

A Distributed Optimal Control Approach for Multi-agent Trajectory Optimization

by

Greg Foderaro

Department of Department of Mechanical Engineering and Materials Science
Duke University

Date: _____

Approved:

Silvia Ferrari, Supervisor

Devandra P Garg

Laurens Howle

Ronald Parr

Thomas A Wettergren

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Department of Mechanical Engineering
and Materials Science
in the Graduate School of Duke University
2013

ABSTRACT

A Distributed Optimal Control Approach for Multi-agent
Trajectory Optimization

by

Greg Foderaro

Department of Department of Mechanical Engineering and Materials Science
Duke University

Date: _____

Approved:

Silvia Ferrari, Supervisor

Devandra P Garg

Laurens Howle

Ronald Parr

Thomas A Wettergren

An abstract of a dissertation submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in the Department of Department of Mechanical
Engineering and Materials Science
in the Graduate School of Duke University
2013

Copyright © 2013 by Greg Foderaro
All rights reserved except the rights granted by the
Creative Commons Attribution-Noncommercial Licence

Abstract

This dissertation presents a novel distributed optimal control (DOC) problem formulation that is applicable to multiscale dynamical systems comprised of numerous interacting systems, or agents, that together give rise to coherent macroscopic behaviors, or coarse dynamics, that can be modeled by partial differential equations (PDEs) on larger spatial and time scales. The DOC methodology seeks to obtain optimal agent state and control trajectories by representing the system's performance as an integral cost function of the macroscopic state, which is optimized subject to the agents' dynamics. The macroscopic state is identified as a time-varying probability density function to which the states of the individual agents can be mapped via a restriction operator. Optimality conditions for the DOC problem are derived analytically, and the optimal trajectories of the macroscopic state and control are computed using direct and indirect optimization algorithms. Feedback microscopic control laws are then derived from the optimal macroscopic description using a potential function approach.

The DOC approach is demonstrated numerically through benchmark multi-agent trajectory optimization problems, where large systems of agents were given the objectives of traveling to goal state distributions, avoiding obstacles, maintaining formations, and minimizing energy consumption through control. Comparisons are provided between the direct and indirect optimization techniques, as well as existing methods from the literature, and a computational complexity analysis is presented.

The methodology is also applied to a track coverage optimization problem for the control of distributed networks of mobile omnidirectional sensors, where the sensors move to maximize the probability of track detection of a known distribution of mobile targets traversing a region of interest (ROI). Through extensive simulations, DOC is shown to outperform several existing sensor deployment and control strategies. Furthermore, the computation required by the DOC algorithm is proven to be far reduced compared to that of classical, direct optimal control algorithms.

To my parents, Albert and Susann, and my brother, Brian.

Contents

Abstract	iv
List of Tables	ix
List of Figures	x
List of Abbreviations and Symbols	xiii
Acknowledgements	xvii
1 Introduction	1
2 Background	7
2.1 Classical Optimal Control	7
2.2 Swarm-intelligence, Behavior-based Control, and Potential Field Approaches	9
3 DOC Multi-agent Trajectory Optimization Problem	12
3.1 Problem Formulation and Assumptions	12
3.2 Conservation Law Analysis	15
4 DOC Optimality Conditions	18
4.1 Deterministic Case	18
4.2 Stochastic Case	21
5 Probability Density Function Approach to Path Planning	26
5.1 Microscopic Feedback Control Laws	26
5.1.1 Decentralization of Agent Density Estimation	27

6	Numerical Solutions to DOC Problem	32
6.1	Solution via Direct Optimization	32
6.2	Solution via Indirect Method	36
6.3	Complexity Analysis	39
7	Planning Optimal Paths through Obstacle-Populated Regions via DOC	41
7.1	Problem Formulation	42
7.2	Numerical Simulations and Results	45
7.2.1	Optimal Agent Trajectories computed with DOC Approach	45
7.2.2	Environmental Effects on Dynamics	51
7.2.3	Comparison of Direct and Indirect Approaches	60
8	A Distributed Optimal Control Approach to Sensor Network Trajectory Optimization	68
8.1	Problem Formulation	69
8.2	Cooperative Track Detection	71
8.2.1	Background on Track Coverage	71
8.2.2	Probabilistic Track Coverage	74
8.3	Numerical Simulations and Results	77
9	Conclusions	91
	Bibliography	93
	Biography	100

List of Tables

6.1	Computational Complexity of SQP Solution	39
7.1	Performance comparison of trajectory planning approaches	67
8.1	Performance comparison of sensor deployment strategies.	83
8.2	Performance comparison of sensor deployment strategies for example case with obstacles.	90

List of Figures

5.1	Example of attractive potential based on difference between optimal PDF and actual sensor density. Yellow circles represent individual sensor positions, and solid black shapes represent geometric obstacles. (a) Optimal distribution φ^* plotted on background. (b) Attractive potential function U_{att} plotted on background.	28
7.1	Initial (a) and goal (b) agent distributions for a workspace with three obstacles (solid black).	46
7.2	Optimal evolution of agent distribution and microscopic state (yellow circles) for $N = 500$ microscopic agents, at four instants in time. State trajectories (blue lines) of 6 randomly chosen agents are plotted in (d).	48
7.3	Microscopic state and control histories for 3 agents randomly chosen from the example in Fig. 7.2.	49
7.4	Numerical error for optimality conditions (4.6) (a) and (4.7) (b) as a function of covariance perturbation parameters.	50
7.5	(a) Initial agent distribution. (b) Target agent distribution	52
7.6	Optimal solution for evolving agent distribution with objective to maintain triangular pattern. (a) $t = 0$, (b) $t = 8$, (c) $t = 15$, (d) $t = 21$	53
7.7	Current velocity field and obstacles in workspace.	54
7.8	(a) Initial agent distribution. (b) Target agent distribution	55
7.9	The evolution of the optimal agent PDF φ^* for the case with <i>no currents</i> is computed with the <i>direct</i> DOC algorithm and plotted at six instants in time.	56
7.10	The evolution of the optimal agent PDF φ^* and the microscopic agent trajectories for the case with <i>no currents</i> is computed with the <i>direct</i> DOC algorithm and plotted at six instants in time.	57

7.11	The evolution of the optimal agent PDF φ^* for the case with <i>currents</i> is computed with the <i>direct</i> DOC algorithm and plotted at six instants in time.	58
7.12	The evolution of the optimal agent PDF φ^* and the microscopic agent trajectories for the case with <i>currents</i> is computed with the <i>direct</i> DOC algorithm and plotted at six instants in time.	59
7.13	The evolution of the optimal agent PDF φ^* for the case with <i>no currents</i> is computed with the <i>indirect</i> DOC algorithm and plotted at six instants in time.	62
7.14	The evolution of the optimal agent PDF φ^* and the microscopic agent trajectories for the case with <i>no currents</i> is computed with the <i>indirect</i> DOC algorithm and plotted at six instants in time.	63
7.15	The evolution of the optimal agent PDF φ^* for the case with <i>currents</i> is computed with the <i>indirect</i> DOC algorithm and plotted at six instants in time.	64
7.16	The evolution of the optimal agent PDF φ^* and the microscopic agent trajectories for the case with <i>currents</i> is computed with the <i>indirect</i> DOC algorithm and plotted at six instants in time.	65
7.17	The microscopic agent trajectories for the case with currents is computed with the stochastic gradient descent algorithm and plotted at six instants in time.	66
7.18	Comparison of the Lagrangian function evaluated over the time period $(T_0, T_f]$ for the direct DOC, indirect DOC, and stochastic gradient descent approaches.	67
8.1	Coverage cone of an omnidirectional sensor at \mathbf{x} for a target track with origin \mathbf{x}_{T_0}	75
8.2	Initial sensor distribution (a) and PDF of initial target position (b) for obstacle-free example.	79
8.3	PDF of target heading for obstacle-free example.	80
8.4	Evolution of target PDF at three instants in time for obstacle-free example in Figs. 8.2-8.3.	80
8.5	Evolution of optimal sensor PDF, φ^* , microscopic state (black dots), and FOVs (black circles) at four instants in time, for obstacle-free example in Figs. 8.2-8.3, and $N = 100$ sensors.	81

8.6	Time-history of cost function Lagrangian for DOC solution in Fig. 8.5.	81
8.7	Grid (a), random (b), and uniform (c) sensor distributions and sampled sensor positions (black dots), and FOVs (black circles).	82
8.8	Initial sensor distribution (a) and PDF of initial target position for example with obstacles (solid black).	85
8.9	PDF of target heading for example with obstacles.	86
8.10	Evolution of target PDF at three instants in time for example with obstacles (solid black) in Figs. 8.8-8.9.	86
8.11	Evolution of optimal sensor PDF, φ^* , microscopic state (black dots), and FOVs (black circles) at four instants in time, for example with obstacles (solid black) in Figs. 8.8-8.9.	87
8.12	Time-history of cost function Lagrangian for DOC solution in Fig. 8.11.	87
8.13	Grid (a), random (b), and uniform (c) sensor distributions with sampled sensor positions (black dots), FOVs (black circles), and geometric obstacles (solid black).	88
8.14	Goal sensor PDF from which the goal microscopic states were sampled for the stochastic gradient method.	89
8.15	States of $N = 250$ sensors (solid yellow circles) with FOVs (black circles) controlled using the stochastic gradient approach and plotted at three instants in time. The sensors travel toward goal states (black dots) sampled from a goal sensor PDF (plotted on background).	90

List of Abbreviations and Symbols

Symbols

\mathcal{B}_i	Geometry of i^{th} obstacle
\mathcal{C}	Configuration space
\mathcal{CB}	Obstacle space
\mathbf{H}	Band-width matrix of j^{th} kernel
\mathcal{H}	Hamiltonian
J	Integral cost function
J_A	Augmented integral cost function
K	Kernel function
\mathcal{L}	Lagrangian
n	State dimensionality
N	The number of agents in the system
m	Control dimensionality
M	Number of basis functions
\wp	Restriction operator and time-varying probability density function of agent states
\wp^*	Optimal time-varying probability density function of agent states
\wp_0	Initial agent distribution
\mathbf{q}	Agent configuration
\mathbf{S}	Stored kernel set of i^{th} agent

t	Time
T_0	Initial time
T_f	Final time
\mathbf{u}	Agent control
\mathbf{u}_i	Control of the i^{th} agent
U	Artificial potential function
U_{att}	Attractive potential
U_{rep}	Repulsive potential
\mathcal{U}	Control space
\mathbf{w}_i	Disturbance on the i^{th} agent
\mathcal{W}	Workspace or region of interest
\mathbf{x}	Microscopic agent state
\mathbf{x}_i	Microscopic state of the i^{th} agent
\mathbf{x}_0	Initial microscopic agent state
\mathbf{X}	Macroscopic state
\mathcal{X}	State space
z	Number of components in Gaussian mixture model
λ	Lagrange multiplier
μ	Mean of Gaussian mixture component
ϕ	Terminal cost
ψ	Hamiltonian function
Σ	Covariance of Gaussian mixture component
ζ	Vector of Gaussian mixture model parameters in discrete time

Abbreviations

BFGS	Broyden-Fletcher-Goldfarb-Shanno
CPA	Closest-point-of-approach
DCLT	Detect, classify, localize, or track
DOC	Distributed optimal control
FOV	Field of view
FTVC	Fundamental theorem of variational calculus
FV	Finite volume
GRG	Generalized reduced gradient
HBVP	Hamiltonian boundary value problem
KDE	Kernel Density Estimation
KL	Kullback-Leibler
KKT	Karush-Kuhn-Tucker
LQR	Linear quadratic regulator
MHT	Multiple hypothesis tracking
NAND	Nested analysis and design
NLP	Nonlinear program
OC	Optimal Control
ODE	Ordinary differential equation
PDE	Partial differential equation
PDF	Probability density function
PSPACE	Polynomial space
QP	Quadratic program
ROI	Region of interest
SAND	Simultaneous analysis and design
SDE	Stochastic differential equation

SQP Sequential quadratic programming
VLSR Very large scale robotic

Acknowledgements

I would first like to thank my advisor, Dr. Silvia Ferrari. Your help, financial support, patience, and dedication to me over the duration of my graduate studies were invaluable. Your drive for your work helped to fuel my own passions and interests and helped to maintain my focus and motivation.

I would also like to give special mention to the members of my Ph.D. qualification, preliminary, and final defense committees for their time and effort: Dr. Tom Wettergren, Dr. Laurens Howle, Dr. Devendra Garg, Dr. Ronald Parr, and Dr. Craig Henriquez. I would also like to thank those who contributed to my better understanding of my research: Dr. Michael Zavlanos, Dr. Wilkins Aquino, Dr. John Albertson, and Dr. Rafael Fierro. A special thank you to my great labmates and fellow graduate students, who were always available and willing to help along the way: Keith Rudd, Wenjie Liu, Hongchuan Wei, Xu Zhang, Ziyu Xu, Ashleigh Swingler, Weston Ross, Pingping Zhu, Guoxian Zhang, Brian Bernard, Vikram Raju, Greyson Daugherty, Andrew Trembley, and Gianluca Di Muro.

I wish to dedicate this dissertation to my family. To my parents, Albert and Susann. Your support, love, and teachings made this possible. To my big brother, Brian. Thank you for your friendship, your belief in me, and for being a great role model throughout my life.

1

Introduction

Optimal control is a well-known approach for optimizing the performance of a dynamical system that is described by a small system of ordinary differential equations (ODEs). It has been successfully implemented in a myriad of applications, such as, for developing controllers for aircraft flight and stabilization, industrial chemical processes, robotics, and economics [75]. However, many existing problems of current interest involve highly complex systems that require capabilities beyond those of optimal control to maximize system performance. These applications can be found in a broad range of fields, and involve systems that are subject to significant non-Gaussian inputs, coupling effects between many systems, or models that consist of partial differential equations (PDEs). Optimal control methods can theoretically optimize coupled systems by formulating a high-level macroscopic performance as a combination of the system states and controls, but the approach is limited in practice because if the number of coupled systems is large, the computational complexity can become prohibitive. Systems that are subject to small random effects can be handled with existing stochastic optimal control methodologies, but these approaches are not suited for systems with large disturbances from random variables with arbitrary

probability distributions. Additionally, several complex systems require complicated expressions, such as PDEs, to represent the macroscopic system dynamics which cannot be derived in closed form and therefore cannot be optimized with optimal control.

This dissertation presents a novel approach for optimizing the system-level, or *macroscopic*, performance of a multiscale stochastic dynamical system consisting of a large number of *agents* that are modeled at a local, or *microscopic*, scale by ODEs or stochastic differential equations (SDEs) and exhibit a collective macroscopic behavior modeled by a PDE. A new *distributed optimal control* (DOC) problem formulation is proposed where the system's macroscopic state is represented as a probability density function (PDF), and the macroscopic performance is defined as an integral cost function subject to the macroscopic dynamics represented by a PDE. Necessary conditions for optimality are derived, and direct and indirect numerical algorithms are described to optimize the system performance and solve for the optimally evolving PDF state representation. Then optimal trajectories of the microscopic agents can be calculated from the optimal PDF by utilizing a novel artificial potential field function to construct microscopic control policies. Therefore, the DOC approach is a new optimal control methodology that extends the capabilities of optimal control to complex, multiscale systems.

The DOC methodology is demonstrated here numerically through multi-agent trajectory optimization problems, where the objective is to plan the paths of large groups of agents through obstacle-populated workspaces to goal configurations while avoiding obstacles and minimizing energy consumed. It was recently shown that optimizing the trajectories of N agents in an obstacle-populated environment is polynomial-space-hard (PSPACE-hard) in N [35]. Problems classified as PSPACE-hard are at least as difficult as any problem solvable in polynomial space (PSPACE), and the PSPACE class contains many problems for which no efficient solutions are

known. Therefore, a PSPACE-hard problem is generally considered to be computationally intractable for large N , as the most efficient algorithms that are currently available would require exponential deterministic time in the worst case [66].

Many complex systems ranging from renewable resources [67] to very large scale robotic (VLSR) systems [64] can be described as multiscale dynamical systems comprised of many interactive agents. Several approaches have been proposed for tackling the control of VLSR systems while avoiding the complexity issues for large N [19]. This is typically done by decoupling the problem into independent components for which solutions can be found quickly at the expense of optimality and completeness (i.e. the ability to find a solution if one exists). These approaches include prioritized planning techniques, which plan the paths of agents individually in order of priority [26, 80, 10, 18], and path coordination methods, which plan the paths of the robots independently then adjust the microscopic control laws to avoid mutual collisions [38, 45, 34]. Swarm-intelligence methods, such as foraging and schooling [30], view each agent as an interchangeable unit subject to local objectives and constraints through which the swarm can converge to a range of pre-defined distributions, but the solutions produced do not optimize the macroscopic system performance subject to the microscopic agent dynamics. Approaches that maintain the system's coupled behavior for multiple agents have also been presented [60, 61]. These methods utilize cell decomposition and roadmap techniques, but the computational complexity grows exponentially with the number of agents.

The DOC approach presented in this dissertation does not rely on decoupling the agents' dynamics or on specifying the agents' distribution *a priori*. Instead, DOC optimizes the macroscopic performance of the system subject to agent dynamics that are coupled via the objective function, and relies on the macroscopic dynamic equations and restriction operator that characterize the multiscale system to reduce the computational complexity of the optimal control problem. As a result, the compu-

tation required is significantly reduced compared to classical optimal control, and near-optimal trajectories of cooperative agents can be computed over large spatial and time scales.

The DOC methodology can also be applied to track coverage optimization for the control of distributed networks of mobile omnidirectional sensors, where the sensors are deployed to cooperatively track moving targets in a ROI. The emergence of dependable sensor network and autonomous-vehicle technologies have allowed for the production of affordable surveillance systems that are both effective and versatile. These systems often employ large numbers of autonomous vehicles carrying wireless sensors to cooperatively detect, classify, localize, or track (DCLT) multiple targets in dynamic nonlinear environments, and they are used for a wide variety of applications in both the civilian and military areas including the detection and tracking of possible intruders [82], the tracking and monitoring of endangered species [70], environmental monitoring, and battlefield surveillance [63]. These types of systems typically have very complex designs with many variables to consider, such as sensor selection, data routing protocols and sensor placements, and high numbers of sensors are often required to cover large regions of interest and to achieve acceptable confidence-levels for sensor detections. For this reason, the ability to optimize these systems can greatly influence their effectiveness [83].

Several authors have addressed the placement of sensors to meet DCLT objectives, assuming that the sensors are stationary [7, 57, 42]. Other existing approaches for generating sensor trajectories include area coverage [20, 1], random [1], grid [48], and optimal search strategies [48, 74]. Cooperative control methods have also been developed to manage the sensors' formation in response to the sensed environment [1, 33]. Although optimal control is often considered the most general and effective approach to trajectory optimization [75, 72], its applicability to mobile sensor networks has been very limited to date due to the lack of suitable DCLT objective

functions and to the computational complexity associated with solving the optimality conditions numerically for a large number of sensors [8].

