$$

where B and A are found to be

$$B = \frac{1}{(\ell + \mathcal{G}^{vc})} \begin{bmatrix} g_{11}^{vc} & g_{12}^{vc} & \cdots & g_{1n}^{vc} \\ g_{21}^{vc} & g_{22}^{vc} & \cdots & g_{2n}^{vc} \\ \vdots & & \ddots & \vdots \\ g_{n1}^{vc} & \cdots & \cdots & g_{nn}^{vc} \end{bmatrix},$$

$$A = \frac{1}{(\ell + \mathcal{G}^{vc})} \begin{bmatrix} -\gamma(\ell + \mathcal{G}^{vc}) & \gamma g_{12}^{vc} & \cdots & \gamma g_{1n}^{vc} \\ \gamma g_{21}^{vc} & -\gamma(\ell + \mathcal{G}^{vc}) & \cdots & \gamma g_{2n}^{vc} \\ \vdots & & \ddots & \vdots \\ \gamma g_{n1}^{vc} & \cdots & \cdots & -\gamma(\ell + \mathcal{G}^{vc}) \end{bmatrix}.$$

After some matrix operations, it can be shown that

$$\dot{\tilde{q}} = F\tilde{q}$$

with $F = (I - B)^{-1}A$. For $\gamma > 0$ [27], the eigenvalues of F have negative real parts, which shows that $\tilde{q} \rightarrow 0$, implying that $q_i \rightarrow q_d$ asymptotically.

It is assumed that the sensing agents have integrator dynamics such that

$$\dot{r}_i = u_i, \quad i = 1, \dots, n \quad (4.7)$$

where $r_i = [x_i, y_i]^T$ is the position and $u_i = [u_{x_i}, u_{y_i}]^T$ is the control input to the i^{th} vehicle. The agents are controlled by the following extended consensus algorithm as shown in [27]:

$$u_i = \dot{r}_i^d - \alpha_i(r_i - r_i^d) - \sum_{j=1}^n g_{ij} k_{ij} [(r_i - r_i^d) - (r_j - r_j^d)] \quad (4.8)$$

where $\alpha_i > 0$, $k_{ij} > 0$, $g_{ij} = 1$ if information flows from vehicle j to vehicle i and 0 otherwise, and $r_i^d = [x_i^d, y_i^d]^T$ with

$$\begin{bmatrix} x_i^d \\ y_i^d \end{bmatrix} = \begin{bmatrix} x_i^{vc} \\ y_i^{vc} \end{bmatrix} + \begin{bmatrix} \cos(\theta_i^{vc}) & -\sin(\theta_i^{vc}) \\ \sin(\theta_i^{vc}) & \cos(\theta_i^{vc}) \end{bmatrix} \begin{bmatrix} x_{iF} \\ y_{iF} \end{bmatrix}$$

where $[x_{iF}, y_{iF}]^T$ is the vector that gives the i^{th} vehicle's position in the formation with respect to the virtual center. The extended consensus algorithm can be arranged into the following matrix form.

$$\tilde{u} = J\tilde{r} \tag{4.9}$$

where $\tilde{r}_i := r_i - r_i^d$, $\tilde{u} := u - r_i^d$ and

$$J = \begin{bmatrix} -\alpha_1 + \sum_{j=1}^n g_{ij}k_{ij} & g_{12}k_{12} & \cdots & g_{1n}k_{1n} \\ g_{21}k_{21} & -\alpha_2 + \sum_{j=1}^n g_{ij}k_{ij} & \cdots & g_{2n}k_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1}k_{n1} & \cdots & \cdots & -\alpha_n + \sum_{j=1}^n g_{ij}k_{ij} \end{bmatrix}.$$

It is important to notice that J is diagonally dominant. From the Gershgorin disc theorem [12], it is easy to see that all of the eigenvalues of J have negative real parts.

This guarantees that $\tilde{r} \rightarrow 0$ exponentially, which shows that $r_i \rightarrow r_i^d, \forall i$.

CHAPTER 5

Feedback Linearization of Nonholonomic Agents

The sensing agents to be used in the implementation of this algorithm have nonholonomic dynamics. Dynamics that are said to be “nonholonomic” have constraints which are not integrable. This essentially means that knowing an ending configuration Ψ_2 does not give information about the starting configuration Ψ_1 . The nonholonomic dynamics simply constrain how the sensing agent can move from any starting configuration Ψ_1 to any ending configuration Ψ_2 . The kinematic equations that govern the dynamics of the i^{th} nonholonomic agent are

$$\begin{aligned}\dot{x}_i &= v_i \cos \theta_i, \\ \dot{y}_i &= v_i \sin \theta_i, \\ \dot{\theta}_i &= \omega_i,\end{aligned}\tag{5.1}$$

where (x_i, y_i) and θ_i give the position and orientation of the i^{th} nonholonomic agent and the inputs v_i and ω_i are, respectively, the forward and angular velocity of the vehicle (Fig. 5.1). There are many methods to feedback linearize equation (5.1) [21]. The controller discussed in this section will be used for trajectory tracking of a reference sensor position. Posture stabilization, a more difficult problem used in parking maneuvers, is not considered in this paper; since sensor position control is

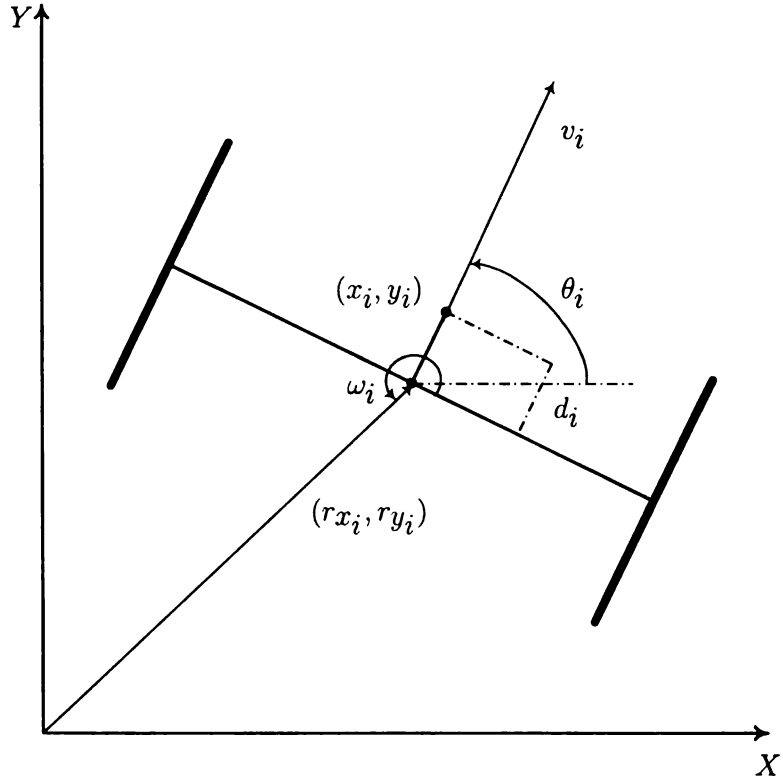


Figure 5.1. Schematic of the i^{th} Nonholonomic Agent

sufficient for the purposes of this research. Posture control can be found in [21].