Recently, a probabilistic track coverage performance function was derived in terms of the probability density function (PDF) of sensor positions by extending the geometric-transversal approach using Poisson flats [27]. Track coverage is a metric that relates to a sensor network’s ability to detect target vehicles as they traverse a region of interest (ROI). In many surveillance applications, track coverage is often seen as advantageous over other objective functions, such as area coverage, since it provides a high level of confidence when a target is detected [84, 46]. This is especially important under conditions where false alarms are frequent. The performance of networks of agents, such as sensors and autonomous vehicles, engaged in collaborative tasks, such as surveillance, search and rescue, and chemical plume detection and tracking, has also been recently shown to be a function of the agents PDF over the ROI [11, 59, 12, 83]. Therefore, one approach that has been proposed to deploy networks of agents is to sample an assumed PDF for the agent position [83]. Another approach is to perform locational optimization based on a given PDF, and obtain a network representation using centroidal Voronoi partitions [22, 53]. Finally, the trajectories of groups of agents can be computed using a hierarchical control approach [3], by first establishing a virtual adaptive boundary for the network, and then computing the lower-level control inputs to satisfy the boundary in lower-dimensional space.

Because existing approaches do not optimize the PDF (or virtual boundary) subject to the agent dynamic constraints, the agents may be unable to achieve the assumed distribution due to bounded state and control inputs. Conversely, if the assumed PDF is too conservative, the actual network performance may be suboptimal, and may not allow the network to achieve its maximum performance. Furthermore, because existing methods cannot be used to determine time-varying agent

distributions, they cannot fully exploit the capabilities of mobile sensors or take into account time-varying and heterogeneous environmental conditions. The DOC approach, addresses the problem of optimizing the time-varying macroscopic state, or PDF, for a multiscale dynamical system comprised of a system of agents in an obstacle-populated environment. Similarly, a sensor network can be viewed as a multiscale dynamical system in which the performance to be optimized is an integral function of the sensors' PDF, and the sensor dynamics are provided by the equations of motion of the sensor platforms.

This dissertation is organized as follows. Background on existing approaches for multi-agent trajectory optimization is reviewed in Chapter 2. The DOC problem formulation and assumptions are presented in Chapter 3. The derivation of the optimality conditions for the DOC problem are given in Chapter 4. Chapter 5 presents the probability density function approach for defining the agents' microscopic feedback control laws. Direct and indirect optimization algorithms for obtaining solutions to the DOC problem are described in Chapter 6. Numerical simulations and results for the problem of planning optimal agent trajectories through obstacle-populated regions via DOC and a comparison of the algorithms outlined in Chapter 6 are presented in Chapter 7. Chapter 8 presents numerical simulations for the sensor network track coverage optimization problem. And finally, Chapter 9 gives the conclusions of this dissertation.

2

Background

2.1 Classical Optimal Control

Optimal control can be considered the most general approach to optimizing the performance of a dynamical system over time. Since its inception in the early 1970s, it has been applied to a variety of dynamical systems, including physical, chemical, economic, mechanical and air vehicles, in order to derive optimal control laws or trajectories. However, despite its effectiveness at solving a large number of control and optimization problems, classical optimal control experiences several shortcomings that severely limit its applicability to distributed dynamical systems [8].

The classical optimal control formulation considers a system whose dynamics can be approximated by a small system of ODEs,

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t], \quad \mathbf{x}(T_0) = \mathbf{x}_0 \quad (2.1)$$

where, $\mathbf{x} \in \mathcal{X} \subset \mathfrak{R}^n$ is the system state, and $\mathbf{u} \in \mathcal{U} \subset \mathfrak{R}^m$ is the control [75]. The dynamics in (2.1) also depend on system parameters that represent the physical characteristics of the system and scale the system's response to control inputs and to its own motions. Optimal control seeks to determine the state and control trajectories

that optimize an integral cost function,

$$J = \phi[\mathbf{x}(T_f)] + \int_{T_0}^{T_f} \mathcal{L}[\mathbf{x}(t), \mathbf{u}(t), t] dt \quad (2.2)$$

over a time interval $[T_0, T_f]$, subject to (2.1) and, potentially, to an r -dimensional inequality constraint

$$\mathbf{q}[\mathbf{x}(t), \mathbf{u}(t), t] \leq \mathbf{0}_{r \times 1} \quad (2.3)$$

The necessary conditions for optimality are given by the well-known Euler-Lagrange equations, which can be derived using calculus of variations [75]. When the system dynamics are linear and the cost function is quadratic, a linear-optimal control law known as a linear quadratic regulator (LQR) can be obtained from the matrix Riccati equation with a terminal condition, and its solutions constitute necessary and sufficient conditions for optimality. For a nonlinear system and a general cost function, the necessary conditions for optimality amount to a Hamiltonian boundary-value (HBVP) problem for which there are no closed-form solutions and, therefore, they typically are solved numerically [75, 14].

If the observation process is uncertain or the dynamic system is forced by random disturbances, then the problem is referred to as a stochastic optimal control problem. In the optimal control literature to date, emphasis has been placed on the class of stochastic systems with small random effects because useful solutions are not yet available for the stochastic optimization of nonlinear systems with random variables of arbitrary probability distributions [75, pg. 421]. Furthermore, optimal control has limited scalability. In principle, optimal control can be extended to N dynamical systems by considering N coupled differential equations and by formulating their cooperative performance as a single cost function of an Nn -dimensional state and an Nm -dimensional control, where n and m are the dimensions of the microscopic

state and control, respectively. However, the computational complexity of solving the corresponding optimality conditions typically becomes prohibitive for large values of N . Additionally, the classical optimal control formulation is not well-suited to systems in which the macroscopic dynamics cannot be derived in closed form. As is demonstrated in the chapters to follow, the DOC methodology presented in this dissertation is capable of overcoming each of these limitations and therefore extends the capabilities of optimal control to complex systems described by numerous interacting dynamical systems.

2.2 Swarm-intelligence, Behavior-based Control, and Potential Field Approaches

The use of potential fields is a well-known approach to agent motion planning that treats the agent as a particle under the influence of an artificial potential field or function, U , that captures the geometric characteristics of the workspace or region of interest (ROI), \mathcal{W} . The approach then results in control laws that instruct the agents to follow the directions of steepest descent within the potential function. So far, several potential field methods have been developed for generating a collision-free path for an agent that must travel from an initial configuration \mathbf{q}_0 to a goal configuration \mathbf{q}_f , without a prior model of the obstacles. The advantage of the potential field technique over other motion planning approaches, such as, cell decomposition and probabilistic roadmap methods, is that it can easily account for obstacles that are sensed *online*, i.e., during the motion execution [44]. However, since each agent exhibits a greedy behavior, the potential field method is not capable of producing trajectories that optimize a macroscopic performance function of a system consisting of several cooperative agents. In addition, because the agents follow directions of steepest descent, an agent can potentially get stuck at a local minimum. In this case, the method can be combined with a graph searching technique, or a random-walk

algorithm, to help the agent escape local minima.

Several potential functions have been proposed in the literature to generate U , such that the agent can be guaranteed to reach \mathbf{q}_f , while effectively avoiding obstacles in \mathcal{W} [65, 69, 31]. Typically, the potential function is the sum of an attractive potential U_{att} that “pulls” the agent toward \mathbf{q}_f , and a repulsive potential U_{rep} that “pushes” the agent away from the obstacles, i.e.:

$$U(\mathbf{q}) = U_{att}(\mathbf{q}) + U_{rep}(\mathbf{q}) \tag{2.4}$$

The method is implemented by discretizing the agent workspace obtained by the Cartesian product, $\mathcal{G} \times \mathcal{C} \rightarrow \mathcal{W}$, between the agent geometry \mathcal{G} , and the configuration space \mathcal{C} . The potential function is evaluated for all $\mathbf{q} \in \mathcal{C}$, using a finite resolution grid [6] which, in on-line motion planning, can be limited to the neighborhood of the agent configuration at the present time t . Subsequently, at any time $t \in [T_0, T_f]$, an artificial force $\mathbf{F}(\mathbf{q})$ that is proportional to the negative gradient of the artificial potential, $-\nabla U(\mathbf{q})$, is applied to the agent, such that the agent will follow the steepest-descent direction of U .

Every obstacle in \mathcal{W} , where \mathcal{B}_i denotes the geometry of the i^{th} obstacle, maps in \mathcal{C} to a C-obstacle that is defined as the subset of \mathcal{C} that causes collisions with \mathcal{B}_i , i.e., $\mathcal{CB}_i \equiv \{\mathbf{q} \in \mathcal{C} \mid \mathcal{G}(\mathbf{q}) \cap \mathcal{B}_i \neq \emptyset\}$, where $\mathcal{G}(\mathbf{q})$ represents the subset of \mathcal{W} occupied by the platform geometry \mathcal{G} when the agent is in the configuration \mathbf{q} . The union of all C-obstacles in \mathcal{W} is referred to as the C-obstacle region. Thus, to avoid collisions, the agent is free to explore the free configuration space, defined as the complement of the C-obstacle region \mathcal{CB} in \mathcal{C} , i.e., $\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{CB}$ [44].

In an alternate branch of work called *swarm intelligence*, potential field methods have been adapted to control systems of distributed agents [30, 64, 49]. Swarming, in the biological sense, refers to the behavior seen in nature where social organisms, such as ants or fish, aggregate together into groups to accomplish high-level objec-

tives, for example foraging and defense. Each individual organism operates with a simple local strategy, usually only reacting to the environment and other nearby organisms. This large number of local interactions combine to form a collective behavior that works to achieve the swarm objectives. Similarly, this concept is applied to a distributed system of agents, where each agent follows a simple local strategy that has been designed to produce a desired macroscopic behavior. In the swarming problems relevant to this research, individual agents within the swarms are controlled by using modified attractive and repulsive potentials to form the local strategy, or microscopic control law. This approach has been implemented on several distributed agent path planning problems with macroscopic objectives that include navigating to a target configuration, maintaining a swarm formation, and traveling to desirable environmental locations (e.g. areas providing nutrition when foraging). This swarming methodology has also been shown to be effective in such problems when the environment is not known *a priori*. However, the approach is not capable of producing trajectories of several agents that optimize the macroscopic performance due to the greedy behavior implicit in potential field methods.

The DOC methodology overcomes the limitations in these methods by generating the optimal trajectories of the macroscopic representation of the distributed system when optimizing the performance where the macroscopic representation is defined as a PDF of the microscopically evolving agent states. Therefore, DOC can perform the optimization subject to the dynamics of all agents in the distributed system. To obtain the microscopic control laws for the individual agents, the potential field approach was utilized to design a novel attractive potential that can be used for navigation via PDFs. Then by using the optimal PDF and the novel potential function, we provide the means to determine the optimal trajectories of the individual sensors from the optimal macroscopic description of the system.

DOC Multi-agent Trajectory Optimization Problem

3.1 Problem Formulation and Assumptions

This work considers the problem of computing the optimal state and control trajectories for a multiscale dynamical system comprised of N dynamical systems, referred to as agents, that can each be described by a small system of SDEs, referred to as the detailed equation,

$$\dot{\mathbf{x}}_i(t) = \mathbf{f}[\mathbf{x}_i(t), \mathbf{u}_i(t), t] + \mathbf{G}\mathbf{w}_i(t), \quad \mathbf{x}_i(T_0) = \mathbf{x}_{i_0}, \quad (3.1)$$

where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^n$ and $\mathbf{u}_i \in \mathcal{U} \subset \mathbb{R}^m$ denote the microscopic state and control of the i^{th} agent, respectively, \mathcal{X} denotes the n -dimensional microscopic state space, and \mathcal{U} is the space of m admissible microscopic control inputs. The microscopic dynamics are influenced by additive Gaussian noise, where the disturbance on the i^{th} agent, $\mathbf{w}_i \in \mathbb{R}^n$, is a vector of independent and identically distributed random variables from a standard Gaussian process, and \mathbf{G} is a time-invariant matrix. A standard Gaussian process is used here for simplicity, but this approach is applicable to any diffusion process. In this paper, the notations \mathbf{x}_i and \mathbf{u}_i represent the state and control of a specific agent i , and \mathbf{x} and \mathbf{u} are used as general geometric values within \mathcal{X} and \mathcal{U}

and do not correspond to any single agent. It is assumed that the microscopic states, \mathbf{x}_i , of every agent $i = 1, \dots, N$ are fully observable and free of error.

On larger spatial and temporal scales, the interactions of the N agents give rise to macroscopic coherent behaviors, or coarse dynamics. A macroscopic state, denoted by $\mathbf{X} \in \mathbb{R}^\ell$, can be introduced that consists of $\ell < n$ variables that capture the macroscopic system dynamics and performance, such as the lower-order moments of the evolving distribution of the agents' microscopic states [40]. It can be deduced from the detailed equation, either by deriving it from first principles, or using equation-free methods [40], and it is assumed that the multiscale system's performance can be defined as a function of \mathbf{X} . A restriction operator φ can be determined that maps the collective microscopic states of all agents to a macroscopic description \mathbf{X} . Since the microscopic agent states change with time, φ is a time-varying PDF of the agent states defined over the microscopic state space \mathcal{X} , such that $\varphi : \mathcal{X} \times \mathbb{R} \rightarrow \mathbb{R}$ and $\mathbf{X}(t) = \varphi(\mathbf{x}, t)$. Then the probability of the i^{th} agent having the state \mathbf{x}_i at time t is given by,

$$P(\mathbf{x}_i \in B) = \int_B \varphi(\mathbf{x}, t) d\mathbf{x} \quad (3.2)$$

for any subset $B \subset \mathcal{X}$, where φ is a non-negative function that satisfies the normalization property,

$$\int_{\mathcal{X}} \varphi(\mathbf{x}, t) d\mathbf{x} = 1 \quad (3.3)$$

Furthermore, $N\varphi(\mathbf{x}, t)$ represents the density of agents in \mathcal{X} .

The coarse dynamics of the multiscale system can be derived from the continuity equation, which is a PDE that governs the motion of a conserved, scalar quantity, where in this context the quantity is a probability. Through state constraints, it is possible to guarantee that, at any time $t \in (T_0, T_f]$, $\mathbf{x}_i(t) \in \mathcal{X}$ for all i , and, thus, agents in \mathcal{X} are never created nor destroyed. Then since the agent dynamics are

given by the detailed equation (3.1), it is known that φ is advected by the velocity field $\mathbf{v} = \mathbf{f}[\mathbf{x}, \mathbf{u}, t] \in \mathbb{R}^n$ and diffused by the additive Gaussian noise $\mathbf{G}\mathbf{w}$. From the continuity equation and Gauss's theorem, the time-rate of change of φ can be defined as the sum of the negative divergence of the advection vector ($\varphi\mathbf{v}$) and the divergence of diffusion vector ($\mathbf{G}\mathbf{G}^T\nabla\varphi$) [52]. Then, the dynamics of the agent PDF are governed by a parabolic PDE called the advection-diffusion equation,

$$\begin{aligned} \frac{\partial\varphi}{\partial t} &= -\nabla \cdot \{\varphi[\mathbf{x}(t), t]\mathbf{v}(t)\} + \nabla \cdot \{(\mathbf{G}\mathbf{G}^T)\nabla\varphi[\mathbf{x}(t), t]\} \\ &= -\nabla \cdot \{\varphi[\mathbf{x}(t), t]\mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t]\} + \nu\nabla^2\varphi[\mathbf{x}(t), t] \end{aligned} \quad (3.4)$$

where, the gradient ∇ denotes a row vector of partial derivatives with respect to the elements of \mathbf{x} , (\cdot) denotes the dot product, and the divergence is written as the dot product between $(\varphi\mathbf{v})$ and the gradient ∇ . If the microscopic dynamics are not influenced by any disturbances, such that $\mathbf{w} = 0$, then the coarse dynamics are given by the advection equation, which is a hyperbolic PDE,

$$\frac{\partial\varphi}{\partial t} = -\nabla \cdot [\varphi(\mathbf{x}, t) \mathbf{v}] \quad (3.5)$$

$$= -\nabla \cdot [\varphi(\mathbf{x}, t) \mathbf{f}(\mathbf{x}, \mathbf{u}, t)] \quad (3.6)$$

The reader is referred to [17] for a detailed derivation of the advection-diffusion and advection equations.

Assuming the initial agent distribution is a known PDF φ_0 , the macroscopic evolution equation (3.6) is subject to the following initial and boundary conditions,

$$\varphi(\mathbf{x}, T_0) = \varphi_0(\mathbf{x}) \quad (3.7)$$

$$\varphi(\mathbf{x}, t) = 0, \quad \forall t \in (T_0, T_f], \quad \forall \mathbf{x} \in \partial\mathcal{X} \quad (3.8)$$

such that agents remain in the interior of \mathcal{X} at all times. Additionally, φ must obey the normalization condition (3.3), and the state constraint

$$\varphi(\mathbf{x}, t) = 0, \quad \forall t \in (T_0, T_f], \quad \forall \mathbf{x} \notin \mathcal{X} \quad (3.9)$$

The macroscopic system performance is a function of the agent distribution and control, and it can be expressed as an integral cost function of the system macroscopic state, $\mathbf{X}(t) = \wp(\mathbf{x}, t)$, and control, \mathbf{u} ,

$$J = \phi[\wp(\mathbf{x}, T_f)] + \int_{T_0}^{T_f} \int_{\mathcal{X}} \mathcal{L}[\wp(\mathbf{x}, t), \mathbf{u}(t), t] d\mathbf{x} dt \quad (3.10)$$

where \mathcal{L} is the Lagrangian and ϕ is the terminal cost. Then, the DOC problem consists of finding the optimal agent distribution, \wp^* , and controls, \mathbf{u}^* , that minimize the macroscopic cost J over a (large) time interval $(T_0, T_f]$, subject to the dynamic constraint (3.6), the normalization condition (3.3), the initial and boundary conditions (3.7)-(3.8), and the state constraint (3.9). Since the DOC problem does not obey the classical optimal control formulation [75], new optimality conditions are derived in Chapter 4, and then they are validated numerically in Chapter 7 through a multi-agent path planning problem.

3.2 Conservation Law Analysis

In this section, we prove that the dynamics of the closed-loop DOC problem have a Hamiltonian structure. Consequently, the dynamics are shown to conserve the agent PDF, \wp , as it evolves in time, and a conservative numerical scheme, such as finite volume (FV), can be used to discretize the evolution equation (3.5) [77]. The Hamiltonian structure provides a constant of motion for the trajectories of the controlled system dynamics [32]. Optimal trajectories thus correspond to trajectories that have vanishing variations along these constants of motion according to the maximum principle of optimal control [41]. Because the coarse dynamics can be described by the advection equation (3.5), the open-loop system is inherently conservative [73]. In this section, we show that the DOC problem satisfies the Hamilton equations,

$$\frac{\partial \psi}{\partial \mathbf{q}} = -\frac{d\mathbf{p}}{dt}, \quad \frac{\partial \psi}{\partial \mathbf{p}} = \frac{d\mathbf{q}}{dt} \quad (3.11)$$

where $\psi = \psi(\mathbf{p}, \mathbf{q}, t)$ is the Hamiltonian function, $\mathbf{q} = \mathbf{q}(t) \in \mathbb{R}^n$ are the generalized coordinates, and $\mathbf{p} = \mathbf{p}(t) \in \mathbb{R}^n$ are the generalized momenta.

For simplicity, the proof is presented for $n = 2$, where $\mathbf{x} = [x \ y]^T$ denotes the position of the i^{th} agent in \mathbb{R}^2 . Then, the Hamiltonian function is determined by recasting the detailed equation (3.1) into a three-dimensional time-invariant ODE. Letting $\hat{\mathbf{x}} = [x \ y \ t]^T$ and $\hat{\mathbf{u}}(\hat{\mathbf{x}}) = \mathbf{u}(t)$, (3.1) can be written as,

$$\begin{bmatrix} \dot{x}(\hat{\mathbf{x}}, \hat{\mathbf{u}}) & \dot{y}(\hat{\mathbf{x}}, \hat{\mathbf{u}}) & \dot{t} \end{bmatrix}^T = \hat{\mathbf{f}}(\hat{\mathbf{x}}, \hat{\mathbf{u}}) \quad (3.12)$$

where, \mathcal{X} is transformed into the time-space domain $\hat{\mathcal{X}} = \mathcal{X} \times (T_0, T_f]$. It also follows that the macroscopic evolution equation (3.5) can be rewritten as,

$$\frac{\partial \varphi(\hat{\mathbf{x}})}{\partial t} + \frac{\partial [\varphi(\hat{\mathbf{x}}) \dot{x}(\hat{\mathbf{x}}, \hat{\mathbf{u}})]}{\partial x} + \frac{\partial [\varphi(\hat{\mathbf{x}}) \dot{y}(\hat{\mathbf{x}}, \hat{\mathbf{u}})]}{\partial y} = 0 \quad (3.13)$$

where, now φ is only a function of $\hat{\mathbf{x}}$.

Now, let $\mathbf{A} \equiv [A_x \ A_y \ A_t] = \mathbf{A}(\mathbf{x})$ denote the vector potential of the product $(\varphi \hat{\mathbf{u}})$, i.e.:

$$\varphi(\hat{\mathbf{x}}) \hat{\mathbf{u}}(\hat{\mathbf{x}}) = \nabla \times \mathbf{A}(\mathbf{x}) \quad (3.14)$$

By performing a coordinate transformation to a canonical reference frame defined such that $A_y = 0$, \mathbf{A} can be used to relate the two-dimensional time-varying system to the three-dimensional time-invariant form, such that the Hamiltonian functions for the two forms are equivalent [4, 73]. The coordinate transformation is then given by $\mathcal{F} : \hat{\mathbf{x}} \rightarrow \tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}} = [x \ p \ t]^T$, and,

$$p = -A_x[x, y(x, p, t), t] \quad (3.15)$$

The resulting vector potential is $\mathbf{A} = \{A_x[x, y(x, p, t), t] \ 0 \ A_t[x, y(x, p, t), t]\}$, which is governed by

$$\varphi \dot{x} = \frac{\partial A_t}{\partial y}, \quad \varphi \dot{y} = \frac{\partial A_x}{\partial t} - \frac{\partial A_t}{\partial x}, \quad \varphi = -\frac{\partial A_x}{\partial y} \quad (3.16)$$

where, the function $y(x, p, t)$ is implicitly defined in (3.15). Then, the equivalent system is,

$$\frac{d\tilde{\mathbf{x}}}{dt} = \tilde{\mathbf{f}}(\tilde{\mathbf{x}}) = \left[\frac{\partial A_t}{\partial p} \quad -\frac{\partial A_t}{\partial x} \quad 1 \right]^T \quad (3.17)$$

and the time scales in the physical and canonical forms are also equivalent.

Finally, choosing the Hamiltonian function,

$$\psi(x, p, t) = A_t[x, y(x, p, t), t] \quad (3.18)$$

The Hamilton equations in (3.11) are satisfied as follows,

$$\frac{\partial \psi}{\partial x} = -\frac{dp}{dt}, \quad \frac{\partial \psi}{\partial p} = \frac{dx}{dt} \quad (3.19)$$

and are equivalent to a two-dimensional time-varying system in canonical space $\tilde{\mathcal{X}} = \mathcal{F}(\hat{\mathcal{X}})$, with Hamiltonian function ψ . Furthermore, this Hamiltonian formulation is unconditionally valid for any system governed by (3.1) and (3.5), and is mathematically equivalent to Lagrangian fluid transport for unsteady flow in two dimensions, proving the conservative property of (3.5) [73].

DOC Optimality Conditions

4.1 Deterministic Case

The necessary conditions for optimality are derived by using calculus of variations to determine the agent distribution and control laws that minimize the integral cost function (3.10). Since the optimization of (3.10) is subject to a set of dynamic and equality constraints, the integral to be minimized is found by adjoining the dynamic constraints to the cost function (3.10) using a Lagrange multiplier [29]. By this approach, necessary conditions for optimality are found from the first-order effects of control variations that must be zero at all times for the integral cost to be stationary. Then, higher-order sensitivity to control variations can be tested to discriminate between cases in which the integral is a minimum, a maximum, or is neither [29].

From the distributive property of the dot product and by change of sign, the advection equation (3.5) is rewritten as the time-varying equality constraint,

$$-\frac{\partial \varphi}{\partial t} - (\nabla \varphi) \cdot \mathbf{f} - \varphi(\nabla \cdot \mathbf{f}) = 0 \quad (4.1)$$

where the functions' arguments are omitted for brevity. Since (4.1) is a dynamic constraint that must be satisfied at all times, a time-varying Lagrange multiplier, $\lambda[\mathbf{x}, t]$, is used to adjoin the equality constraint (4.1) to the integral cost (3.10). Then, the augmented cost function,

$$\begin{aligned}
J_A &= \phi[\varphi(\mathbf{x}, T_f)] + \int_{T_0}^{T_f} \int_{\mathcal{X}} \left\{ \mathcal{L}(\varphi, \mathbf{u}, t) - \lambda \left[\frac{\partial \varphi}{\partial t} \right. \right. \\
&\quad \left. \left. + (\nabla \varphi) \cdot \mathbf{f} + \varphi(\nabla \cdot \mathbf{f}) \right] \right\} d\mathbf{x}dt
\end{aligned} \tag{4.2}$$

is to be minimized with respect to the functional forms of the time-varying agent distribution φ and control \mathbf{u} , and subject to the equality constraints (3.3),(3.7)-(3.9).

The integrand of (4.2) must satisfy stationarity conditions throughout $(T_0, T_f]$ in order for J_A to be stationary [29]. This is proven by introducing the Hamiltonian,

$$\mathcal{H}[\varphi(\mathbf{x}, t), \mathbf{u}(t), \lambda(\mathbf{x}, t), t] \equiv \mathcal{L}[\cdot] - \lambda[(\nabla \varphi) \cdot \mathbf{f} + \varphi(\nabla \cdot \mathbf{f})] \tag{4.3}$$

which is a function of the agent distribution, the control, and the Lagrange multiplier, and is analogous to the Hamiltonian from Pontryagin's minimum principle [75]. The augmented cost function (4.2) is then re-written in terms of the Hamiltonian and simplified using integration by parts,

$$\begin{aligned}
J_A &= \phi[\cdot] + \int_{T_0}^{T_f} \int_{\mathcal{X}} \left\{ \mathcal{H}[\cdot] - \lambda \frac{\partial \varphi}{\partial t} \right\} d\mathbf{x}dt \\
&= \phi[\cdot] + \int_{\mathcal{X}} [\lambda(T_0)\varphi(T_0) - \lambda(T_f)\varphi(T_f)] d\mathbf{x} \\
&\quad + \int_{T_0}^{T_f} \int_{\mathcal{X}} \left\{ \mathcal{H}[\cdot] + \frac{\partial \lambda}{\partial t} \varphi \right\} d\mathbf{x}dt
\end{aligned} \tag{4.4}$$

By the fundamental theorem of calculus of variations [29], an integral with fixed end points, T_0 and T_f , is stationary for weak variations if the first order effect of

variations in the function, or curve, to be optimized are zero throughout $(T_0, T_f]$. Thus, for J_A to be stationary, the first-order effect of control variations $\delta \mathbf{u}(t)$ on (4.4) must be zero for all $t \in (T_0, T_f]$. By the causality of the macroscopic dynamic equation (3.6), control perturbations lead to perturbations in φ , and thus the first variation of J_A is

$$\begin{aligned} \delta J_A &= \int_{T_0}^{T_f} \int_{\mathcal{X}} \left(\left\{ \frac{\partial \mathcal{H}[\cdot]}{\partial \varphi} - \dot{\lambda} \right\} \delta \varphi(\delta \mathbf{u}) + \frac{\partial \mathcal{H}[\cdot]}{\partial \mathbf{u}} \delta \mathbf{u} \right) d\mathbf{x} dt \\ &+ \left(\left\{ \frac{\partial \phi[\cdot]}{\partial \varphi} - \lambda \right\} \delta \varphi(\delta \mathbf{u}) \right) \Big|_{t=T_f} + [\lambda \delta \varphi(\delta \mathbf{u})] \Big|_{t=T_0} \end{aligned} \quad (4.5)$$

For an extremum, we must have $\delta J_A = 0$ for all $\delta \varphi, \delta \mathbf{u}$, and the variations from $\delta \varphi$ and $\delta \mathbf{u}$ must independently vanish along the optimal solution curve. Thus, the equations,

$$\dot{\lambda} = -\frac{\partial \mathcal{H}[\cdot]}{\partial \varphi} = -\frac{\partial \mathcal{L}[\cdot]}{\partial \varphi} - \lambda(\nabla \cdot \mathbf{f}) \quad (4.6)$$

and,

$$0 = \frac{\partial \mathcal{H}[\cdot]}{\partial \mathbf{u}} = \frac{\partial \mathcal{L}[\cdot]}{\partial \mathbf{u}} + \lambda \left[(\nabla \varphi) \frac{\partial \mathbf{f}}{\partial \mathbf{u}} + \varphi \frac{\partial}{\partial \mathbf{u}} (\nabla \cdot \mathbf{f}) \right] \quad (4.7)$$

must be satisfied for $T_0 \leq t \leq T_f$, subject to the terminal conditions

$$\lambda(T_f) = \left\{ \frac{\partial \phi[\cdot]}{\partial \varphi} \right\} \Big|_{t=T_f} \quad (4.8)$$

Equations (4.6)-(4.8) constitute necessary conditions for optimality for the DOC problem in Chapter 3. Thus, the optimal agent distribution φ^* must satisfy (4.6)-(4.8) along with the normalization condition (3.3), the initial and boundary conditions (3.7)-(3.8), and the state constraint (3.9). If these conditions are satisfied, the extremals can be tested using higher-order variations to verify that they lead

to a minimum of the augmented cost function J_A in (4.2). In particular, sufficient conditions for optimality could be derived from the second-order derivatives of the Hamiltonian (4.3) with respect to \mathbf{u} , or Hessian matrix that is positive definite for a convex Hamiltonian. In this dissertation, we consider admissible solutions of (4.6)-(4.8) to be optimal if perturbations at any $t \in (T_0, T_f]$ only increase the value of J_A .

4.2 Stochastic Case

The optimality conditions for the stochastic case of the DOC problem presented in Section 3 are derived here using calculus of variations. A Lagrange multiplier, $\lambda(\mathbf{x}, t)$, is used to adjoin the dynamic and equality constraints, (3.7)-(3.9), (3.3), to the integral cost function (3.10), obtaining the augmented integral cost function,

$$\begin{aligned}
 J_A = & \int_{\mathcal{X}} \phi \{X(t_f), t_f\} d\mathbf{x} + \int_t \int_{\mathcal{X}} \left\{ \mathcal{L} [X(t), \mathbf{u}(t)] + \right. \\
 & \lambda(\mathbf{x}, t) \left[\frac{\partial \varphi(\mathbf{x}, t)}{\partial t} + \nabla \cdot [\varphi(\mathbf{x}, t) \mathbf{f}(\mathbf{x}, \mathbf{u}, t)] - \right. \\
 & \left. \left. \nu \nabla^2 \varphi(\mathbf{x}, t) \right] \right\} d\mathbf{x} dt. \tag{4.9}
 \end{aligned}$$

Let $\boldsymbol{\xi} = [\mathbf{u}^T, \varphi, \lambda]^T$ denote a vector of variables for the DOC problem, where function arguments are omitted hereon for brevity. The necessary condition for optimality is,

$$\nabla J_A(\mathbf{u}, \varphi, \lambda) = \lim_{\epsilon \rightarrow 0} \frac{J_A(\boldsymbol{\xi} + \epsilon \delta \boldsymbol{\xi}) - J_A(\boldsymbol{\xi})}{\epsilon} = 0, \tag{4.10}$$

where ∇J_A is the gradient of J_A with respect to the variables, \mathbf{u} , φ , λ , and the vector $\epsilon \delta \boldsymbol{\xi} = \epsilon [\delta \mathbf{u}^T, \delta \varphi, \delta \lambda]^T$ contains the variations of the DOC variables.

The variation in the PDF, $\varphi \rightarrow \varphi + \epsilon\delta\varphi$, results in the condition,

$$\begin{aligned} \lim_{\epsilon \rightarrow 0} \frac{J_A(\mathbf{u}, \varphi + \epsilon\delta\varphi, \lambda) - J_A(\mathbf{u}, \varphi, \lambda)}{\epsilon} &= \int_{\mathcal{X}} \left. \frac{\partial \phi}{\partial \varphi} \right|_{t_f} \delta\varphi d\mathbf{x} \\ &+ \int_t \int_{\mathcal{X}} \frac{\partial \mathcal{L}}{\partial \varphi} \delta\varphi + \lambda \left[\frac{\partial \delta\varphi}{\partial t} + \nabla \cdot (\delta\varphi \mathbf{f}) - \nabla^2 \delta\varphi \right] d\mathbf{x} dt = 0. \end{aligned} \quad (4.11)$$

which provides the weak formulation of the DOC optimality conditions. The fundamental theorem of variational calculus (FTVC) is used to arrive at the strong formulation of the DOC optimality conditions. From the FTVC, and integration by parts, the partial derivatives acting on the variations are

$$\begin{aligned} \int_t \int_{\mathcal{X}} \lambda \frac{\partial \delta\varphi}{\partial t} d\mathbf{x} dt &= \\ \int_{\mathcal{X}} \lambda \delta\varphi d\mathbf{x} \Big|_{t_0}^{t_f} - \int_t \int_{\mathcal{X}} \frac{\partial \lambda}{\partial t} \delta\varphi d\mathbf{x} dt, \end{aligned} \quad (4.12)$$

$$\begin{aligned} \int_t \int_{\mathcal{X}} \lambda \nabla \cdot (\delta\varphi \mathbf{f}) d\mathbf{x} dt &= \\ \int_t \int_{\partial \mathcal{X}} \lambda (\mathbf{f} \cdot \hat{\mathbf{n}}) \delta\varphi d\mathbf{x} dt - \int_t \int_{\mathcal{X}} \nabla \lambda \cdot \mathbf{f} \delta\varphi d\mathbf{x} dt, \end{aligned} \quad (4.13)$$

$$\begin{aligned} \int_t \int_{\mathcal{X}} \nu \lambda \nabla^2 \delta\varphi d\mathbf{x} dt &= \\ \int_t \int_{\partial \mathcal{X}} \nu \lambda (\nabla \delta\varphi \cdot \hat{\mathbf{n}}) \delta\varphi d\mathbf{x} dt - \int_t \int_{\mathcal{X}} \nu \nabla \lambda \cdot \nabla \delta\varphi d\mathbf{x} dt = \\ \int_t \int_{\partial \mathcal{X}} \nu \lambda (\nabla \delta\varphi \cdot \hat{\mathbf{n}}) \delta\varphi d\mathbf{x} dt - \int_t \int_{\partial \mathcal{X}} \nu \nabla \lambda \cdot \hat{\mathbf{n}} \delta\varphi d\mathbf{x} dt + \\ \int_t \int_{\mathcal{X}} \nu \nabla^2 \lambda \delta\varphi d\mathbf{x} dt. \end{aligned} \quad (4.14)$$

Because an initial condition for φ is given at t_0 , as shown in (3.7), the initial

variation in the PDF is $\delta\varphi|_{t_0} = 0$, and (4.12) simplifies to

$$\begin{aligned} & \int_t \int_{\mathcal{X}} \lambda \frac{\partial \delta\varphi}{\partial t} d\mathbf{x} dt = \\ & \int_{\mathcal{X}} \lambda \delta\varphi d\mathbf{x} \Big|_{t_f} - \int_t \int_{\mathcal{X}} \frac{\partial \lambda}{\partial t} \delta\varphi d\mathbf{x} dt. \end{aligned} \quad (4.15)$$

The boundary condition (3.8) implies that (4.14) simplifies to

$$\begin{aligned} & \int_t \int_{\mathcal{X}} \nu \lambda \nabla^2 \delta\varphi d\mathbf{x} dt = \\ & - \int_t \int_{\partial\mathcal{X}} \nu \nabla \lambda \cdot \hat{\mathbf{n}} \delta\varphi d\mathbf{x} dt + \int_t \int_{\mathcal{X}} \nu \nabla^2 \lambda \delta\varphi d\mathbf{x} dt. \end{aligned} \quad (4.16)$$

Then, by substituting the results in (4.13), (4.15), and (4.16) into (4.11), and grouping like terms, the variation in (4.11) can be written as

$$\begin{aligned} 0 = & \int_{\mathcal{X}} \left(\frac{\partial \phi}{\partial \varphi} + \lambda \right) \delta\varphi \Big|_{t_f} d\mathbf{x} + \\ & \int_t \int_{\partial\mathcal{X}} (\lambda(\mathbf{f} \cdot \hat{\mathbf{n}}) + \nu \nabla \lambda \cdot \hat{\mathbf{n}}) \delta\varphi d\mathbf{x} dt + \\ & \int_t \int_{\mathcal{X}} \left(\frac{\partial \mathcal{L}}{\partial \varphi} - \frac{\partial \lambda}{\partial t} - \nabla \lambda \cdot \mathbf{f} - \nu \nabla^2 \lambda \right) \delta\varphi d\mathbf{x} dt. \end{aligned} \quad (4.17)$$

By the FTVC, the variation in (4.17) can be written as the adjoint PDE:

$$\frac{\partial \lambda}{\partial t} = \frac{\partial \mathcal{L}}{\partial \varphi} - \nabla \lambda \cdot \mathbf{f} - \nu \nabla^2 \lambda \quad (4.18)$$

$$\text{SJT: } \lambda(\mathbf{x}, t_f) = - \frac{\partial \phi}{\partial \varphi} \Big|_{t_f} \quad \mathbf{x} \in \mathcal{X},$$

$$\lambda(\mathbf{f} \cdot \hat{\mathbf{n}}) + \nu(\nabla \lambda) \cdot \hat{\mathbf{n}} = 0 \quad \mathbf{x} \in \partial\mathcal{X}$$

The variation in the control law, $\mathbf{u} \rightarrow \mathbf{u} + \epsilon \delta \mathbf{u}$,

$$\begin{aligned} \lim_{\epsilon \rightarrow 0} \frac{J_A(\mathbf{u} + \epsilon \delta \mathbf{u}) - J_A(\mathbf{u})}{\epsilon} = & \quad (4.19) \\ \int_t \int_{\mathcal{X}} \frac{\partial \mathcal{L}}{\partial \mathbf{u}} + \lambda \left[\nabla \cdot \left(\varphi \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \delta \mathbf{u} \right) \right] d\mathbf{x} dt = & \\ \int_t \int_{\mathcal{X}} \frac{\partial \mathcal{L}}{\partial \mathbf{u}} - \nabla \lambda \cdot \left(\varphi \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right) \delta \mathbf{u} d\mathbf{x} dt + & \\ \int_t \int_{\partial \mathcal{X}} \lambda \left(\varphi \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \cdot \hat{\mathbf{n}} \right) \delta \mathbf{u} d\mathbf{x} dt. & \end{aligned}$$

must equal zero for optimality, by the FTVC, i.e.:

$$0 = \frac{\partial \mathcal{L}}{\partial \mathbf{u}} - \nabla \lambda \cdot \left(\varphi \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right). \quad (4.20)$$

Finally, the variation in the Lagrange multiplier, $\lambda \rightarrow \lambda + \epsilon \delta \lambda$, leads to the macroscopic state equation. Thus, the DOC optimality conditions are given by the set of PDEs:

$$\frac{\partial \varphi}{\partial t} = -\nabla \cdot (\varphi \mathbf{f}) + \nu \nabla^2 \varphi \quad (4.21)$$

$$\text{SJT: } \varphi(\mathbf{x}, t_0) = p(\mathbf{x}) \quad \mathbf{x} \in \mathcal{X},$$

$$\nabla \varphi \cdot \hat{\mathbf{n}} = 0 \quad \mathbf{x} \in \partial \mathcal{X}$$

$$\frac{\partial \lambda}{\partial t} = \frac{\partial \mathcal{L}}{\partial \varphi} - \nabla \lambda \cdot \mathbf{f} - \nu \nabla^2 \lambda \quad (4.22)$$

$$\text{SJT: } \lambda(\mathbf{x}, t_f) = -\frac{\partial \phi}{\partial \varphi} \Big|_{t_f} \quad \mathbf{x} \in \mathcal{X},$$

$$\lambda = 0 \quad \mathbf{x} \in \partial \mathcal{X}$$

$$0 = \frac{\partial \mathcal{L}}{\partial \mathbf{u}} - \nabla \lambda \cdot \left(\varphi \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right). \quad (4.23)$$

The macroscopic state (4.21) and adjoint (4.22) equations are parabolic PDEs. The control equation (4.23) is an algebraic equation relating the optimal \mathbf{u} to φ and

λ . If (4.21)-(4.23) are satisfied, then the resulting φ and \mathbf{u} are the optimal control and resulting agent distribution for the macroscopic control problem. To obtain the sufficient conditions for optimality, the second-order variations of J_A may be tested to verify that these values in fact are at an extremal that is a minimum of J , but in this paper, the solutions are considered to be optimal if any perturbations only increase the value of J . Chapter 6 presents a generalized reduced gradient (GRG) method to solve the optimality conditions to determine optimal DOC trajectories.