5.1 Single-Integrator Compensated System

To achieve single-integrator dynamics, the coordinate transformation of

$$\begin{aligned} x_i &= rx_i + d_i \cos \theta_i, \\ y_i &= ry_i + d_i \sin \theta_i, \end{aligned} \tag{5.2}$$

is applied where (x_i, y_i) denotes the position of the i^{th} sensor and d_i represents the distance from the wheel axis to the sensor (Fig. 5.1). This moves the point of interest from (rx_i, ry_i) off the wheel axis by a distance of d_i to (x_i, y_i) . Taking the derivative of equation (5.2) and entering the values for (\dot{rx}_i, \dot{ry}_i) from equation (5.1) produces

the following equations:

$$\begin{aligned} \dot{x}_i &= v_i \cos \theta_i - d_i \sin(\theta_i) \omega_i, \\ \dot{y}_i &= v_i \sin \theta_i + d_i \cos(\theta_i) \omega_i. \end{aligned} \quad (5.3)$$

The output of i^{th} nonholonomic agent is defined as $\eta_i = (x_i, y_i)^T$. Taking the derivative of the output produces

$$\dot{\eta}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} \cos \theta_i & -d_i \sin \theta_i \\ \sin \theta_i & d_i \cos \theta_i \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}, \quad (5.4)$$

which shows that both v_i and ω_i are present in $\dot{\eta}_i$, meaning they can be used to control the output of the i^{th} nonholonomic agent. Thus, the linearizing feedback control can be found by letting $\eta_i = (u_{x_i}, u_{y_i})^T$ such that

$$\begin{bmatrix} u_{x_i} \\ u_{y_i} \end{bmatrix} = \begin{bmatrix} \cos \theta_i & -d_i \sin \theta_i \\ \sin \theta_i & d_i \cos \theta_i \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}. \quad (5.5)$$

By taking the inverse of the matrix above the inputs v_i and ω_i to the i^{th} nonholonomic agent are found to be

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} \cos \theta_i & \sin \theta_i \\ -\frac{1}{d_i} \sin \theta_i & \frac{1}{d_i} \cos \theta_i \end{bmatrix} \begin{bmatrix} u_{x_i} \\ u_{y_i} \end{bmatrix}. \quad (5.6)$$

By entering the values of v_i and ω_i found in equation (5.6) into equation (5.3), it is easy to see that

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} u_{x_i} \\ u_{y_i} \end{bmatrix}. \quad (5.7)$$

It should be noted that a singularity occurs when $d_i = 0$. If the position of interest is (r_{x_i}, r_{y_i}) , then double-integrator dynamics are necessary [25]. However, in this case, only control of the sensor position is necessary. Thus this will be sufficient since the sensor can be placed at an arbitrary distance d_i from the wheel axis as shown in Fig. 5.1.

CHAPTER 6

Control Strategies

6.1 Control strategy for Problem 1

In this section, a navigation strategy that coordinates the mobile sensor network to improve the quality of the parameter estimates is provided. Let T_s be the sensing sampling time. Every p^{th} sampling time, the leader vehicle will collect the spatially distributed measurements from the follower vehicles to update the parameter estimate $\hat{\theta}$. The estimate of the parameter is updated via the nonlinear least-squares optimization (for example, by the Levenberg-Marquardt method)

$$\hat{\theta}(t_k) = h(x(t_k), \hat{\theta}(t_k - T_s p)). \quad (6.1)$$

It is then natural to coordinate the mobile vehicles $x(t)$ for $t \in (t_k, t_k + T_s p]$ in order to maximize the determinant of Σ in equation (6.4) evaluated at the current parameter estimate and minimize the covariance of the estimated parameter $\hat{\theta}$. A flow chart of the optimal sampling strategy is given in Fig. 6.1. We introduce the Fisher Information Matrix (\mathcal{I}), given by [16]

$$\mathcal{I} := \frac{\Sigma(\theta^*)}{\sigma^2}, \text{ with } \Sigma := \sum_{i=1}^n C'(\mathcal{Q}_i, \theta^*) C'(\mathcal{Q}_i, \theta^*)^T \quad (6.2)$$

where $C'(\mathcal{Q}_i, \theta) := \left[\frac{\partial}{\partial \theta_j} C(\mathcal{Q}_i, \theta) \right]_j$. Since the true parameter θ^* is unknown, the estimated parameter $\hat{\theta}$ is used to compute the Fisher Information Matrix in equation