Probability Density Function Approach to Path Planning

5.1 Microscopic Feedback Control Laws

The microscopic control laws are determined from the optimal macroscopic description φ^* by defining an attractive potential that pulls the agents toward φ^* . Since φ^* is a time-varying distribution, the potential function is defined as a quadratic function of the error between φ^* and the estimated agent distribution, $\hat{\varphi}$, at time $(t + \delta t)$:

$$U_{att} \triangleq \frac{1}{2} \{\hat{\varphi}[\mathbf{x}, t + \delta t] - \varphi^*[\mathbf{x}, t + \delta t]\}^2 \quad (5.1)$$

The time interval δt is a small time constant that is chosen to prevent the agents from lagging behind φ^* . The estimate $\hat{\varphi}[\mathbf{x}, t + \delta t]$ is computed by stepping the macroscopic dynamic equation (3.5) forward in time by an interval δt from $\hat{\varphi}[\mathbf{x}, t]$, and $\hat{\varphi}[\mathbf{x}, t]$ is computed via kernel density estimation (KDE) from the agents' positions at time t . KDE is a non-parametric technique used to estimate the PDF of a random variable by representing it as a summation of kernel functions [71]. For simplicity, the kernel

function was selected here as the standard multivariate Gaussian kernel,

$$\kappa(\mathbf{x}) = \frac{1}{2\pi} \exp\left(\frac{-1}{2} \mathbf{x}^T \mathbf{x}\right) \quad (5.2)$$

Figure 5.1 illustrates an example attractive potential function, U_{att} , where KDE is used to approximate $\hat{\phi}$.

To prevent the sensors from colliding with obstacles a repulsive potential is also introduced and defined as,

$$U_{rep}(\mathbf{x}) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(\mathbf{x})} - \frac{1}{\rho_0}\right)^2 & \text{if } \rho(\mathbf{x}) \leq \rho_0 \\ 0 & \text{if } \rho(\mathbf{x}) > \rho_0 \end{cases} \quad (5.3)$$

where, $\rho(\mathbf{x})$ is the Euclidean distance from \mathbf{x} to the nearest obstacle space \mathcal{CB} ,

$$\rho(\mathbf{x}) = \min_{\mathbf{x}' \in \mathcal{CB}} \|\mathbf{x} - \mathbf{x}'\|, \quad \mathcal{CB} \equiv \cup_{i=1}^N \mathcal{CB}_i \quad (5.4)$$

$\eta > 0$ is a scaling factor, and $\rho_0 > 0$ is a distance-of-influence parameter that is chosen by the user. The total potential field can then be defined as a weighted sum of the attractive and repulsive potentials,

$$U(\mathbf{x}) = w_a U_{att}(\mathbf{x}) + w_r U_{rep}(\mathbf{x}) \quad (5.5)$$

where $w_a > 0$ and $w_r > 0$ are user-defined weighting terms. Then, a microscopic control law that minimizes (5.5) can be obtained that returns agent trajectories that follow the negative gradient of U , based on the detailed equation (3.1), such that $\mathbf{u}^* = \mathbf{c}[\phi^*(\mathbf{x}, t + \delta t)]$.

5.1.1 Decentralization of Agent Density Estimation

Kernel density estimation is a well-known non-parametric approach for estimating the probability density function (PDF) from which a set of independent and identically distributed data samples were taken. Given a data set $\mathbf{y}_j, j = 1, \dots, N_y, \mathbf{y}_j \in \mathfrak{R}^d$

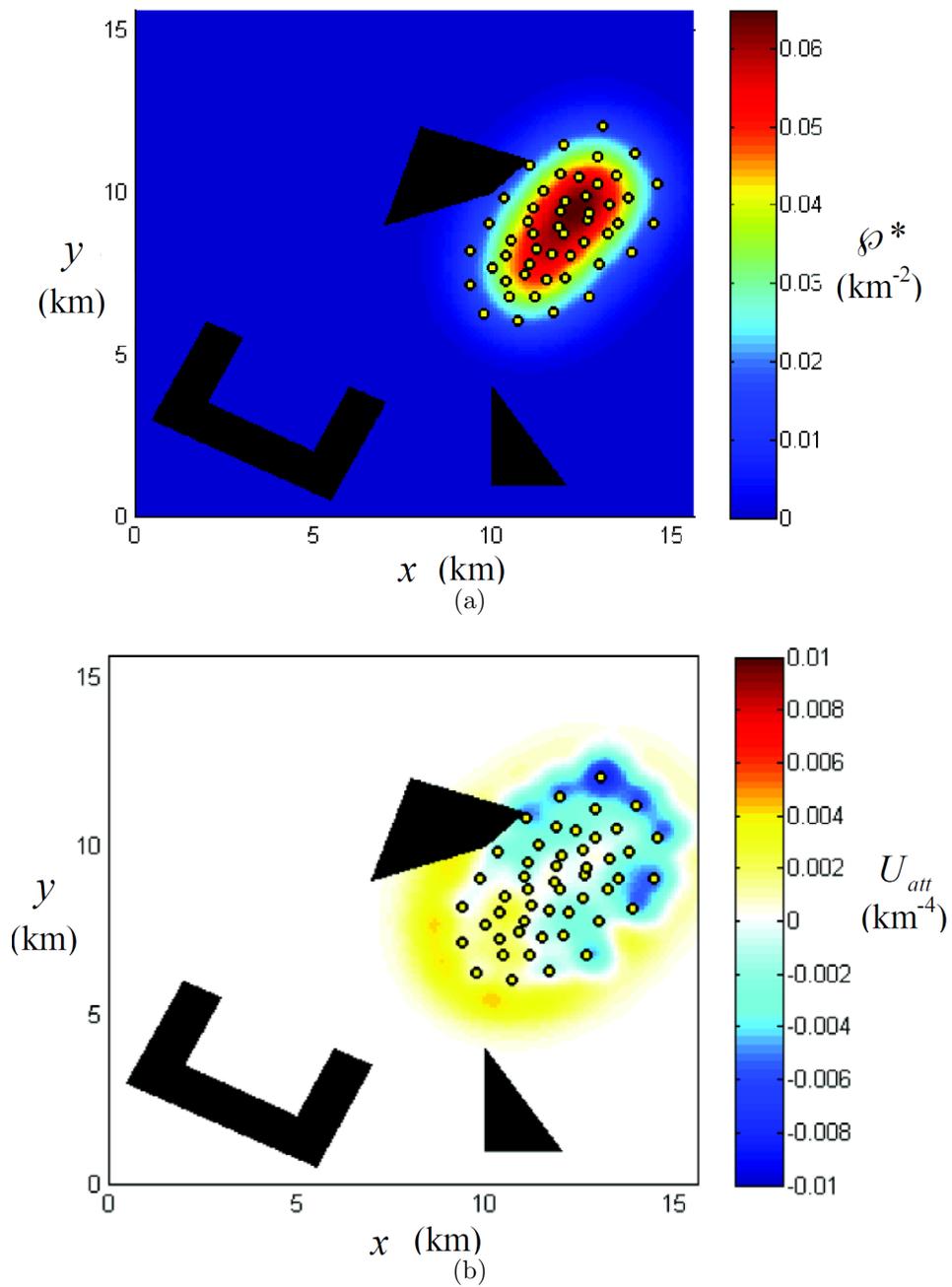


FIGURE 5.1: Example of attractive potential based on difference between optimal PDF and actual sensor density. Yellow circles represent individual sensor positions, and solid black shapes represent geometric obstacles. (a) Optimal distribution ϕ^* plotted on background. (b) Attractive potential function U_{att} plotted on background.

that is assumed to be from some unknown PDF f , the kernel density estimation takes the form [68],

$$\hat{f} = \sum_{j=1}^{N_y} w_j K_{\mathbf{H}_j}(\mathbf{y} - \mathbf{y}_j) \quad (5.6)$$

where $w_j, j = 1, \dots, N_y$ is the weighting coefficients satisfying the condition $\sum_{j=1}^{N_y} w_j = 1$, and the j^{th} kernel centered at \mathbf{y}_j is defined as,

$$K_{\mathbf{H}_j}(\mathbf{y} - \mathbf{y}_j) = |\mathbf{H}_j|^{-\frac{1}{2}} K(\mathbf{H}_j^{-\frac{1}{2}}(\mathbf{y} - \mathbf{y}_j)) \quad (5.7)$$

The kernel function K is a user-defined d -variate non-negative symmetric real-value function [68]. The band-width matrix \mathbf{H}_j is a parameter that controls the smoothing of the KDE algorithm, and it must be positive definite and symmetric. With appropriate parameter choices, KDE has been shown to be an effective method for estimating the underlying PDF in many cases and often only requires a few samples to give adequate results [68]. However the general KDE algorithm described above requires centralized processing due to the summation, which might not be feasible in a distributed agent network.

Alternatively a distributed KDE algorithm based on information spreading can be used that does not require centralized processing and is asymptotically consistent with the centralized version in cases where the network is fully connected [36]. The primary difference is that the distributed KDE algorithm uses an information sharing protocol to exchange kernel information between sensors incrementally until a complete and accurate approximation of the global KDE is achieved by each agent. It has been shown that as long as the network is fully connected, the connectivity structure will only affect the convergence speed and will not worsen the estimation accuracy [51]. Therefore, KDE can be performed in a distributed manner with accuracies that are nearly identical to the centralized method and with the only requirement being the full connectivity of the network.

Each agent maintains a local estimation of the agent distribution, governed by a stored kernel set $\mathbf{S}_i = \{ \langle w_{i,k}, \mathbf{x}_{i,k}, \mathbf{H}_{i,k} \rangle, k = 1, \dots, N_i \}$, where $\mathbf{x}_{i,k}$ denotes the position of agent k perceived by the i^{th} agent, N_i is the number of kernels stored by agent i , and $\mathbf{H}_{i,k}$ and $w_{i,k}$ are the bandwidth matrix and weighting coefficient of the k^{th} kernel stored by agent i . At time t_0 , the kernel set of each agent only contains the kernel generated using its own position. The agents also maintain a neighbor set, where the i^{th} agent's neighbors are defined as any agents located within the distance of a communication radius r . Then through an information spreading process, the agents choose a random neighbor and compare their kernel sets with one another. If an agent sees that its neighbor has newer or previously unknown kernel information, they will save the information to their own stored kernel set. Then a new random neighbor will be chosen, and the process repeats.

In practice, the information communicated would include the sensors' positions and kernel parameters to construct the kernels and the corresponding sensors' indices and positional measurement timestamps to enable the overwriting of old data. Note that for many homogeneous agent networks (networks with identical sensors), the bandwidth matrices $\mathbf{H}_{i,k}$ and weighting parameters $w_{i,k}$ may be defined to be consistent across the network, making their communication unnecessary and reducing communication requirements. For simplicity, in this paper the bandwidth matrix is defined as $\mathbf{H}_{i,k} = c\mathbf{I}_2 \forall i, k$, where c is a constant and \mathbf{I}_2 is the two-dimensional identity matrix, and the weighting parameters are calculated as $w_{i,k} = \frac{1}{N_i} \forall i, k$. Then the purpose of the communications essentially becomes to give each agent full positional knowledge of all agents in the network.

Using their known sets of agent positions, each agent can then generate the corresponding kernels and combine them to obtain a local estimation of the PDF. For simplicity, the standard two-dimensional Gaussian kernel function is chosen in

this paper and defined as,

$$K(\mathbf{x}) = \frac{1}{2\pi} e^{-\frac{1}{2}\mathbf{x}^T\mathbf{x}} \quad (5.8)$$

which is used to construct the kernels as follows,

$$K_{\mathbf{H}_{i,k}}(\mathbf{x} - \mathbf{x}_i) = |\mathbf{H}_{i,k}|^{-\frac{1}{2}} K(\mathbf{H}_{i,k}^{-\frac{1}{2}}(\mathbf{x} - \mathbf{x}_i)) \quad (5.9)$$

Then the local estimation of the PDF can be calculated by each agent as,

$$\hat{f}_i = \sum_{k=1}^{N_i} w_{i,k} K_{\mathbf{H}_{i,k}}(\mathbf{x} - \mathbf{x}_{i,k}) \quad (5.10)$$

Once the i^{th} agent has a local density estimation, the potential function (5.1) can be computed with $\hat{\phi} \approx \hat{f}_i$. Then the feedback control law is constructed from the negative gradient of the potential function $-\nabla U_i$. Therefore by using the decentralized kernel density estimation approach, each agent in the network can derive its own feedback control law without centralized network computations.

Numerical Solutions to DOC Problem

6.1 Solution via Direct Optimization

The DOC problem presented in Chapter 3 is solved using a direct method inspired by direct methods traditionally used to solve classical optimal control problems numerically [14]. The direct method presented in this paper discretizes the continuous DOC problem about a finite set of collocation points, and then transcribes it into a finite-dimensional nonlinear program (NLP) that can be solved using a sequential quadratic programming (SQP) algorithm.

By representing φ as a parametric approximation, such as Gaussian mixture models [28] or harmonic functions [54, 25], the size of the optimization problem can be reduced, and a functional form of φ can be obtained. To reduce complexity, a finite Gaussian mixture model is used in this paper to provide a parametric approximation of φ^* obtained from the superposition of z components with Gaussian PDFs f_1, \dots, f_z , and corresponding mixing proportions or *weights* w_1, \dots, w_z . The n -dimensional multivariate Gaussian PDF,

$$f_j[\mathbf{x}, t] = \frac{1}{(2\pi)^{n/2} |\boldsymbol{\Sigma}_j|^{1/2}} e^{[-(1/2)(\mathbf{x}-\boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\mathbf{x}-\boldsymbol{\mu}_j)]} \quad (6.1)$$

is referred to as the *component density* of the mixture, and is characterized by a time-varying mean vector $\boldsymbol{\mu}_j \in \mathbb{R}^n$, and a time-varying covariance matrix $\boldsymbol{\Sigma}_j \in \mathbb{R}^{n \times n}$, with $j = 1, \dots, z$. We assume that, at any $t \in (T_0, T_f]$, the agent distribution can be represented as follows,

$$\wp(\mathbf{x}, t) = \sum_{j=1}^z w_j(t) f_j[\mathbf{x}, t] \quad (6.2)$$

where, $0 \leq w_j \leq 1 \forall j$, and $\sum_{j=1}^z w_j = 1$ [56]. In this paper, it is assumed that z is fixed and is chosen by the user, based on problem complexity. Then, an optimal agent distribution \wp^* can be obtained by determining the optimal trajectories of the mixture model parameters from the DOC problem, where the mixture model parameters to be optimized over time are the weights w_j , the elements of $\boldsymbol{\mu}_j$, and the covariances in $\boldsymbol{\Sigma}_j$, with $j = 1, \dots, z$. In addition to satisfying the DOC constraints and optimality conditions, the mixture model parameters must be determined such that the component densities f_1, \dots, f_z are nonnegative and obey the normalization condition (3.3) for all $t \in (T_0, T_f]$.

The optimal trajectories of the mixture model parameters can be obtained by discretizing the continuous DOC problem in state space and time about a finite set of collocation points in $\mathcal{X} \times (T_0, T_f]$ and formulating it as a finite-dimensional NLP. Let Δt denote a constant discretization time interval, and k denote the discrete time index, such that $\Delta t = (T_f - T_0)/K$, and thus $t_k = k\Delta t$, for $k = 0, \dots, K$. It is assumed that the microscopic control inputs, \mathbf{u} , are piecewise-constant during every time interval, and that,

$$\begin{aligned} \wp_k &= \wp[\mathbf{x}, t_k] = \sum_{j=1}^z w_j(t_k) f_j[\mathbf{x}, t_k] \\ &\equiv \sum_{j=1}^z w_{jk} \frac{1}{(2\pi)^{n/2} |\boldsymbol{\Sigma}_{jk}|^{1/2}} e^{-(1/2)(\mathbf{x} - \boldsymbol{\mu}_{jk})^T \boldsymbol{\Sigma}_{jk}^{-1} (\mathbf{x} - \boldsymbol{\mu}_{jk})} \end{aligned} \quad (6.3)$$

represents the agent distribution at t_k . Then the set of weights $\{w_{jk}\}$, the elements of $\boldsymbol{\mu}_{jk}$, and $\boldsymbol{\Sigma}_{jk} \forall j, k$ are all grouped into a vector $\boldsymbol{\zeta}$ that represents the trajectories of the mixture model parameters in discrete time.

Since \wp is a conserved quantity of a Hamiltonian system (Section 3.2), the evolution equation (3.5) can be discretized using a conservative numerical scheme. The FV scheme is used for this paper because it does not suffer from dissipative error when using a coarse-grained state discretization, as is often favorable to save computational cost [77]. The FV approach partitions the state space \mathcal{X} into FVs defined by a constant discretization interval $\Delta \mathbf{x} \in \mathbb{R}^n$, and each centered about a collocation point $\mathbf{x}_l \in \mathcal{X} \subset \mathbb{R}^n$, with $l = 1, \dots, X$. Let $\wp_{l,k}$ and $\mathbf{u}_{l,k}$ denote the finite-difference approximations of $\wp(\mathbf{x}_l, t_k)$ and $\mathbf{c}[\wp(\mathbf{x}_l, t_k)]$, respectively. Then, the finite-difference approximation of the evolution equation (3.5) is obtained by applying the divergence theorem to (3.5) for every FV, such that, $\wp_{k+1} = \wp_k + \Delta t \rho_k$, where,

$$\rho_k = - \int_S [\wp_k \mathbf{f}(\wp_{l,k}, \mathbf{u}_{l,k}, t_k)] \cdot \hat{\mathbf{n}} dS \quad (6.4)$$

and S and $\hat{\mathbf{n}}$ denote the FV boundary and unit normal, respectively. To ensure numerical stability, the discretization intervals Δt and $\Delta \mathbf{x}$ are chosen to satisfy the Courant-Friedrichs-Lewy condition [77].

Then, letting $\Delta \mathbf{x}_{(j)}$ denote the j^{th} element of $\Delta \mathbf{x}$, the discretized DOC problem

can be written as the finite-dimensional NLP,

$$\begin{aligned}
\min J_D &= \sum_{j=1}^n \Delta \mathbf{x}^{(j)} \sum_{l=1}^X [\phi_{l,K} + \Delta t \sum_{k=1}^K \mathcal{L}(\wp_{l,k}, \mathbf{u}_{l,k}, t_k)] \\
\text{sbj to } &\wp_{k+1} - \wp_k - \Delta t \rho_k = 0, \quad k = 1, \dots, K \\
&\sum_{j=1}^n \Delta \mathbf{x}^{(j)} \sum_{l=1}^X \wp_{l,k} - 1 = 0, \quad k = 1, \dots, K \\
&\wp_{l,0} = g_0(\mathbf{x}_l), \quad \forall \mathbf{x}_l \in \mathcal{X} \\
&\wp_{l,k} = 0, \quad \forall \mathbf{x}_l \in \partial \mathcal{X}, \quad k = 1, \dots, K
\end{aligned} \tag{6.5}$$

where $\phi_{l,K} \triangleq \phi(\wp_{l,K})$ is the terminal constraint.

From (6.3) it can be seen that $\wp_{l,k}$ and $\mathbf{u}_{l,k}$ are functions solely of the mixture model parameters ζ , which constitute the NLP variables. Also, since \wp is modeled by a Gaussian mixture, the state constraint (3.9) is always satisfied and needs not be included in the constraints. The solution ζ^* of the NLP in (6.5) is obtained using an SQP algorithm that solves the Karush-Kuhn-Tucker (KKT) optimality conditions by representing (6.5) as a sequence of unconstrained quadratic programming (QP) subproblems with objective function $J_S(\zeta) = J_D(\zeta) + \sum_j \lambda_j \xi_j(\zeta)$, where ξ_j denotes the j^{th} constraint in (6.5), and λ_j denotes a vector of multipliers of proper dimensions.

At every major iteration ℓ of the SQP algorithm, the Hessian matrix $\mathbf{H} = \partial J_S / \partial \zeta$ is approximated using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method

$$\mathbf{H}_{\ell+1} = \mathbf{H}_\ell + \frac{\mathbf{q}_\ell \mathbf{q}_\ell^T}{\mathbf{q}_\ell^T \Delta \zeta_\ell} - \frac{\mathbf{H}_\ell^T \Delta \zeta_\ell^T \Delta \zeta_\ell \mathbf{H}_\ell}{\Delta \zeta_\ell^T \mathbf{H}_\ell \Delta \zeta_\ell} \tag{6.6}$$

Where $\Delta \zeta_\ell = \zeta_\ell - \zeta_{\ell-1}$, and \mathbf{q}_ℓ is the change in the gradient $\nabla J_S = \partial J_S / \partial \zeta$ at the ℓ^{th} iteration [62]. The Hessian approximation (6.6) is then used to generate a QP

subproblem,

$$\begin{aligned} \min \quad & h(\mathbf{d}_\ell) = (1/2) \mathbf{d}_\ell^T \mathbf{H}_\ell \mathbf{d}_\ell + \nabla J_S^T \mathbf{d}_\ell \\ \text{sbj to} \quad & \nabla \xi_j^T \mathbf{d}_\ell + \xi_j = \mathbf{0}, \quad \forall j \end{aligned} \quad (6.7)$$

in \mathbf{d}_ℓ , the search direction. The optimal search direction \mathbf{d}_ℓ^* is computed from the above QP using an off-the-shelf QP solver [55], such that $\zeta_{\ell+1} = \zeta_\ell + \alpha_\ell \mathbf{d}_\ell^*$.

The step-length α_ℓ is determined by an approximate line search in the direction \mathbf{d}_ℓ^* , aimed at producing a sufficient decrease in the merit function,

$$\Psi(\zeta_\ell) = J(\zeta_\ell) + \sum_j \mathbf{r}_{\ell,j}^T \xi_j(\zeta_\ell) \quad (6.8)$$

based on the Armijo condition, and a penalty parameter $\mathbf{r}_{\ell,j}$ defined in [62]. The algorithm terminates when the KKT conditions are satisfied within a desired tolerance.

6.2 Solution via Indirect Method

The DOC optimality conditions (4.21)-(4.23) consist of a coupled set of parabolic PDEs. Because analytical solutions to these PDEs are not available, this section presents a GRG approach for computing the numerical solution of the DOC optimality conditions. The approach exploits the causality of the macroscopic dynamic equation (3.4) to represent J_A solely as a function of \mathbf{u} . Then an extremum of the DOC problem (3.1)-(3.4) can be found by determining the parameters of the control laws (6.9) that satisfy the optimality conditions.

GRG methods improve iteratively upon the approximation of the optimal control law and of the macroscopic state and Lagrange multiplier by holding the other fixed during each update. During every iteration of the GRG algorithm, the latest approximation of $\mathbf{u}^* = \mathbf{c}^*[\mathbf{x}(t), t]$, in parameterized form (6.9) is used to solve macroscopic state and adjoint PDEs, (4.21) and (4.22), to obtain an approximation

for φ^* and λ^* . Subsequently, holding the approximations of φ^* and λ^* fixed, the approximation for \mathbf{u}^* is updated so as to minimize (3.10), and satisfy the third and final optimality condition. This process is repeated until the norm of the gradient is below a user-defined tolerance or any update to \mathbf{u}^* causes an increase in J .

The GRG method falls under a larger class of optimization techniques referred to as *Nested Analysis and Design* (NAND). In NAND approaches, the gradient is obtained at each iteration of the optimization by eliminating the state and co-state variables by solving the PDEs using a numerical algorithm, and only the control is considered [15]. Alternatively, a *Simultaneous Analysis and Design* (SAND), or full space, optimization strategy could be used in which the optimization over the state, co-state, and control are preformed simultaneously. However, it has been shown that SAND methods are often very ill-conditioned, where the individual PDEs in the NAND techniques are typically better conditioned [16].

Algorithm 1 GRG Optimality Solver

```

initialize  $\alpha_{j,k}(t)$ 
while  $\|\mathbf{g}\| > \text{TOL}$  do
     $\tilde{\varphi} \leftarrow$  solve macroscopic state PDE ( $\mathbf{u}$ )
     $\tilde{\lambda} \leftarrow$  solve adjoint PDE ( $\tilde{\varphi}, \mathbf{u}$ )
    for all  $\ell$  do
         $\mathbf{g}_\ell \leftarrow$  compute gradient ( $\tilde{\varphi}, \tilde{\lambda}, \mathbf{u}$ )
    end for
    for all  $j, k$  do
         $\alpha_{j,k} \leftarrow$  update  $\alpha_{j,k}$  ( $J, \mathbf{g}$ )
    end for
end while

```

To obtain a closed form representation of the control for all \mathbf{x} , the j^{th} element of the control vector, u_j , can be parameterized as the sum of M linearly-independent basis functions $\phi_1(\cdot), \dots, \phi_M(\cdot)$, such that

$$u_j = \sum_k \phi_k(\mathbf{x}) \alpha_{j,k}(t), \quad \text{for } j = 1, \dots, M. \quad (6.9)$$

Then, the goal of the indirect method is to obtain the parameters, $\alpha_{j,k}^*(t)$, that

minimize the cost function (3.10), subject to the aforementioned constraints. As shown in Section 4.2, since the macroscopic state, \wp , and the Lagrangian multiplier, λ , can be found explicitly as a function of \mathbf{u} , a generalized reduced gradient (GRG) method [76] can be used to determine the optimal parameters of the control law (6.9).

An analytical representation of the gradient of the cost function J , denoted by \mathbf{g} , with respect to the controls \mathbf{u} can be found, thereby circumventing the need for finite difference to approximate the gradient, greatly reducing the computational requirements. The gradient of J is calculated as follows. Let $\tilde{\wp}$ and $\tilde{\lambda}$ satisfy (4.21) and (4.22), respectively, for a given \mathbf{u} . Then the gradient is given by

$$\begin{aligned} \nabla_{\mathbf{u}} J &= \nabla_{\mathbf{u}} J_A \Big|_{\tilde{\wp}, \tilde{\lambda}} = \int_{\mathcal{X}} \frac{\partial \phi}{\partial \wp} \nabla_{\mathbf{u}} \wp \delta \mathbf{u} \Big|_{\tilde{\wp}, t_f} d\mathbf{x} + \\ &\int_t \int_{\mathcal{X}} \left\{ \frac{\partial \mathcal{L}}{\partial \mathbf{u}} \delta \mathbf{u} + \frac{\partial \mathcal{L}}{\partial \wp} \nabla_{\mathbf{u}} \wp \delta \mathbf{u} + \nabla_{\mathbf{u}} \lambda \delta \mathbf{u} \left[\frac{\partial \wp}{\partial t} + \nabla \cdot (\wp \mathbf{f}) \right] + \right. \\ &\lambda \left[\frac{\partial}{\partial t} (\nabla_{\mathbf{u}} \wp \delta \mathbf{u}) + \nabla \cdot \left(\nabla_{\mathbf{u}} \wp \mathbf{f} \delta \mathbf{u} + \wp \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \delta \mathbf{u} \right) \right] \\ &\left. - \nu \nabla^2 \nabla_{\mathbf{u}} \wp \delta \mathbf{u} \right\} \Big|_{\tilde{\wp}, \tilde{\lambda}} d\mathbf{x} dt \end{aligned} \quad (6.10)$$

Performing integration by parts and recalling that $\tilde{\wp}$ and $\tilde{\lambda}$ were defined to satisfy (4.21) and (4.22), equation (6.10) becomes

$$\nabla_{\mathbf{u}} J = \int_t \int_{\mathcal{X}} \left[\frac{\partial \mathcal{L}}{\partial \mathbf{u}} - \nabla \tilde{\lambda} \cdot \left(\tilde{\wp} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right) \right] \delta \mathbf{u} d\mathbf{x} dt. \quad (6.11)$$

Let the time be discretized into Q equally spaced points, $t_q = t_0 + q\Delta t$, where $q = 0, \dots, Q$, and $\Delta t = (t_f - t_0)/Q$. Then from (6.11) it follows that

$$\frac{\partial J}{\partial \alpha_{j,k}} \Big|_{t=t_q} \approx \Delta t \int_{\mathcal{X}} \left[\frac{\partial \mathcal{L}}{\partial u_j} - \nabla \tilde{\lambda} \cdot \left(\tilde{\wp} \frac{\partial \mathbf{f}}{\partial u_j} \right) \right] \Big|_{t=t_q} \phi_k d\mathbf{x}. \quad (6.12)$$

The previous equation gives the gradient of the cost function with respect to the parameters that determine the control \mathbf{u} . Using this expression of the gradient, \mathbf{u} can be updated using one of many gradient-based optimization schemes, such as the Sequential Quadratic Programming (SQP) method described in Section 6.1. The algorithm for solving the optimality conditions is then given in Algorithm 1.

6.3 Complexity Analysis

The computational complexities of the DOC methods presented in the previous sections are compared to that of a classical direct method for optimal control taken from [14]. The classical direct method was also used in [8] to optimize the track coverage of a mobile sensor network. The results in [8] showed that the applicability of classical optimal control is limited by the network size, and that the classical direct method can optimize the track coverage of networks with up to $N \approx 20$. The approach obtains an NLP representation of the classical optimal control problem for N -coupled ODEs in the form (3.1), and an integral cost function of the microscopic sensors state and control vectors. Subsequently, the NLP solution can be obtained using an SQP algorithm with the computational complexity shown in Table 6.1.

Table 6.1: Computational Complexity of SQP Solution

	Direct DOC	Indirect DOC	Classical OC
Hessian update	$O(zXK^2)$	$O(MXK)$	$O(nmN^2K^2)$
QP subproblem	$O(z^2XK^3)$	$O(M^2XK^3)$	$O(nm^2N^3K^3)$
Line search	$O(XK)$	$O(XK)$	$O(nNK)$

The Hessian update (6.6), the solution of the QP subproblem (6.7), and the line-search minimization of the merit function (6.8) are the most computationally-expensive steps of the SQP algorithm described in Section 6.1. The computational complexity of these three steps is shown in Table 6.1 for the classical optimal control (OC), the direct DOC, and the indirect DOC methods. For all methods, the QP

subproblem is the dominant computation, which is carried out by a QR decomposition of the active constraints using Householder Triangularization [62]. Then, the computation required by the classical OC method exhibits cubic growth with respect to K and N , and becomes prohibitive for $N \gg 1$. The computation required by the DOC methods exhibits cubic growth only with respect to K , and quadratic growth with respect to z or M . Thus, for systems with $X \ll nN$ and $z \ll mN$ (or $M \ll mN$), the DOC approach can bring about considerable computational savings.

Unlike the direct methods, the indirect method requires the solution of two PDEs, (4.21) and (4.22), at each step of the iterative optimization process. Thus, the computational time depends upon the number of computations required for the quasi-newton and line search methods, as well as the number of computations required to numerically solve the PDEs (4.21) and (4.22). However, if it is assumed that the solutions to (4.21) and (4.22), φ and λ , have been obtained, and that the integration within the gradient (6.12) can be accurately approximated in $O(X)$ computations, then for the indirect method, the Hessian update requires $O(MXK)$ computations. If it is further assumed that $z \approx M$, or that the number of parameters required to approximate φ^* sufficiently in the direct method is approximately equal to the number of parameters needed to adequately approximate \mathbf{u}^* in the indirect method, then the indirect approach provides a significant reduction in computations required in the Hessian update step of the optimization process, as it is linear in the number of temporal collocation points, K , whereas the direct methods have quadratic dependence on K .

Planning Optimal Paths through Obstacle-Populated Regions via DOC

This chapter addresses the problem of planning optimal trajectories of large groups of agents through obstacle-populated regions to goal configurations. With existing approaches, such as optimal control and roadmap techniques, optimizing the paths of a coupled multi-agent system is classified as PSPACE-hard and is intractable due to the computational complexity associated with a large number of agents. Other methods solve the problem by decoupling the system or relaxing the global optimality conditions. By doing so, the solutions sacrifice performance and completeness. The DOC methodology presented in this dissertation can determine near-optimal agent trajectories by optimizing the system's macroscopic performance subject to the agents' dynamics, which allows systems to plan paths for many agents that achieve objectives such as minimizing travel distances and energy consumed, avoiding collisions with obstacles, and maintaining a specific formation for the agent distribution to maintain, in addition to reaching a given goal configuration.

7.1 Problem Formulation

The DOC problem and optimality conditions presented in the previous section are demonstrated through a multi-agent trajectory optimization problem. Consider a system of N cooperative unicycle robots traveling through an obstacle-populated compact space $\mathcal{W} \subset \mathbb{R}^2$, referred to as the *workspace*, and occupied by M obstacles $\mathcal{B}_1, \dots, \mathcal{B}_M$, where $\mathcal{B}_j \subset \mathcal{W}$. The dynamics of each robot are described by the nonlinear unicycle model,

$$\dot{x}_i = v_i \cos \theta_i \quad \dot{y}_i = v_i \sin \theta_i \quad \dot{\theta}_i = \omega_i \quad (7.1)$$

where $\mathbf{q}_i = [x_i \ y_i \ \theta_i]^T$ is the configuration of agent i , which contains the xy -coordinates, x_i and y_i , and heading angle, θ_i , with $i = 1, \dots, N$. The microscopic control vector of agent i is $\mathbf{u}_i = [v_i \ \omega_i]^T$, where v_i and ω_i are the linear and angular velocities, respectively.

The macroscopic state of the system is described by the time-varying PDF, or restriction operator, $\varphi : \mathcal{X} \times \mathbb{R} \rightarrow \mathbb{R}$, such that the probability of $\mathbf{x}_i = [x_i \ y_i]^T$ is given by (3.2), in terms of φ . It follows that $\mathcal{W} = \mathcal{X}$, and φ can be regarded as the density of agents in \mathcal{W} at time $t \in (T_0, T_f]$. Given an initial distribution $\varphi_0(\mathbf{x})$, the agents must travel in \mathcal{W} to meet a goal distribution $g(\mathbf{x})$, while avoiding obstacles, and minimizing energy consumption. The goal distribution is assumed to be time-invariant, and all M obstacles' positions and geometries are assumed known without error. This section shows that all of these trajectory optimization objectives can be expressed in terms of the PDF, φ , to be optimized.

A measure of the difference between φ and the goal distribution, g , is given by the instantaneous Kullback-Leibler (KL) divergence at time t ,

$$D(\varphi \parallel g) = \int_{\mathcal{X}} \varphi(\mathbf{x}, t) \log_2 \frac{\varphi(\mathbf{x}, t)}{g(\mathbf{x})} d\mathbf{x} \quad (7.2)$$

where, by definition, the support set of φ is contained by the support set of g , and the value $0 \log_2(0/0)$ is replaced with 0 for continuity [23]. Although the KL divergence is not a true distance function because it is not symmetric, it is a suitable objective function because its value increases when the difference between φ and g increases, and vice versa. Also, the KL divergence of φ and g is zero when the two distributions are equal.

A repulsive potential U_{rep} can be generated from the obstacles' geometries $\mathcal{B}_1, \dots, \mathcal{B}_M$ in \mathcal{W} , as shown in [44]. Then, the obstacle avoidance objective can be represented by the product φU_{rep} . The energy consumption is modeled as a quadratic function of the control. The DOC cost function to be minimized is,

$$J = \int_{T_0}^{T_f} \left[w_d D(\varphi || g) + \int_{\mathcal{W}} \left(w_r \varphi U_{rep} + w_e \mathbf{u}^T \mathbf{R} \mathbf{u} \right) d\mathbf{x} \right] dt$$

where, \mathbf{R} is a diagonal positive-definite matrix. The scalar weights w_d , w_r , and w_e can be chosen by the user or from a Pareto optimization curve, and represent the desired tradeoff between the three competing objectives. By this formulation of the cost function, the KL divergence of φ and g is minimized throughout $(T_0, T_f]$.