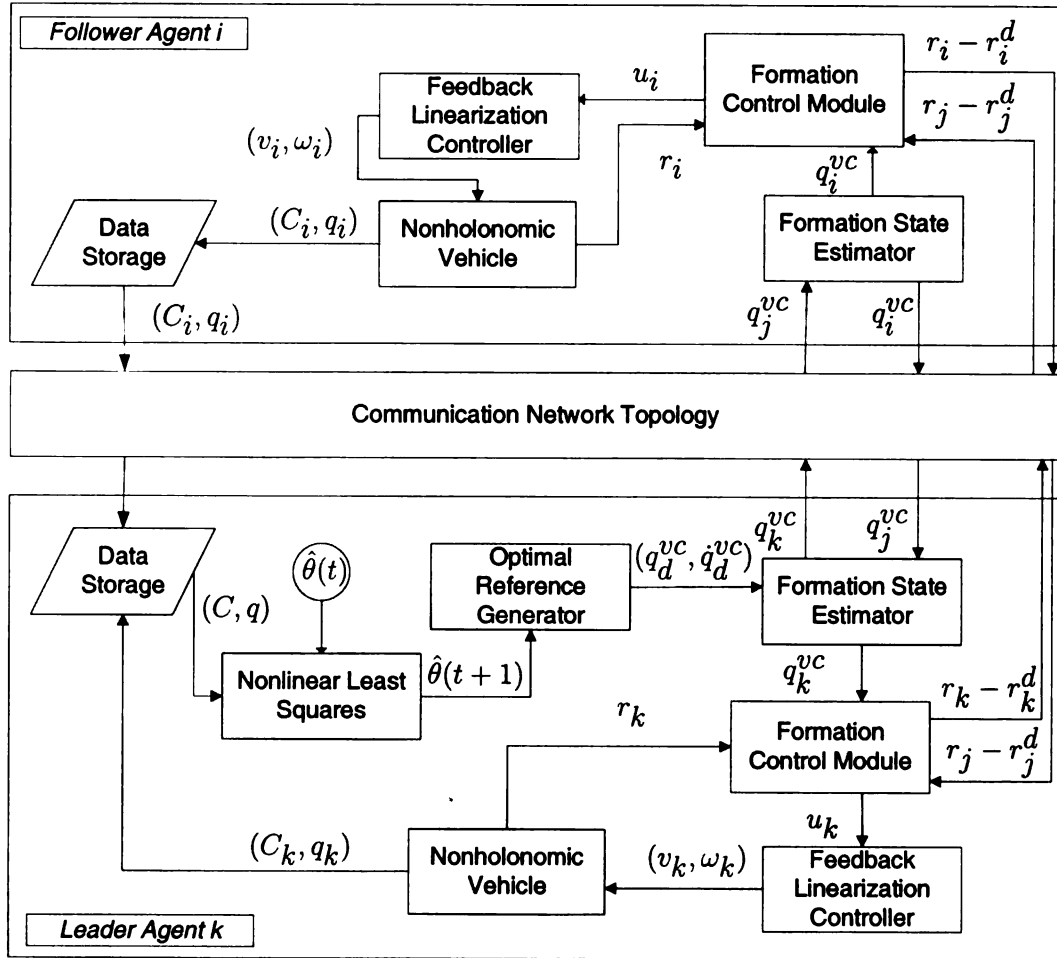


Figure 6.1. Flow Chart of the Optimal Sampling Strategy

(6.2) instead. The Cramer-Rao theorem states that the covariance matrix of any unbiased estimator $\hat{\theta}$ is lower bounded by the Cramer-Rao Lower Bound (CRLB), which is the inverse of the Fisher Information Matrix [7].

$$E \left\{ (\hat{\theta} - \theta^*)(\hat{\theta} - \theta^*)^T \right\} \geq \mathcal{I}^{-1}(\theta^*) \quad (6.3)$$

In order to improve the quality of the estimated parameter $\hat{\theta}$, the determinant of the Fisher Information Matrix will be maximized (D-optimality [7]).

$$\mathcal{J} = \det \mathcal{I} = \frac{1}{\sigma^2} \det \Sigma \quad (6.4)$$

The objective function \mathcal{J} is maximized by steering the sensing vehicles so that they climb the gradient of the objective function with respect to χ , which is given by

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \chi} &= \sum_{ij} \left(\frac{\partial \mathcal{J}(\Sigma)}{\partial \Sigma} \right) \left(\frac{\partial \Sigma}{\partial \chi} \right)_k \\ &= \sum_{ij} \left(\det(\Sigma)(\Sigma)^{-T} \right) \left(\frac{\partial \Sigma(\chi)}{\partial \chi} \right)_k. \end{aligned} \quad (6.5)$$

The vehicle control is defined as

$$\dot{q}_d^{vc} = \begin{cases} v & \text{if } \|v\| < v_{sat}; \\ \frac{v}{\|v\|} v_{sat} & \text{if } \|v\| \geq v_{sat}; \end{cases} \quad (6.6)$$

where $v = \varepsilon \frac{\partial \mathcal{J}}{\partial \chi}$, v_{sat} is the saturation velocity of the vehicle and $\varepsilon > 0$ is a gain. Also, to ensure that the sensing agents stay inside of the surveillance region Ω , a projection algorithm is used, which updates the current position if it belongs to Ω and keeps the previous position if it does not belong to Ω [18].

6.2 Control Strategy for Problem 2

The approach to solve this problem combines (A) the nonlinear least-squares estimation approach, which was used to solve the simpler more ideal problem posed in the previous section, with (B) the wind tracking strategy known as “anemotaxis”

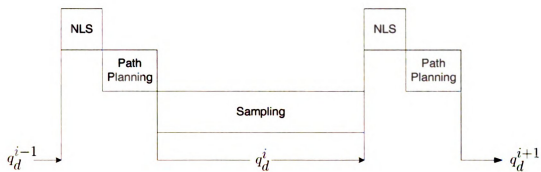


Figure 6.2. Sampling Strategy Timeline

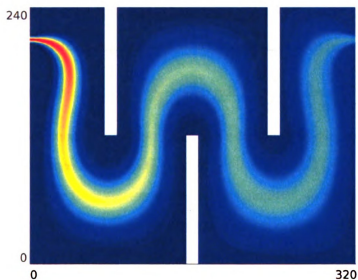


Figure 6.3. Contaminant Source with complex flow field

[14]. Anemotaxis, as stated in the introduction, is a technique used to send a robot in the direction against the wind for locating an odor source. A switching mechanism is used to determine whether the sensing agents will go to the goal position given by the estimated parameters or follow the wind velocity. The condition for choosing which strategy to use is based on the objective function (6.4). Only parameter estimates that produce a \mathcal{J} greater than \mathcal{J}_{min} are trusted. If a value of \mathcal{J} is computed to be lower than \mathcal{J}_{min} , then “anemotaxis” is implemented.

Since the complicated non-convex region is considered (Fig. 6.3), an obstacle avoidance strategy is necessary to ensure that the sensing agents do not bump into the walls. In this work, it is assumed that the robots have knowledge of the nonconvex geometry. Each of the walls (as in Fig. 6.3) are placed inside a no-fly zone of which the leader robot has full knowledge.

The choice of the physical process that models the local behavior of the plume is important. Many options exist from which to choose. In the previous section, an advection-diffusion impulse response is used. For this problem, the following three options were considered: diffusion impulse response Eq. (3.6) (with $V = 0$), advection-diffusion impulse response Eq. (3.6), and continuous source isotropic diffusion [19]

$$C(\chi, t, \chi', q_0, t_0) = \frac{q_0(t - t_0)}{2\pi K d} \cdot \text{erfc}\left(\frac{d}{2\sqrt{K(t - t_0)}}\right) \quad (6.7)$$

where q_0 is the continuous source rate, K is the isotropic diffusion coefficient, and d is the Euclidean distance from χ to the source χ' , $d = \|\chi - \chi'\|_2$.

After each nonlinear least-square estimation is complete, an updated estimate of the target position is obtained. This new target position is then given to the obstacle avoidance algorithm, which creates a safe, collision free path from the starting position to the target position. Before a path can be found, the surveillance region must be partitioned into a grid as illustrated by the sample region in Fig. 6.4. A node is placed at the center of each grid box. Also, the no-fly zone and boundary are indicated on the grid by giving any node in a grid box a terrain cost of ∞ , such that these nodes will not be considered during the path search. To find the path from the start node

to the target node, a search algorithm must be used to create a reasonable path. The search algorithm we used in this paper is the A* algorithm, which was introduced by Nilsson [20] and uses heuristic information to improve performance when compared to Dijkstra's algorithm [6]. The A* graph search method creates a hierarchical tree search graph G (as displayed by the sample search in Fig. 6.6) to test many paths to find one with the lowest cost. The search graph G begins with the start node on the top of the tree. The possible nodes that can be travelled to from the start node are installed in G as children of the start node. Then, based on the heuristic cost [9]

$$h = \sqrt{2}a \times \min(|\delta_x|, |\delta_y|) + a \times ||\delta_x| - |\delta_y||, \quad (6.8)$$

where a is the axial distance between each path node, δ_x is the distance between the current node and the target node in the x coordinate, and δ_y is the distance between the current node the goal node in the y coordinate, the next node to expand is chosen. This process is repeated until a path is found. The A* graph search is summarized below.