The solution of the DOC problem can be approached by a parametrization technique that approximates the function to be optimized by a weighted linear combination of basis functions [50, 79]. Finite Gaussian mixture models are commonly used to provide parametric approximations of PDFs. Thus, in this chapter, the agent distribution is approximated by a mixture model comprised of z components with Gaussian PDFs f_1, \dots, f_z , and corresponding mixing proportions (or weights) w_1, \dots, w_z . The n -dimensional multivariate Gaussian PDF,

$$f_j(\mathbf{x}, t) = \frac{e^{\{-(1/2)[\mathbf{x}-\boldsymbol{\mu}_j(t)]^T \boldsymbol{\Sigma}_j(t)^{-1} [\mathbf{x}-\boldsymbol{\mu}_j(t)]\}}}{(2\pi)^{n/2} |\boldsymbol{\Sigma}_j(t)|^{1/2}} \quad (7.3)$$

is referred to as the component density of the mixture, and is characterized by a

time-varying mean vector $\boldsymbol{\mu}_j \in \mathbb{R}^n$, and a time-varying covariance matrix $\boldsymbol{\Sigma}_j \in \mathbb{R}^{n \times n}$, with $j = 1, \dots, z$. We assume that, at any $t \in (T_0, T_f]$, the agent distribution can be approximated as follows,

$$\wp(\mathbf{x}, t) \approx \sum_{j=1}^z w_j(t) f_j(\mathbf{x}, t) \quad (7.4)$$

where, $0 \leq w_j \leq 1$ for any j , and $\sum_{j=1}^z w_j = 1$ [56]. In this chapter, it is assumed that z is fixed, and chosen by the user. Then, an approximately-optimal agent distribution \wp^* can be obtained by determining the optimal trajectories of the mixture model parameters, i.e., $\boldsymbol{\mu}_j^*$, $\boldsymbol{\Sigma}_j^*$, and w_j^* , for $j = 1, \dots, z$.

In addition to satisfying the DOC constraints and optimality conditions, the mixture model parameters must be determined such that the component densities f_1, \dots, f_z are nonnegative and obey the normalization condition for all $t \in (T_0, T_f]$. This is accomplished by discretizing the continuous DOC problem in space and time, about a finite set of collocation points in $\mathcal{X} \times (T_0, T_f]$. Let $\Delta \mathbf{x}$ and Δt denote constant space and time discretization intervals, respectively, that, to guarantee numerical stability, are chosen according to the Courant-Friedrichs-Lewy condition [77]. Then, by formulating the discretized DOC problem as a finite dimensional NLP, the optimal mixture model parameters can be computed via sequential quadratic programming (SQP) [13], as detailed in Chapter 6.

Once an optimal agent distribution \wp^* is obtained from the DOC problem (7.1)-(7.3), the microscopic control laws are obtained from the negative gradient of the potential function (5.1). For robots described by the unicycle model (7.1), the microscopic control law is,

$$\mathbf{u}_i = [v_c \quad Q(\hat{\theta}_i, -\nabla U)]^T \quad (7.5)$$

where,

$$Q(\cdot) = \{a(\hat{\theta}_i) - a[\Theta(-\nabla U)]\} \text{sgn}\{a[\Theta(-\nabla U)] - a(\hat{\theta}_i)\}$$

is the minimum differential between the agent’s actual heading angle $\hat{\theta}_i$ and the desired heading angle $\Theta(-\nabla U)$, v_c is the agent’s speed, $\text{sgn}(\cdot)$ is the sign function, and $a(\cdot)$ is an angle wrapping function [44].

7.2 Numerical Simulations and Results

7.2.1 *Optimal Agent Trajectories computed with DOC Approach*

The DOC solution of the multi-agent trajectory optimization problem presented in Chapter 3 is illustrated through an example in which $N = 500$ agents with unicycle dynamics (7.1) must travel from the initial distribution, g_0 , to the goal distribution, g , plotted in Fig. 7.1, during a time interval $(0, 22]$ hr. The initial microscopic states \mathbf{x}_{i_0} are determined by sampling g_0 . Subsequently, the agents must travel in a workspace $\mathcal{W} = [0, L] \times [0, L]$, with $L = 15$ km, and three obstacles plotted in solid black in Fig. 7.1. All agents are assumed to have a linear velocity $v_i = 0.7$ km/hr, and an angular velocity $\omega_i \in [-\omega_{max}, +\omega_{max}]$, where $\omega_{max} = 0.52$ rad/s. The cost function weights, $w_d = 20$, $w_r = 0.1$, and $w_e = 1$, are chosen based on the units and relative magnitudes of the three navigation objectives.

The computational complexity of the optimization performed in this example is of the order of the dominant computation of the algorithm’s quadratic program (QP) subproblem, which is a QR decomposition using Householder Triangularization [62]. This leads to a complexity of $O(z^2 X K^3)$ that does not grow with the number of agents N . The number of mixture components, $z = 6$, is chosen to obtain the best tradeoff between accuracy and computational complexity. Time is discretized in intervals of $\Delta t = 1$ hr, such that $K = 22$, and the state space is discretized using $X = 900$ collocation points. As a result, the optimal agent distribution could be computed in several hours on a Core-2 Duo CPU 2.13-GHz computer with 8-GB RAM, while the corresponding classical optimal control problem for $N = 500$ was found to be intractable on the same machine.

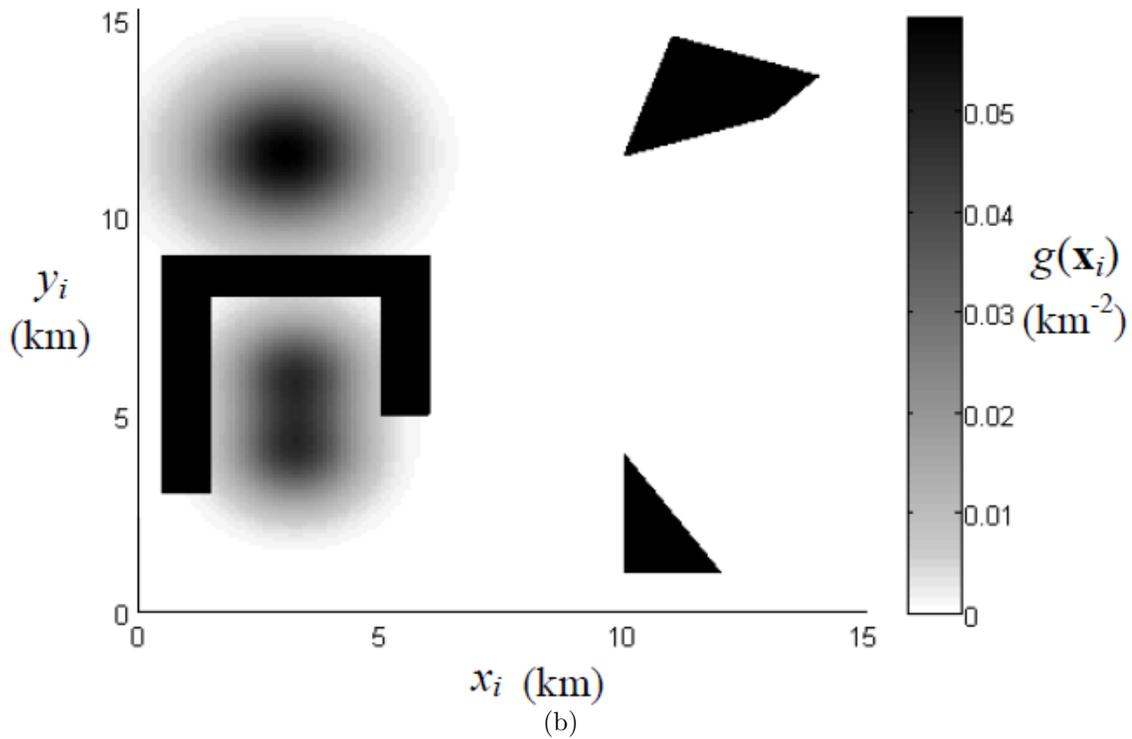
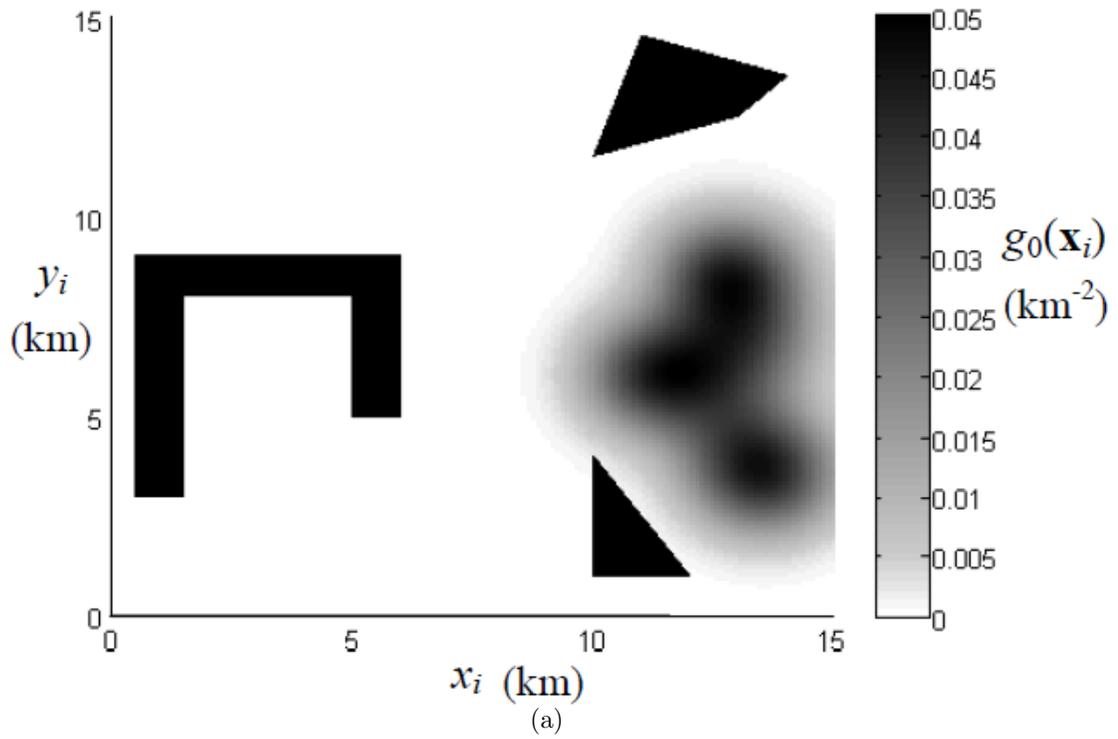


FIGURE 7.1: Initial (a) and goal (b) agent distributions for a workspace with three obstacles (solid black).

The optimal agent distribution, φ^* , and the values of the agents' microscopic state variables, \mathbf{x}_i , are plotted in Fig. 7.2 at four sample moments in time, $t = 5$ hr (a), $t = 10$ hr (b), $t = 15$ hr (c), and $t = 22$ hr (d). The evolution of the microscopic state, \mathbf{x}_i , is simulated by integrating the closed-loop detailed equation (7.1) numerically for all i , using a time interval $\delta t = 3$ s. At every time step of the numerical integration, the feedback control law is evaluated according to (7.5), from the attractive potential (5.1) defined in terms of the optimal distribution φ^* . The time-histories of the DOC microscopic state and control for three, randomly-chosen agents are plotted in Fig. 7.3, and the state trajectories of six, randomly-chosen agents are plotted in Fig. 7.2(d). The results show that, as specified by the cost function (7.3), over time φ^* meets the goal distribution g , while agents also avoid obstacles in \mathcal{W} , and minimize energy consumption.

The optimal agent distributions obtained via SQP are also used to show that any perturbations from the optimal mixture model parameters increase the error in the optimality conditions derived in Section 4.1. Figure 7.4 shows the effects of perturbations in the covariances of two mixture components at $t = 21$ hr, for the optimal distribution in Fig. 7.2. Here, the j^{th} component's covariance is modified such that $\Sigma_j = \Sigma_j^* + c_j I_2$, where c_j is the perturbation parameter varied in Fig. 7.4, and e_1 and e_2 denote the mean-squared errors for the optimality conditions (4.6) and (4.7), respectively. These results are representative of an extensive set of simulations in which the means, covariances, and component weights were perturbed from optimal at various times. In all cases, the optimality conditions were validated numerically by showing that e_1 and e_2 were at a minimum for the mixture model parameters ζ^* computed via SQP.

It is also simple to add additional objectives to the optimization problem. For example, we can force the distribution to maintain specific coupled formations by modifying the Lagrangian. This is illustrated with a distribution of $m = 3$ mixture

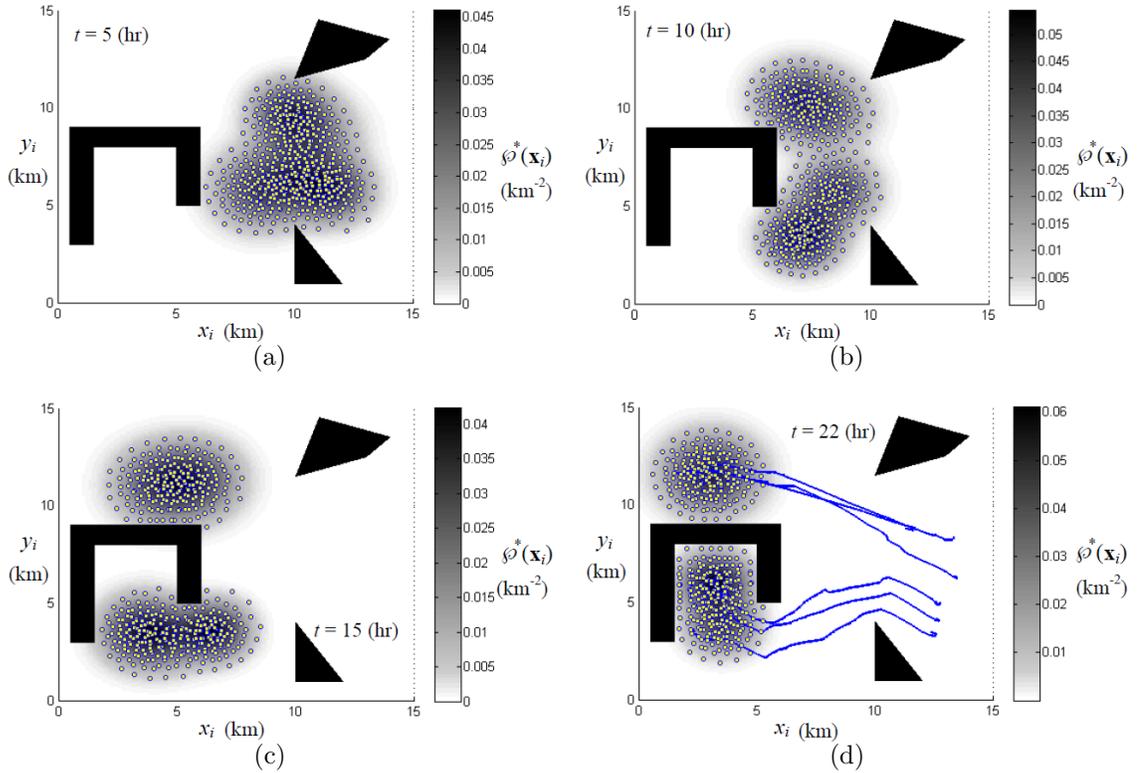


FIGURE 7.2: Optimal evolution of agent distribution and microscopic state (yellow circles) for $N = 500$ microscopic agents, at four instants in time. State trajectories (blue lines) of 6 randomly chosen agents are plotted in (d).

components and an added objective of holding the pattern of an equilateral triangle with a constant edge length, $a = 2$. To achieve this, we introduce a new term,

$$S(\varphi(\mathbf{x}, t)) = \sum_{m=1, m \neq \ell}^M \left| \|\mu_m - \mu_\ell\| - a \right| \quad (7.6)$$

that gives a penalty when the distribution differs from the desired configuration and is minimized when the pattern is matched, where μ_m is the mean of the m^{th} mixture component, and $|\cdot|$ represents absolute value. Then $S[\varphi(\mathbf{x}, t)]$ and a new weighting

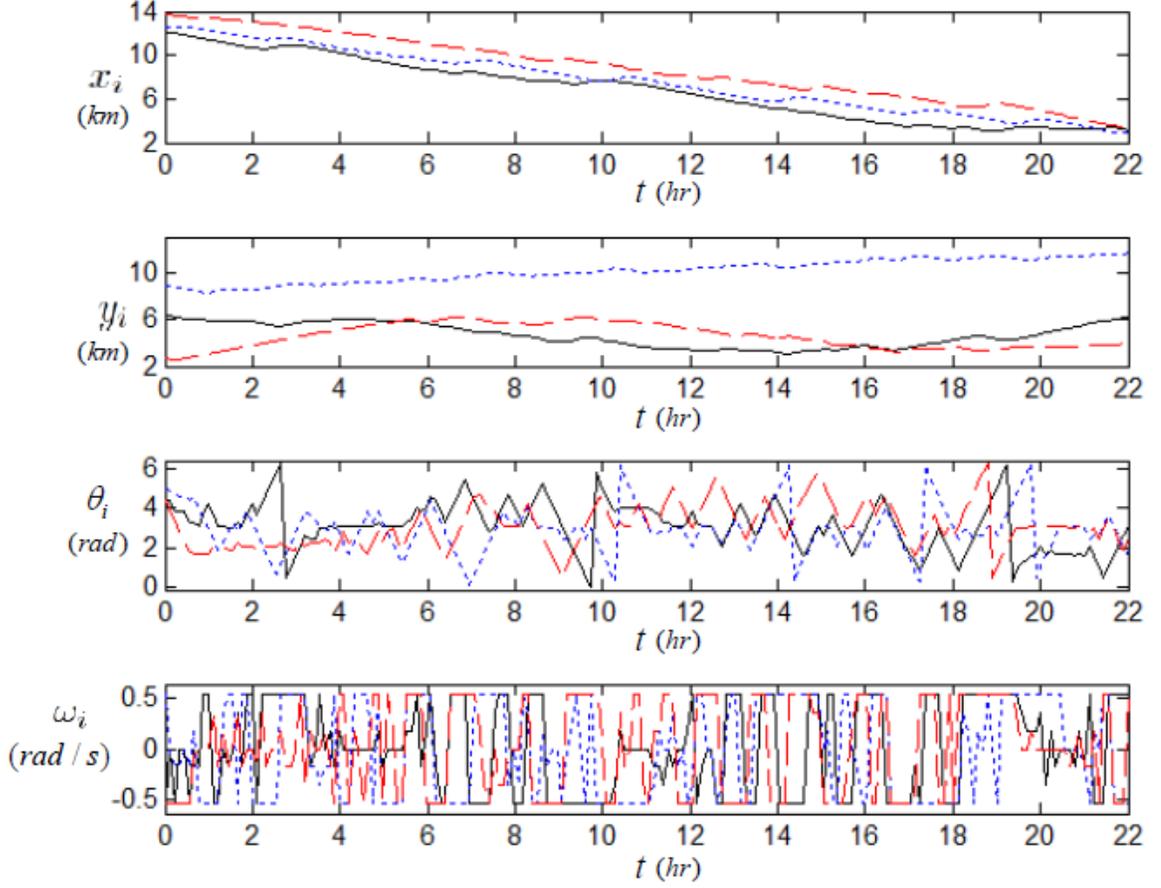


FIGURE 7.3: Microscopic state and control histories for 3 agents randomly chosen from the example in Fig. 7.2.