1. Create a search graph, G , which consists of only the start node, s . Place s on a list called OPEN.
2. Create a list called CLOSED which initially has no nodes.
3. Start the search loop. If OPEN is empty, then exit with failure.
4. Select the first node on OPEN and name it n . Remove n from OPEN and place it on CLOSED.
5. If n is the goal node, then exit the loop with the path obtained by tracing the pointers from n to s .
6. Expand node n to find all possible children nodes which are not parent nodes of n . Place these nodes in a set called M .
7. Create a pointer from the nodes in M , which are not already in OPEN or CLOSED, to n . Add these members of M to OPEN.

8. Re-arrange OPEN according to the lowest heuristic cost.

9. Repeat the loop.

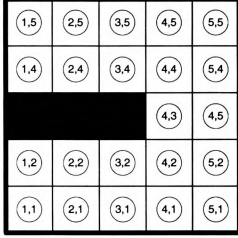


Figure 6.4. Sample grid of a non-convex region

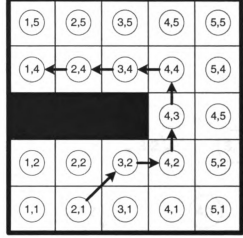


Figure 6.5. Path found from (2,1) to (1,4) by A* search algorithm

When this algorithm is carried out to find a path from a start node of (2,1) on the sample grid in Fig. 6.4 to a goal node of (1,4), the search graph is expanded as shown in Fig. 6.6 to find the path traced out in Fig. 6.5.

Once a path, $p = [p_0, p_1, \dots, p_n]$, is found the reference trajectory for the formation control, $q_d^i(t)$, is computed by the leader vehicle as shown in Fig. 6.2. The reference trajectory is created by filtering a step function from $p_0 = [x_0, y_0]$ to $p_1 = [x_1, y_1]$ through a low pass filter

$$H(s) = \frac{1}{\tau s + 1} \quad (6.9)$$

such that

$$q_d^i(t) = (p_1 - p_0) \left(1 - \exp \left(-\frac{t - t_0}{\tau} \right) \right) + p_0 \quad (6.10)$$

and

$$\dot{q}_d^i(t) = \left(\frac{p_1 - p_0}{\tau} \right) \exp \left(-\frac{t - t_0}{\tau} \right). \quad (6.11)$$

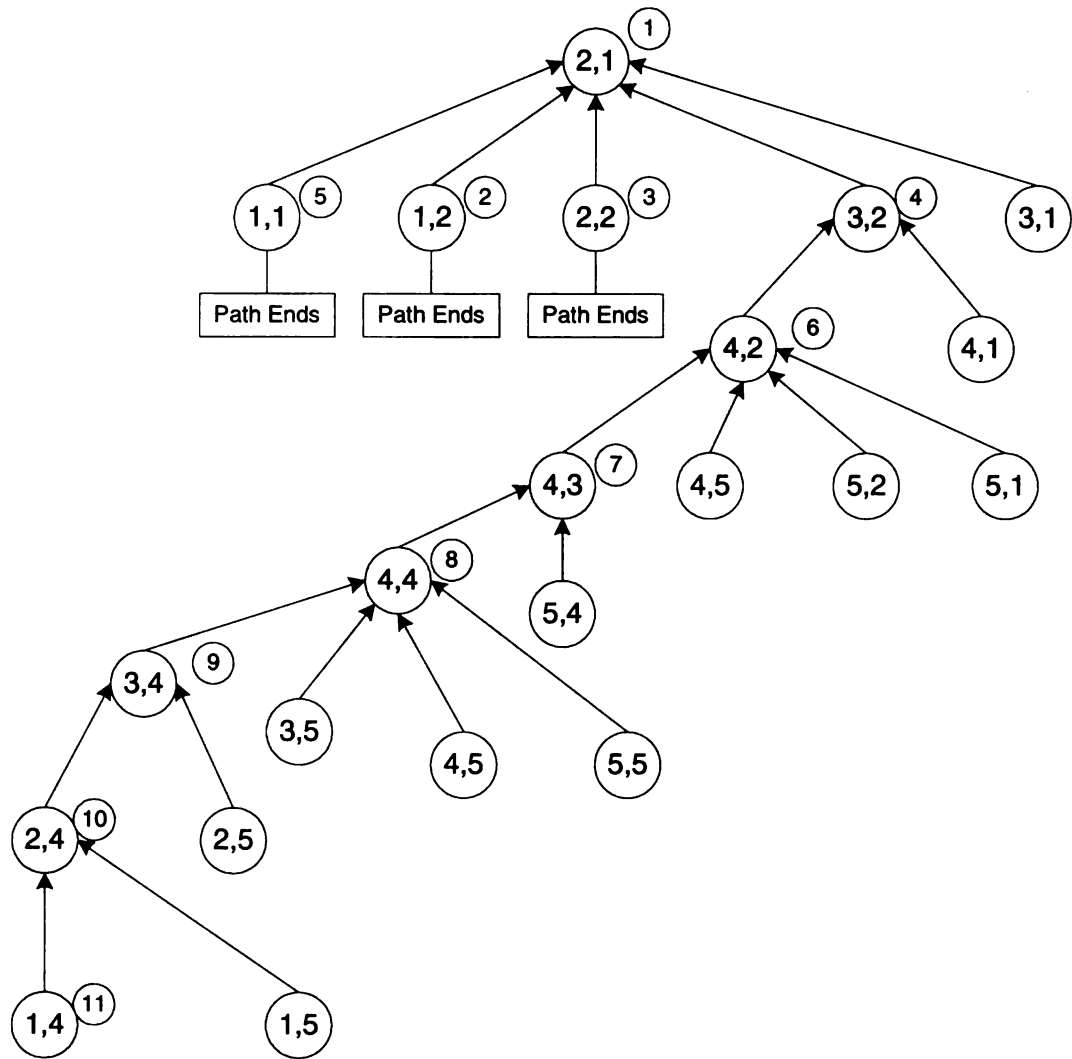


Figure 6.6. Search graph to get from node (2,1) to node (1,4).

CHAPTER 7

Simulation Results

7.1 Simulation Results from Problem 1

The sampling strategy detailed Section 6.1 is tested using a Matlab simulation code. To simulate a physical process, a true set of parameters is defined. Table 7.1 displays the true parameters and the assumed apriori values used in the simulations. The true parameters are used to create the field of interest that the sensing agents take noisy measurements from. The concentration measurements are then used with the apriori parameter values to estimate the true parameters using nonlinear least-squares optimization. The apriori knowledge for each of the parameters, with the exception of the initial position, was taken to be twenty percent away from the true value. The initial position of the plume was given a value far from the true value since the no apriori knowledge about the position of the plume is assumed except that it is inside of the surveillance region. In Table 7.2, the estimated values for each of the simulation trials are displayed.

Figs. 7.1 - 7.3 display the sensing agents starting position in grey and their final position in black. The trajectory that each of the agents followed during the simulation is plotted in magenta and the reference trajectory is displayed in red. Figs. 7.1 - 7.3 also show that the sensing agents can be inserted into the field at any location without affecting their performance. However, Table 7.2 does show that

Table 7.1. Parameters in the simulation

Parameters	True Values	apriori Values
$D_{xx} (\frac{m^2}{s})$	0.07	0.084
$D_{yy} (\frac{m^2}{s})$	0.07	0.084
$D_{xy} (\frac{m^2}{s})$	-0.01	-0.012
$C_0 (\frac{kg}{m^3})$	5.0×10^6	6.0×10^6
$(x_0, y_0) (m)$	(80, 80)	(1, 1)
$t_0 (s)$	30.0	36.0

Table 7.2. Estimated Parameters from each simulation

Parameters	Estimated Values with starting position (x_s, y_s)		
	(20, 20)	(120, 20)	(20, 120)
$D_{xx} (\frac{m^2}{s})$	0.0699	0.0703	0.0702
$D_{yy} (\frac{m^2}{s})$	0.0697	0.0703	0.0702
$D_{xy} (\frac{m^2}{s})$	-0.0099	-0.0101	-0.0101
$C_0 (\frac{kg}{m^3})$	5.0043×10^6	5.0149×10^6	5.0124×10^6
$(x_0, y_0) (m)$	(79.93, 79.89)	(79.93, 80.08)	(80.03, 79.95)
$t_0 (s)$	24.2717	30.2394	29.4507

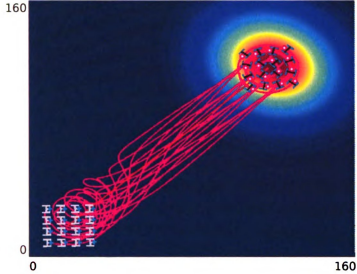


Figure 7.1. Nonholonomic Agents starting at $x = 20$ and $y = 20$

the parameters estimated by the robots does change slightly depending on the initial position of the agents. The estimated parameter which shows the highest amount of variation is the initial time, t_0 . Fig. 7.4 shows the comparison of the plume computed using the true values and the plume computed using estimated values from the simulation starting at $(x_s, y_s) = (20, 20)$.

7.2 Simulation Results from Problem 2

The sampling strategy that was detailed in Section 6.2 was tested using a Matlab simulation code. The concentration field used in these simulations was created using Comsol. The concentration field was created to model a constant contaminant release inside of a closed area. In Fig. 7.5 - 7.6, the nonholonomic agents are represented by a yellow plus sign at (x_s, y_s) and by a white dot at their final location.

Without utilizing the switching algorithm based on the objective function, the best results obtained are displayed in Fig. 7.5. The sensing agents clearly have not

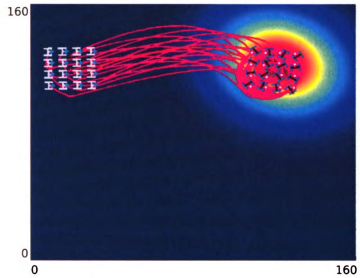


Figure 7.2. Nonholonomic Agents starting at $x = 20$ and $y = 120$

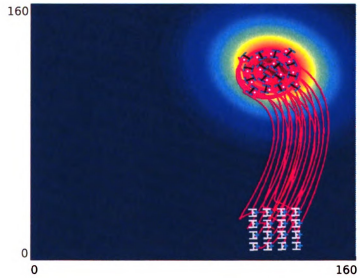
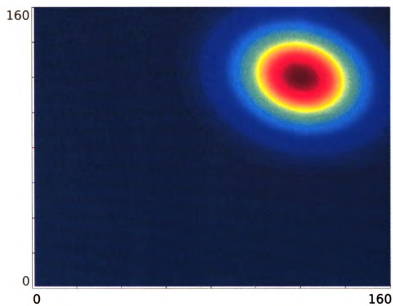


Figure 7.3. Nonholonomic Agents starting at $x = 120$ and $y = 20$

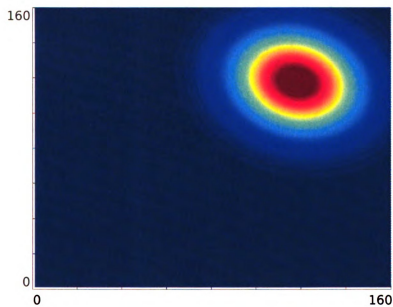
found the maximum position in the field. This has happened because the physical process used to model the local plume does not match the local plume. To keep the sensing agents moving, the switching algorithm based on the objective function 6.4 was used. The simulation results obtained when using the switching algorithm are displayed in Fig. 7.5. In this simulation, the sensing agents do find the maximum of the field of interest. However, it is noticeable that the parameter estimation algorithm does not get utilized until the sensing agents are almost at the maximum. It is also clear that with this field can be tracked with only the use of “anemotaxis”.

To more completely test the strategy developed in Section 6.2, another concentration field was created using Comsol. In this concentration field, a contaminate-free inlet was added below the contaminated inlet. The contaminate-free inlet is denoted in Fig. 7.6 by the letter A, and the outlet is denoted by the letter B. The resulting concentration field cannot be traced with only the use of “anemotaxis”. However, as Fig. 7.6 illustrates, it appears as though the strategy developed cannot trace it either. This, again, is due to the mismatch between the modelled plume and the actual local plume.

The physical process that most closely resembles the local plume behavior (of those considered) is the diffusion impulse response. The wind in the advection-diffusion impulse response moves the maximum concentration area of the plume away from the estimated initial point (x_0, y_0) . Since the robots are moving towards the estimated initial point of the local plume, it is desirable that it does not move. Also, when investigating the continuous-source, isotropic-diffusion model, it was noticed that the source location has a very steep gradient that does not model the local behavior well at all. However, clearly, the diffusion impulse response does not model the local plume behavior well enough. For this reason, a numerical solution will be developed to better model the local plume behavior. The framework for creating such a numerical solution is laid out in the next section.

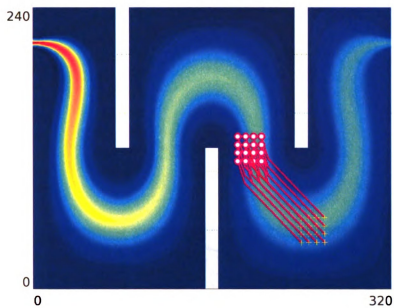


(a) True Plume

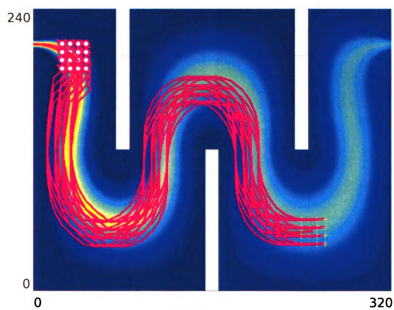


(b) Estimated Plume with starting position $(x_s, y_s) = (20, 20)$

Figure 7.4. Comparison of the true plume and estimated plume after 150 iterations.



(a) Parameter Estimation Strategy Only



(b) Switching between Parameter Estimation and “anemotaxis”

Figure 7.5. Comparison of performance with and without “anemotaxis”.

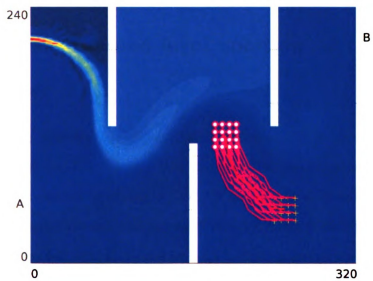


Figure 7.6. Sensing Agents attempting to trace a more complex concentration field.