constant, w_S , can be appended to the Lagrangian,

$$\begin{aligned}
 \mathcal{L}[\varphi(\mathbf{x}, t), \mathbf{u}(t), t] &= w_D D_\alpha[\varphi(\mathbf{x}, t) \| g(\mathbf{x}, t)] & (7.7) \\
 &+ w_S S[\varphi(\mathbf{x}, t)] \\
 &+ \int_{\mathcal{A}} [w_p \varphi(\mathbf{x}, t) U_{rep}(\mathbf{x}) \\
 &+ w_e \mathbf{u}^T R \mathbf{u}] d\mathbf{x}
 \end{aligned}$$

The optimization can then be performed with the same numerical method as described above. The initial and target distributions shown in Figure 7.2.1 are given, and like the previous examples, the objectives are to match the target distribution

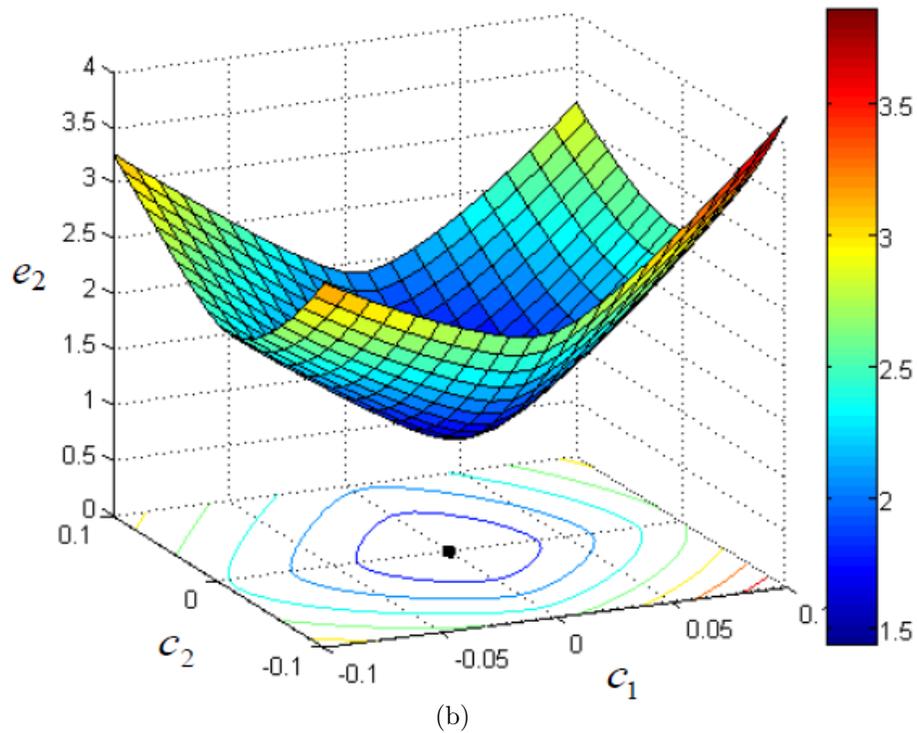
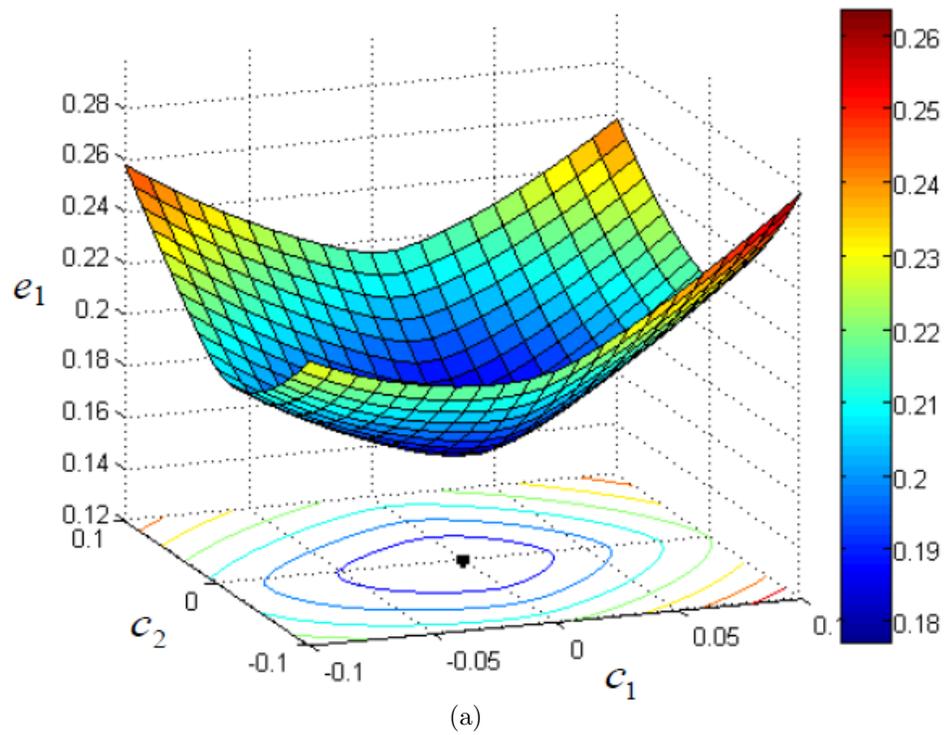


FIGURE 7.4: Numerical error for optimality conditions (4.6) (a) and (4.7) (b) as a function of covariance perturbation parameters.

while avoiding obstacles and minimizing energy consumed. The optimized PDF evolution is displayed in Figure 7.2.1. It can be seen that the distribution quickly converges to the desired pattern and maintains the shape throughout the time interval.

7.2.2 Environmental Effects on Dynamics

The DOC approach can also be applied to problems where the environmental effects in the workspace can affect the agent dynamics. In this case, we consider an additive velocity effect that can be thought of as similar to ocean currents or wind, such that the agents' velocities in inertial frame is $\dot{\mathbf{q}}' = \dot{\mathbf{q}} + \boldsymbol{\nu}(\mathbf{q})$, where $\boldsymbol{\nu}(\mathbf{q}) \in \mathbb{R}^2$ is the local current velocity vector at position \mathbf{q} . In practice, environmental effects can be modeled using environmental forecasts with assimilated data [9, 2, 37, 47], to exploit the natural dynamics for the transport of agents and minimize the energy required.

For simplicity, consider microscopic agent dynamics in the body coordinate frame (fixed to the vehicle) that are modeled by a single integrator model for a point robot that was modified from the model proposed in [85],

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} + \sigma \mathbf{I}_2 \begin{bmatrix} \eta_x \\ \eta_y \end{bmatrix} \quad (7.8)$$

where $\mathbf{q}_i = [x_i \ y_i]^T$ denotes the configuration vector of the i^{th} agent, and x_i and y_i are the xy -coordinates. The microscopic control vector of the i^{th} agent is $\mathbf{u}_i = [v_{x_i} \ v_{y_i}]^T$, where v_{x_i} and v_{y_i} are linear velocities in the x and y directions, respectively. The disturbance vector is $\mathbf{w}_i = [\eta_x \ \eta_y]^T$, where η_x and η_y are independent random variables with values given by standard Gaussian processes, σ is a constant, and \mathbf{I}_2 is the identity matrix. The agents exist in a workspace $\mathcal{W} = [0, L_x] \times [0, L_y]$, $L_x = 20$ km and $L_y = 16$ km, over a time interval $(t_0, t_f]$, where $t_0 = 0$ and $t_f = 15$ hr. The workspace is occupied by M obstacles $\mathcal{B}_1, \dots, \mathcal{B}_M$, where $\mathcal{B}_j \subset \mathcal{W}$. A static current velocity field, $\boldsymbol{\nu}(\mathbf{q})$, is assumed to be known and is plotted in Figure 7.7. The agents

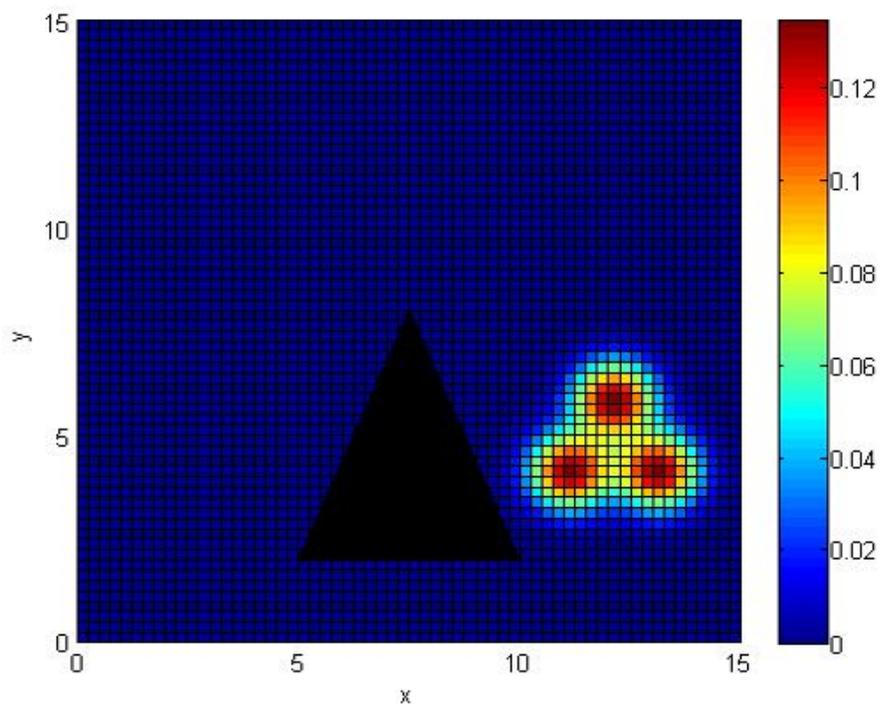
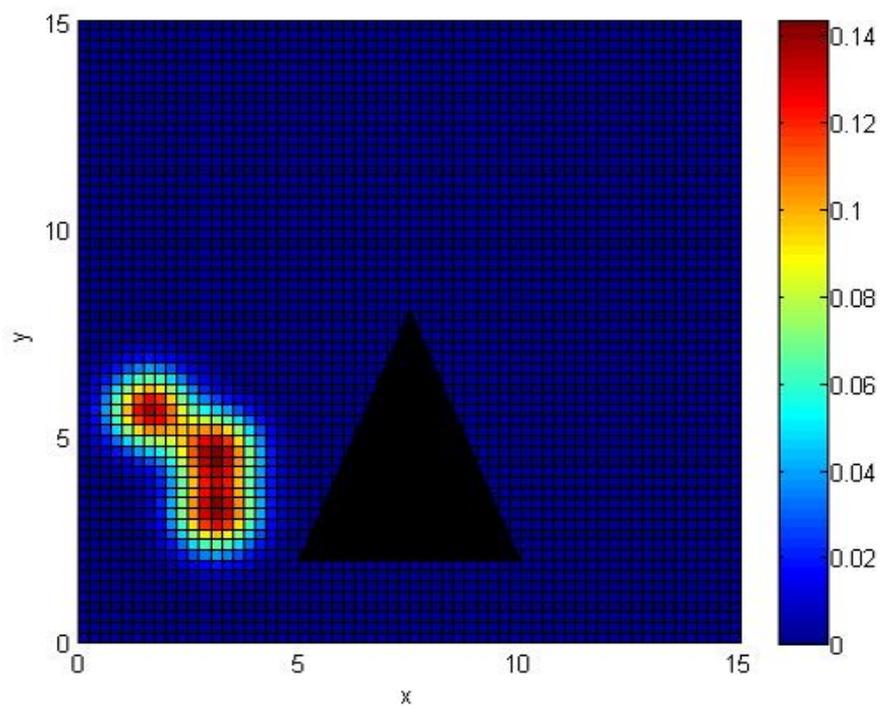


FIGURE 7.5: (a) Initial agent distribution. (b) Target agent distribution

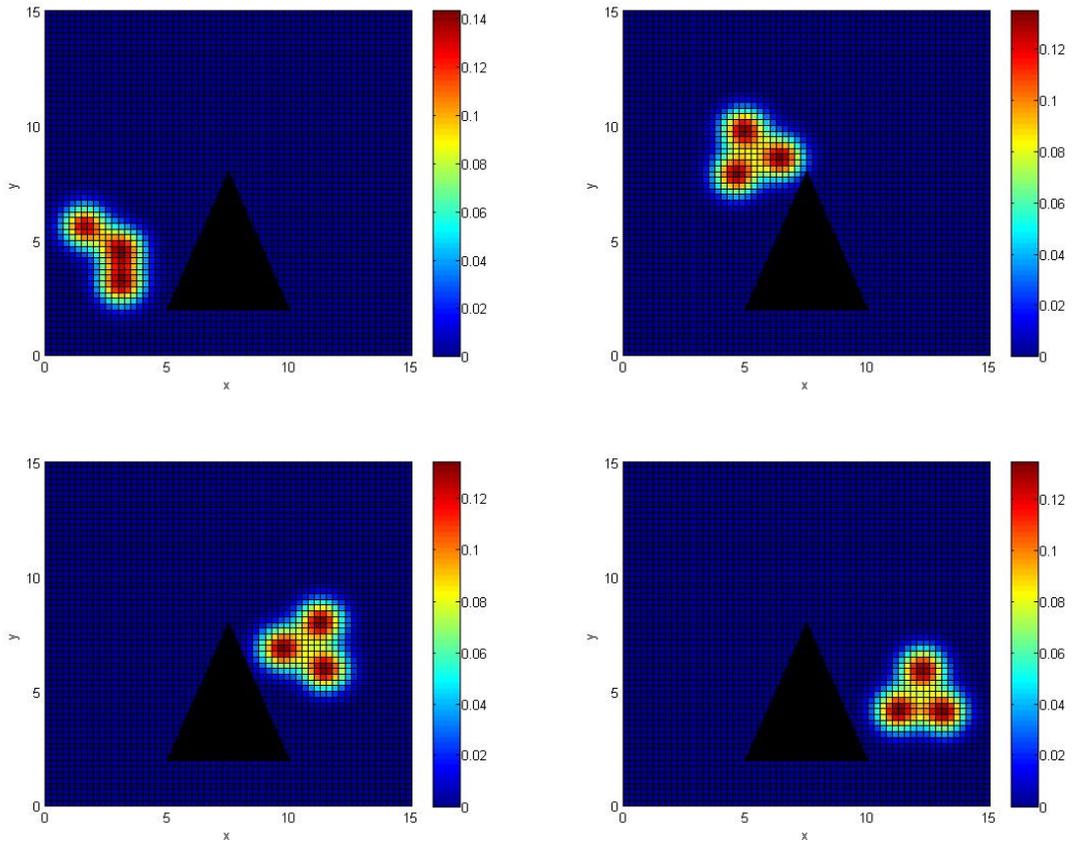


FIGURE 7.6: Optimal solution for evolving agent distribution with objective to maintain triangular pattern. (a) $t = 0$, (b) $t = 8$, (c) $t = 15$, (d) $t = 21$

have a given initial distribution φ_0 shown in Figure 7.2.2, and the initial microscopic states and sampled from φ_0 . The system objectives are again to match a goal agent distribution, g , plotted in Figure 7.2.2, avoid the geometric obstacles, and minimize energy consumed. The objective function (7.3) is used.

The optimal state and control trajectories are computed using the DOC approach with the direct optimization algorithm, described in Section 6.1, for the case described both with and without the added environmental effects. A comparison is presented here to demonstrate how the DOC approach can consider and exploit the environment when calculating optimal trajectories. The optimal evolution of the

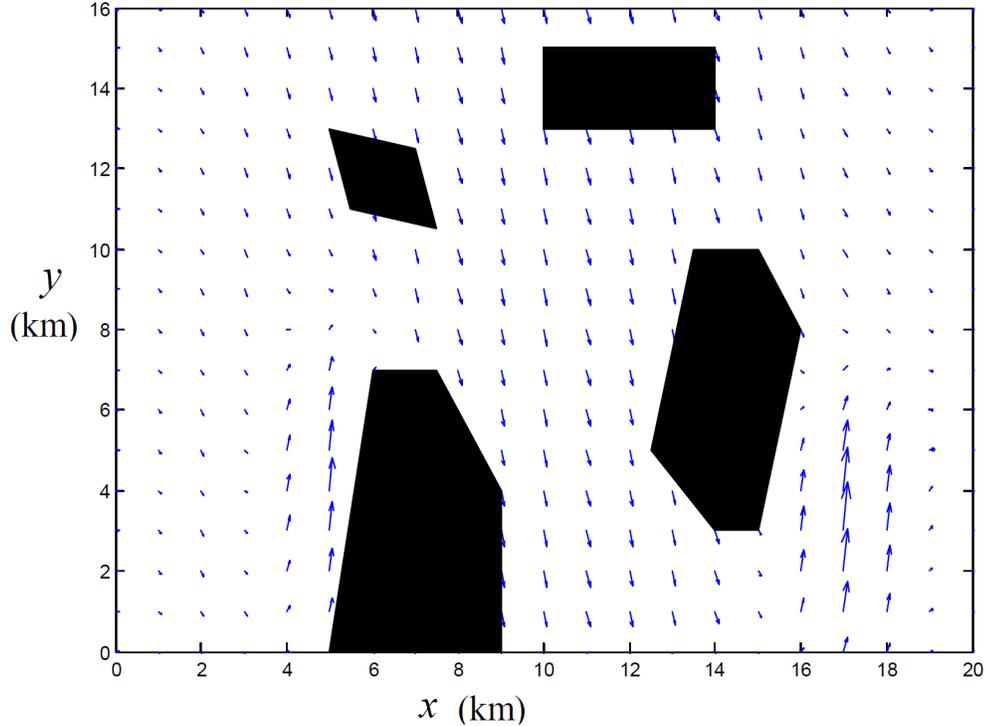


FIGURE 7.7: Current velocity field and obstacles in workspace.

agent PDF for the case without currents is plotted in Figure 7.2.2, and the same solution is shown in Figure 7.10 with the microscopic states of $N = 500$ agents that are control by the DOC method. The same is done for the case with currents, where the optimal time-varying agent PDF is shown in Figure 7.11, and the solution with the resulting $N = 500$ agent states and current vector field superimposed is presented in Figure 7.12.