CHAPTER 8

Future Work

8.1 Numerical closed-form solution of the Local Process

When considering different advection-diffusion processes, it is necessary to have a closed-form solution, like the one that provides the concentration field of the advection-diffusion process in equation (3.7). If an analytical closed-form solution for the local process of interest does not exist then a numerical closed-form solution must be found. The following steps can be done to find such a solution.

Before the numerical analysis can be done, a dimensional analysis is carried out first so that non-dimensional parameters can be found. The steps used are outlined below.

1. First list all of the variables that are involved in the problem

$$C = f(x, y, d, q_0, D, V, \nu) \quad (8.1)$$

where C is the concentration, (x, y) is the position of the concentration C , d is the inlet size, q_0 is the inlet rate, D is the isotropic diffusion coefficient, V is the inlet velocity, and ν is the kinematic fluid viscosity.

2. Next the basic dimensions of each variable are expressed

$$C \doteq ML^{-3}$$

$$x \doteq L$$

$$y \doteq L$$

$$d \doteq L$$

$$q_0 \doteq MT^{-1}$$

$$D \doteq L^2T^{-1}$$

$$V \doteq LT^{-1}$$

$$\nu \doteq L^2T^{-1}.$$

3. The number of pi terms is determined using the Buckingham pi theorem. The Buckingham pi theorem states that a relationship, with k variables, that is dimensionally homogeneous can be reduced to a relationship of $k-r$ independent dimensionless variables, where r is the number of reference dimensions necessary to describe the variables. Since there are eight variables in (8.1) and three references dimensions are required to describe them, the number of pi terms is five

$$k - r = 8 - 3 = 5 \tag{8.2}$$

4. Next, three repeating variables were selected to create the five non-dimensional pi terms. In this case, the repeated variables choosen are d , q_0 , and ν .
5. The remaining variables are each multiplied by a combination of the repeating

variables to create the pi terms.

$$\Pi_1 = \frac{Cd\nu}{q_0} \quad (8.3)$$

$$\Pi_2 = \frac{x}{d} \quad (8.4)$$

$$\Pi_3 = \frac{y}{d} \quad (8.5)$$

$$\Pi_4 = \frac{Vd}{\nu} \quad (8.6)$$

$$\Pi_5 = \frac{D}{\nu} \quad (8.7)$$

6. To make sure that each of the pi terms are dimensionless, the dimensions are double checked.

$$\Pi_1 \doteq \frac{(ML^{-3})(L)(L^2T^{-1})}{MT^{-1}} \doteq M^0L^0T^0$$

$$\Pi_2 \doteq \frac{L}{L} \doteq M^0L^0T^0$$

$$\Pi_3 \doteq \frac{L}{L} \doteq M^0L^0T^0$$

$$\Pi_4 \doteq \frac{(LT^{-1})(L)}{L^2T^{-1}} \doteq M^0L^0T^0$$

$$\Pi_5 \doteq \frac{L^2T^{-1}}{L^2T^{-1}} \doteq M^0L^0T^0$$

7. Finally, the function is written as a relationship of the pi terms. Recognizing that inverse Π_5 is the Schmidt number, it's inverse is used in the final relationship.

$$\frac{Cd\nu}{q_0} = f\left(\frac{x}{d}, \frac{y}{d}, \frac{Vd}{\nu}, \frac{\nu}{D}\right) \quad (8.8)$$

8.2 Function Approximation

To approximate the function given in (8.8), numerical simulations were completed for five different inlet velocities, five different inlet sizes, and five different isotropic diffusion coefficients. The data produced from the simulations was approximated with a combination of radial basis functions given by

$$f(\lambda) := \sum_{j=1}^m \phi(\lambda)\theta_j = \phi^T(\lambda)\Theta \quad (8.9)$$

where $\phi(\lambda)$ and Θ are defined as

$$\begin{aligned}\phi^T(\lambda) &:= [\phi_1(\lambda) \phi_2(\lambda) \cdots \phi_m(\lambda)], \\ \Theta &:= [\theta_1 \theta_2 \cdots \theta_m]^T.\end{aligned}$$

The Gaussian radial basis functions $\{\phi(\lambda)\}$ are given by

$$\phi_j(\lambda) = \frac{1}{\Gamma_j} \exp \left(- \sum_{i \in \{1,2,3,4\}} \frac{\|\lambda_i - \kappa_{ij}\|^2}{\sigma_j^2} \right), \quad (8.10)$$

where σ_j is the width of the Gaussian basis and Γ_j is a normalizing constant. The centers of the basis functions $\{\kappa_{ij} \mid i \in \{1,2,3,4\}, j \in \{1,2,\dots,m\}\}$ are uniformly distributed in the function space, where $m = n_1 \times n_2 \times n_3 \times n_4$ and n_i is the number of centers needed to approximate λ_i . From the numerical simulations we have many observations of $\Pi_1(\lambda_k) = \phi^T(\lambda_k)\theta$, where k is the measurement index. Based on the observations and refressors $\{(\Pi_1(\lambda_k), \phi(\lambda_k))\}_{k=1}^n$, our objective is to find $\hat{\Theta}$ which minimizes the least-square error

$$\frac{1}{n} \sum_{k=1}^n |\Pi_1(\lambda_k) - \phi^T(\lambda_k)\hat{\Theta}|^2. \quad (8.11)$$

For a set $\{(\Pi_1(\lambda_k), \phi(\lambda_k))\}_{k=1}^n$, the optimal least-squares estimation solution is

$$\hat{\Theta}(\lambda_n) = P\Phi^T Y, \quad (8.12)$$

where

$$\begin{aligned}Y &:= [\Pi_1(\lambda_1) \ \Pi_1(\lambda_2) \ \cdots \ \Pi_1(\lambda_n)]^T \in \mathbb{R}^n, \\ \Phi &:= [\phi(\lambda_1) \ \phi(\lambda_2) \ \cdots \ \phi(\lambda_n)]^T \in \mathbb{R}^{n \times m}, \\ P &:= [\Phi^T \Phi]^{-1} \\ &= \left[\sum_{k=1}^n \phi(\lambda_k) \phi^T(\lambda_k) \right]^{-1} \in \mathbb{R}^{m \times m}.\end{aligned}$$

Since the local coordinates of the process used to model the locally model the concentration of the plume is different than the global coordinates as in Fig. 8.1, a

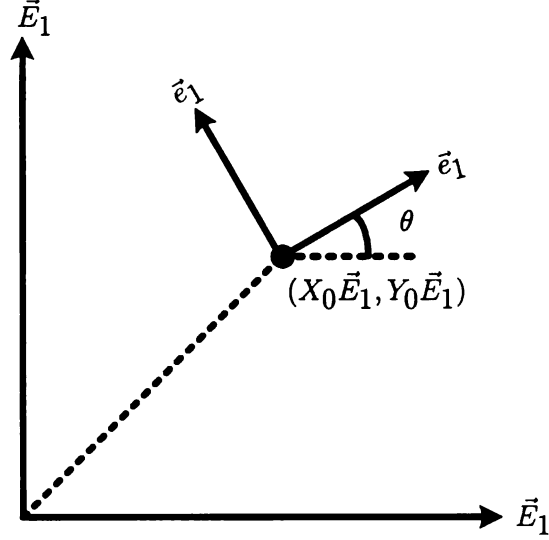


Figure 8.1. Local and Global Coordinate Relationship

similarity transformation must be used. In this case, the similarity transformation will include a transformation and rotation. Any set of global coordinates (X_i, Y_i) , can be written in local local coordinates (x_i, y_i) as follows:

$$(X_i - X_0)\vec{E}_1 + (Y_i - Y_0)\vec{E}_2 = x_i\vec{e}_1 + y_i\vec{e}_2. \quad (8.13)$$

The rotation between the local and global coordinate systems is handled by

$$\begin{bmatrix} \vec{e}_1 \\ \vec{e}_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \vec{E}_1 \\ \vec{E}_2 \end{bmatrix}, \quad (8.14)$$

where θ is the rotation angle. The following relationship is used to change the coordinates:

$$\begin{cases} x_i = (X_i - X_0) \cos \theta + (Y_i - Y_0) \sin \theta \\ y_i = -(X_i - X_0) \sin \theta + (Y_i - Y_0) \cos \theta \end{cases}, \quad (8.15)$$

where x_i and y_i are local coordinates and X_i and Y_i are global coordinates.

CHAPTER 9

Conclusion

This paper presents optimal sampling strategies for estimating the parameters of an advection-diffusion process using nonholonomic sensing agents. Two problems are stated, then the components needed to approach these problems are discussed. Simulation results are presented which show that the strategy proposed in Section 6.1 works. The simulation results for strategy proposed in Section 6.2 suggest that a more accurate model of the local plume is needed. Future work will be done to make a numerical closed-form solution to model the local plume behavior. Once the numerical closed-form solution is found, it will be used to trace the concentration field displayed in Fig. 7.6 and other more complicated concentration fields. Future work will also include using other strategies trace the source of realistic advection-diffusion processes.

Appendix

Sensitivity Coefficients

Computation

In what follows, the sensitivity matrix is computed, which will require partial derivatives with respect to each of parameters.

The analytical solution of the PDE equation is given by

$$\begin{aligned}\bar{C}(x, y, t) &= \frac{C_0}{\sqrt{(4\pi)^3 (t - t_0)^3 (D_{xx}D_{yy} - D_{xy}^2)}} \\ &\times \exp \left\{ -\frac{1}{4(D_{xx}D_{yy} - D_{xy}^2) (t - t_0)} \right. \\ &\times [(\chi - \chi_0 - V(t - t_0))]^T \begin{bmatrix} D_{yy} & -D_{xy} \\ -D_{xy} & D_{xx} \end{bmatrix} [(\chi - \chi_0 - V(t - t_0))] \left. \right\}\end{aligned}\quad (1)$$

The partial derivative of $C(\cdot)$ w.r.t D_{xx} is

$$\frac{\partial \bar{C}}{\partial D_{xx}} = \left(\frac{-D_{yy}C_0}{2\sqrt{(4\pi)^3 (t - t_0)^3 (D_{xx}D_{yy} - D_{xy}^2)^3}} + C_0^* \frac{\partial}{\partial D_{xx}}(\star) \right) \exp(\star), \quad (2)$$

where

$$\begin{aligned}C_0^* &= \frac{C_0}{\sqrt{(4\pi)^3 (t - t_0)^3 (D_{xx}D_{yy} - D_{xy}^2)}} \\ \star &= -\frac{1}{4(D_{xx}D_{yy} - D_{xy}^2) (t - t_0)} \\ &\times [(\chi - \chi_0 - V(t - t_0))]^T \begin{bmatrix} D_{yy} & -D_{xy} \\ -D_{xy} & D_{xx} \end{bmatrix} [(\chi - \chi_0 - V(t - t_0))]\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial}{\partial D_{xx}}(\star) &= \frac{D_{yy}}{4(D_{xx}D_{yy} - D_{xy}^2)^2(t-t_0)} \times [(\chi - \chi_0 - V(t-t_0))]^T \\
&\times \begin{bmatrix} D_{yy} & -D_{xy} \\ -D_{xy} & D_{xx} \end{bmatrix} \times [(\chi - \chi_0 - V(t-t_0))] \\
&- \frac{1}{4(D_{xx}D_{yy} - D_{xy}^2)(t-t_0)} \times [(\chi - \chi_0 - V(t-t_0))]^T \\
&\times \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \times [(\chi - \chi_0 - V(t-t_0))] \\
\frac{\partial \bar{C}}{\partial D_{yy}} &= \left(\frac{-D_{xx}C_0(t-t_0)^3}{2\sqrt{(4\pi)^3(t-t_0)^3(D_{xx}D_{yy} - D_{xy}^2)^3}} + C_0^* \frac{\partial}{\partial D_{yy}}(\star) \right) \exp(\star), \quad (3)
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial}{\partial D_{yy}}(\star) &= \frac{D_{xx}}{4(D_{xx}D_{yy} - D_{xy}^2)^2(t-t_0)} \times [(\chi - \chi_0 - V(t-t_0))]^T \\
&\times \begin{bmatrix} D_{yy} & -D_{xy} \\ -D_{xy} & D_{xx} \end{bmatrix} \times [(\chi - \chi_0 - V(t-t_0))] \\
&- \frac{1}{4(D_{xx}D_{yy} - D_{xy}^2)(t-t_0)} \times [(\chi - \chi_0 - V(t-t_0))]^T \\
&\times \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \times [(\chi - \chi_0 - V(t-t_0))] \\
\frac{\partial \bar{C}}{\partial D_{xy}} &= \left(\frac{D_{xy}C_0}{\sqrt{(4\pi)^3(t-t_0)^3(D_{xx}D_{yy} - D_{xy}^2)^3}} + C_0^* \frac{\partial}{\partial D_{xy}}(\star) \right) \exp(\star), \quad (4)
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial}{\partial D_{xy}}(\star) &= \frac{-D_{xy}}{2(D_{xy}^2 - D_{xx}D_{yy})^2(t-t_0)} \times [(\chi - \chi_0 - V(t-t_0))]^T \\
&\times \begin{bmatrix} D_{yy} & -D_{xy} \\ -D_{xy} & D_{xx} \end{bmatrix} \times [(\chi - \chi_0 - V(t-t_0))] \\
&- \frac{1}{4(D_{xx}D_{yy} - D_{xy}^2)(t-t_0)} \times [(\chi - \chi_0 - V(t-t_0))]^T \\
&\times \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \times [(\chi - \chi_0 - V(t-t_0))]
\end{aligned}$$

$$\frac{\partial \bar{C}}{\partial C_0} = \frac{1}{\sqrt{(4\pi)^3 (t - t_0)^3 (D_{xx} D_{yy} - D_{xy}^2)}} \exp(\star) \quad (5)$$

$$\begin{aligned} \frac{\partial \bar{C}}{\partial \chi_0} &= C_0^* \exp(\star) \\ &\times \left[\frac{1}{2(D_{xx} D_{yy} - D_{xy}^2)(t - t_0)} \begin{bmatrix} D_{yy} & -D_{xy} \\ -D_{xy} & D_{xx} \end{bmatrix} (\chi - \chi_0 - V(t - t_0)) \right]. \end{aligned} \quad (6)$$

Using chain-rule

$$\frac{\partial \bar{C}}{\partial t_0} = \frac{\partial \bar{C}}{\partial \Delta t} \frac{\partial \Delta t}{\partial t_0}$$

with

$$\Delta t = t - t_0$$

such that

$$\frac{\partial \Delta t}{\partial t_0} = -1$$

and

$$\frac{\partial \bar{C}}{\partial t_0} = - \left(\frac{\partial \bar{C}}{\partial \Delta t} \right) \quad (7)$$

where

$$\frac{\partial C}{\partial \Delta t} = \left(- \frac{3C_0}{2\sqrt{(4\pi)^3 \Delta t^5 (D_{xx} D_{yy} - D_{xy}^2)}} + C_0^* \frac{\partial}{\partial \Delta t}(\star) \right) \exp(\star)$$

and

$$\begin{aligned} \frac{\partial}{\partial \Delta t}(\star) &= \frac{-1}{4\Delta t^2 (D_{xx} D_{yy} - D_{xy}^2)} \times (\chi - \chi_0 - V(\Delta t))^T \\ &\times \begin{bmatrix} D_{yy} & -D_{xy} \\ -D_{xy} & D_{xx} \end{bmatrix} (\chi - \chi_0 - V(\Delta t)) \\ &+ \frac{1}{4\Delta t (D_{xx} D_{yy} - D_{xy}^2)} \\ &\times \left[2V^T \begin{bmatrix} D_{yy} & -D_{xy} \\ -D_{xy} & D_{xx} \end{bmatrix} (\chi - \chi_0 - V(\Delta t)) \right]. \end{aligned}$$

BIBLIOGRAPHY

- [1] Julius Adler. Chemotaxis in bacteria. *Journal of Supramolecular Structure*, 4(3):305–317, 1966.
- [2] M. H. G. Amin, S. J. Gibbs, R. J. Chorley, K. S. Richards, T. A. Carpenter, and L. D. Hall. Study of flow and hydrodynamics dispersion in a porous medium using pulsed field gradient magnetic resonance. *Proc. R. Soc. Lond. A*, 453:489–513, 1997.
- [3] Vassilios N. Chistopoulos and Stergios Roumeliotis. Adaptive sensing for instantaneous gas release parameter estimation. In *Proceeding of the 2005 IEEE International Conference on Robotics and Automation*, 2005.
- [4] David S. DeCroix. Large-eddy simulated of urban dispersion during the URBAN2000 field program IOP-10, 25-26 October 2000.
- [5] Amit Dhariwal, Gaurav S. Sukhatme, and Aristides A. G. Requicha. Bacterium-inspired robots for environmental monitoring. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004.
- [6] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numeriche Mathematik*, pages 269–271, 1959.
- [7] A F Emery and A V Nenarokomov. Meas. sci. technol. 9. pages 864–876, 1998.
- [8] Jay A. Farrell. Chemical plume tracing via an autonomous underwater vehicle. *IEEE Journal of Oceanic Engineering*, 30(2):428–442, April 2005.

- [9] Yongxing Hao and Sunil K. Agrawal. Planning and control of ugv formations in a dynamic environment: a practical framework with experiments. *Robotics and Autonomous Systems*, pages 101–110, 2005.
- [10] Adam T. Hayes, Alcherio Martinoli, and Rodney M. Goodman. Swarm robotic odor localization. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robotics and Systems*, 2001.
- [11] Petter Ögren, Edward Fiorelli, and Naomi Ehrich Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transaction on Automatic Control*, 49(8):1292, August 2004.
- [12] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
- [13] H. Ishida, Y. Kagawa, T. Nakamoto, and T. Moriizumi. Odor-source localization in the clean room by an autonomous mobile sensing system. *Sensors and Actuators B*, 33:115–121, 1996.
- [14] H. Ishida, Y. suetsugu, T. Nakamoto, and T. Moriizumi. Study of an autonomous mobile sensing system for localization of odor source using gas sensors and anemometric sensors. *Sensors and Actuators A*, 45:153–157, 1994.
- [15] P. Kathirgamanathan, R. McKibbin, and R.I. McLachlan. Source term estimation of pollution from an instantaneous point source. *Research Letters in the Information and Mathematical Sciences*, 3(1):59–67, April 2002.
- [16] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993.
- [17] N. Leonardo and A. Robinson. Adaptive sampling and forecasting plan AOSN-II MB'03 project. Technical report, Available: <http://www.princeton.edu/dcs/aosn/documents/ASFP.pdf>, 2003.

- [18] Lennart Ljung and Torsten Söderström. *Theory and Practice of Recursive Identification*. The MIT Press, Cambridge, Massachusetts, London, England, 1983.
- [19] Jorg Matthes, Lutz Groll, and Hubert D. Keller. Source localization by spatially distributed electronic noses for advection and diffusion. *IEEE Transactions on Signal Processing*, pages 1711–1719, May 2005.
- [20] Nils J. Nilsson. *Principles of Artificial Intelligence*. Tioga Publishing Co., 1980.
- [21] Giuseppe Oriolo, Alessandro De Luca, and Marilena Vendittelli. Wmr control via dynamic feedback linearization: Design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, pages 835–852, November 2002.
- [22] J. C. Parker and M. T. van Genuchten. Flux-averaged and volume-averaged concentrations in continuum approaches to solute transport. *Water Resour. Res.*, 20:866–872, 1984.
- [23] Boaz Porat and Arye Nehorai. Localization vapor-emitting sources by moving sensors. *IEEE Transactions on Signal Processing*, 44(4):1018–1021, April 1996.
- [24] Wei Ren. Consensus seeking in multi-vehicle systems with a time-varying reference state. *Proceedings of the 2007 American Control Conference*, pages 717–722, July 2007.
- [25] Wei Ren. Second-order consensus algorithm with extensions to switching topologies and reference models. *Proceedings of the 2007 American Control Conference*, pages 1431–1436, July 2007.
- [26] W. S. Smith, M. J. Brown, and D. S. DeCoix. Evaluation of CFD simulations using laboratory data and urban field experiments. *12th Joint Conference on the Applications of Air Pollution Meteorology with the Air and Waste Management Association*, 2002.

- [27] Nathan Sorensen and Wei Ren. A unified formation control scheme with a single or multiple leaders. *Proceedings of the 2007 American Control Conference*, pages 5412–5418, July 2007.
- [28] Panos Tzanos, Milos Zefran, and Arye Nehorai. Information based distributed control for biochemical source detection and localization. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*.
- [29] Dimitri Zarzhitsky, Diana F. Spears, and William M. Spears. Distributed robotics approach to chemical plume tracing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 03062 5341