The two solutions are seen to differ significantly, where in the case without currents, most agents travel above the right-most obstacle since that path offers the shortest, and consequently, optimal distance to the goal distribution. When the environmental effects are added, since the current velocities are generally in the direction of the path below the right-most obstacle, the agents can save significant energy costs by exploiting the currents along the path. Therefore, the path below the obstacle is optimal for more agents, and most agents are shown to travel along

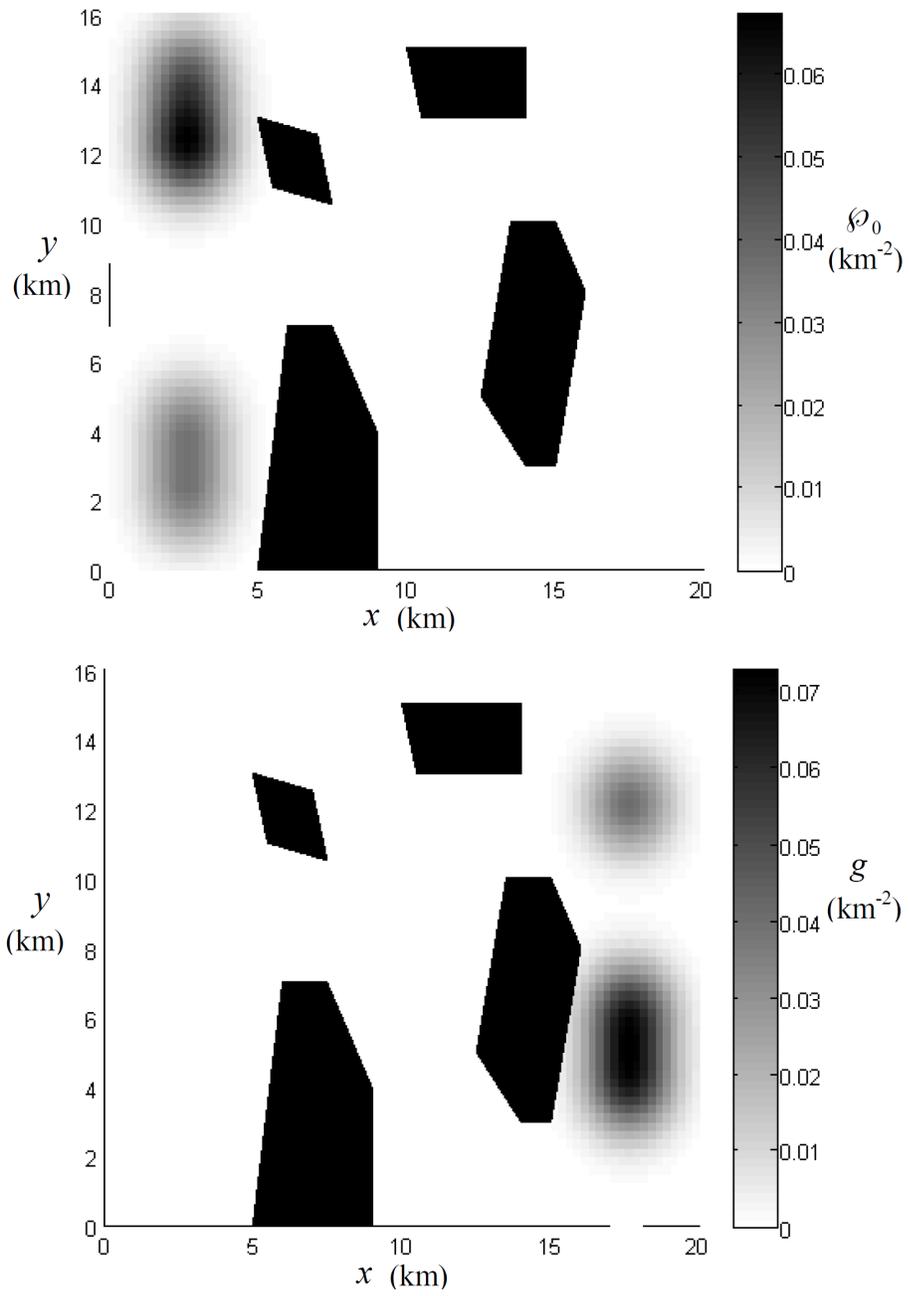


FIGURE 7.8: (a) Initial agent distribution. (b) Target agent distribution

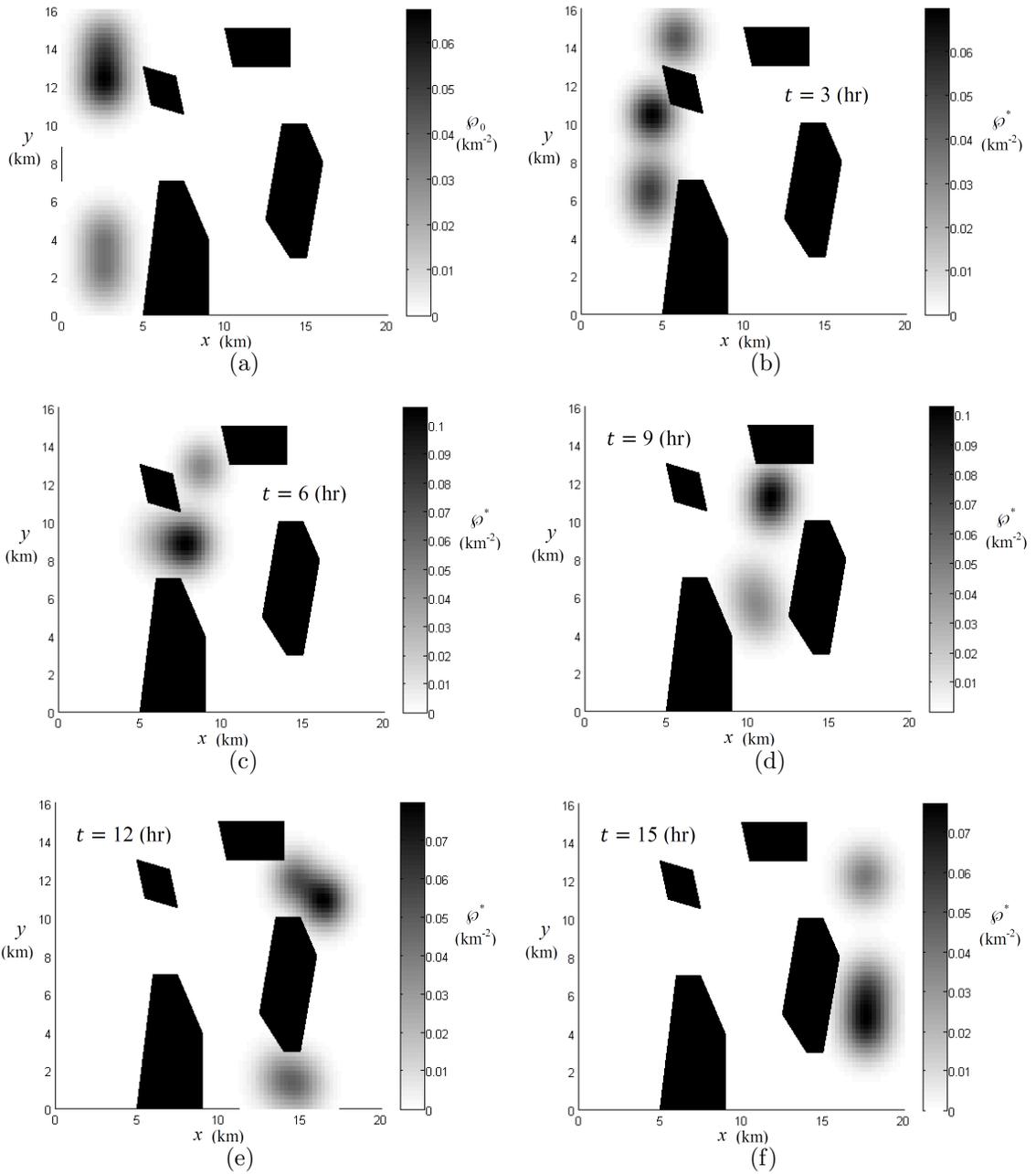


FIGURE 7.9: The evolution of the optimal agent PDF ϕ^* for the case with *no currents* is computed with the *direct* DOC algorithm and plotted at six instants in time.

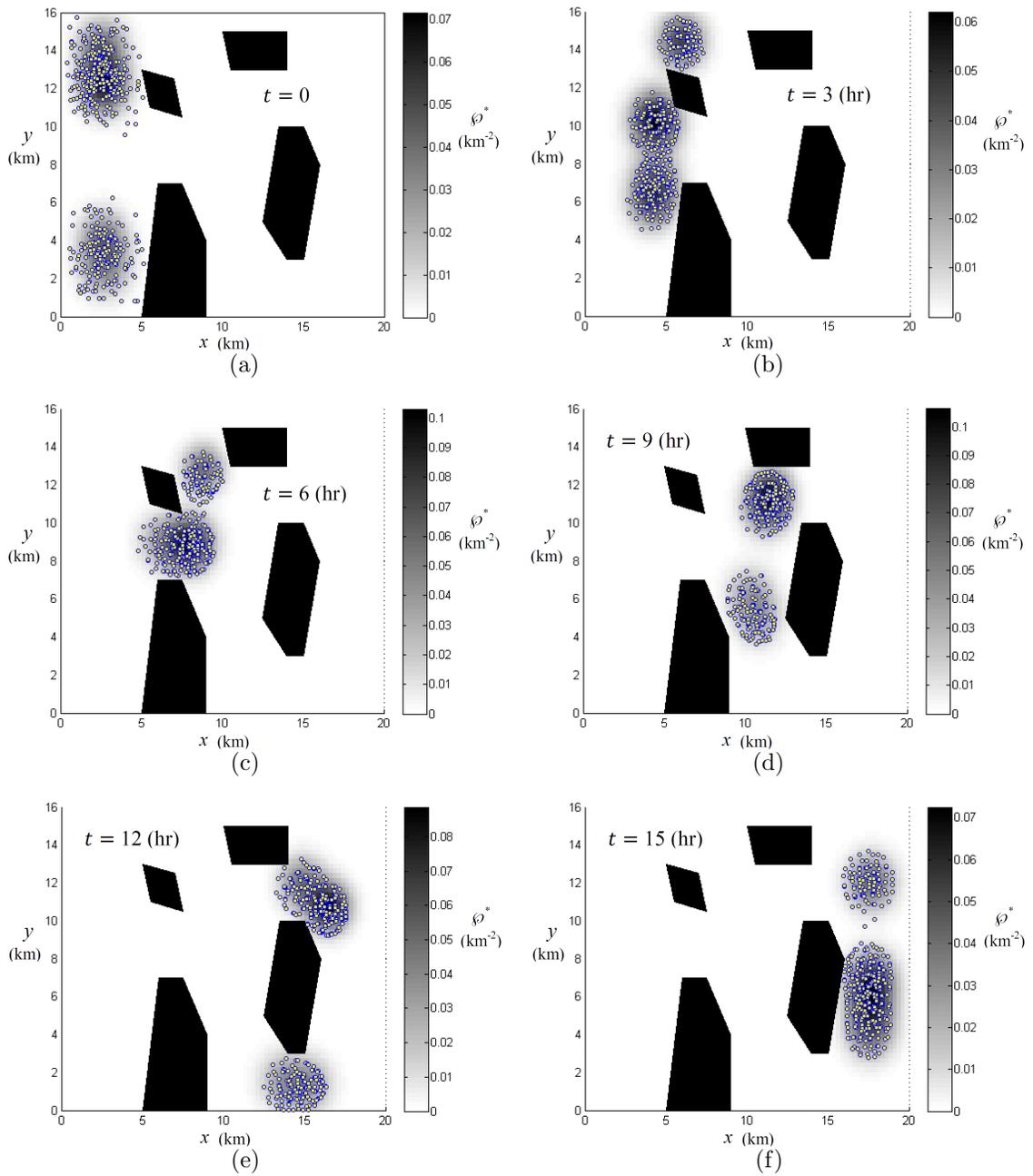


FIGURE 7.10: The evolution of the optimal agent PDF φ^* and the microscopic agent trajectories for the case with *no currents* is computed with the *direct* DOC algorithm and plotted at six instants in time.

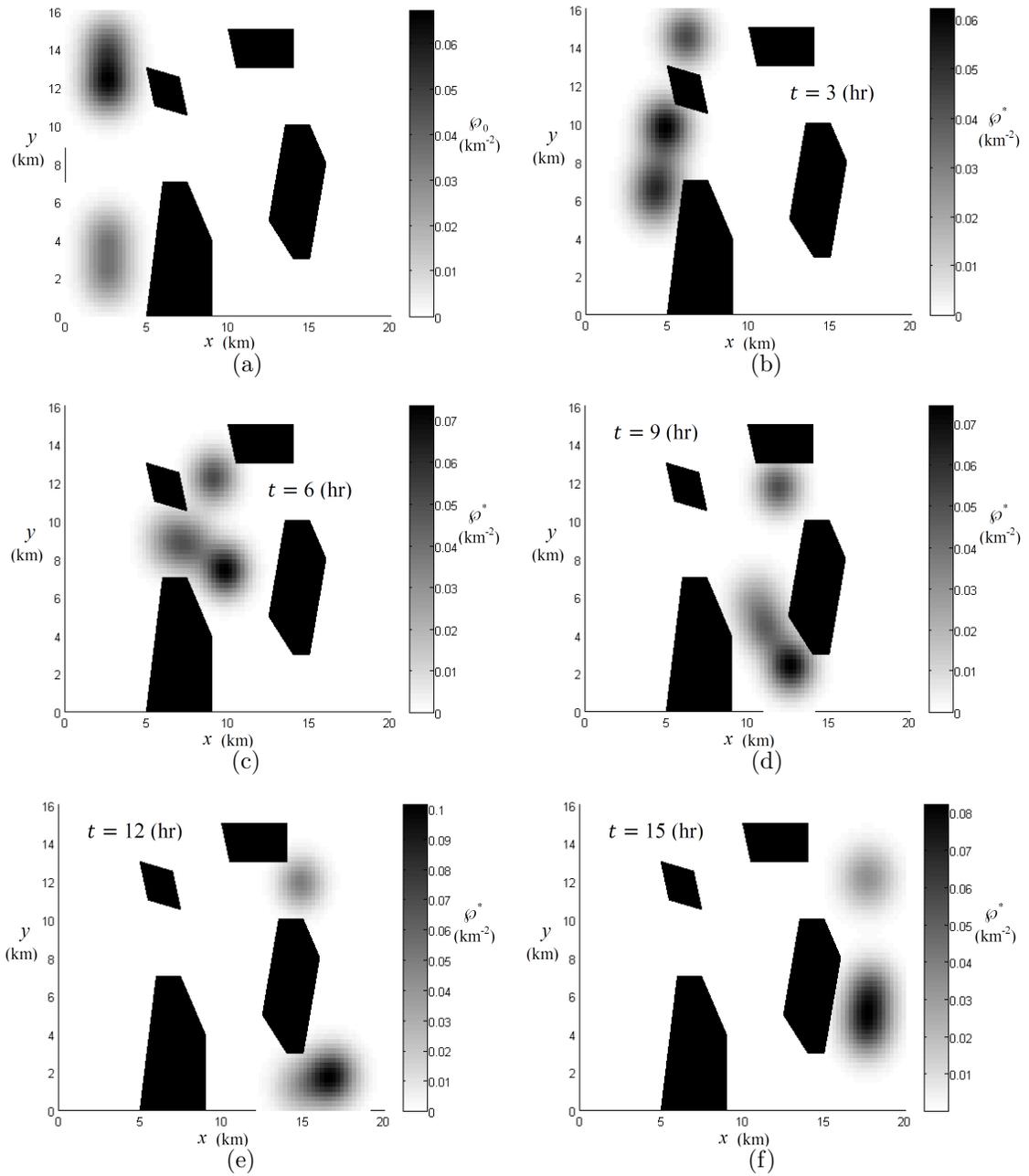


FIGURE 7.11: The evolution of the optimal agent PDF φ^* for the case with *currents* is computed with the *direct* DOC algorithm and plotted at six instants in time.

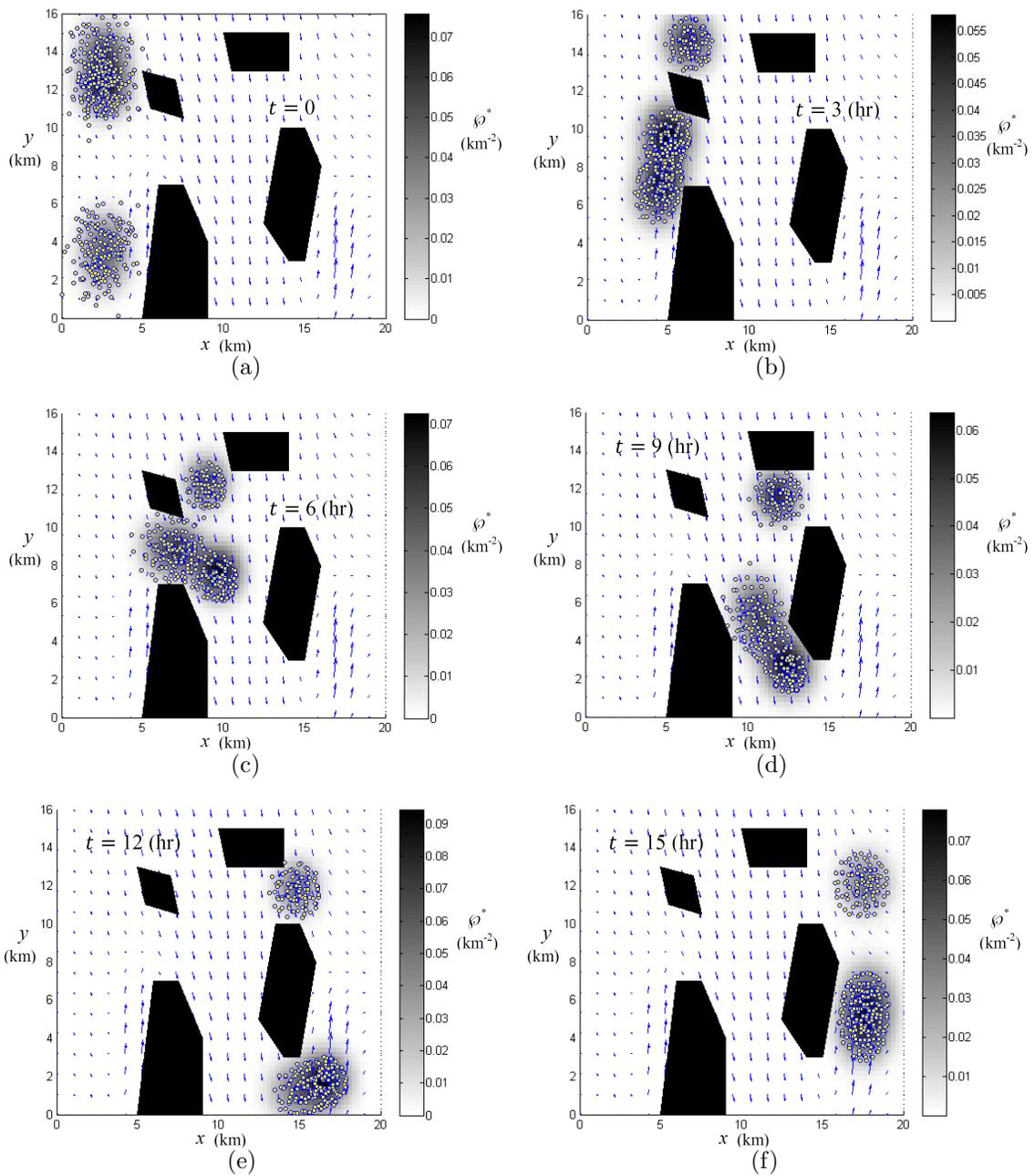


FIGURE 7.12: The evolution of the optimal agent PDF φ^* and the microscopic agent trajectories for the case with *currents* is computed with the *direct* DOC algorithm and plotted at six instants in time.

that path in this case.

7.2.3 Comparison of Direct and Indirect Approaches

In this section, a comparison between the direct and indirect DOC optimization approaches is presented to demonstrate the differences and advantages of the two algorithms. Also results from a stochastic gradient descent approach, a popular existing method for similar problems [58], are shown for comparison. As described in Chapter 6, the direct method discretizes the continuous DOC problem about a finite set of collocation points, and the PDE constrained optimization problem is transcribed into a finite-dimensional nonlinear program, which is then solved using a sequential quadratic programming algorithm. For agent dynamics that are governed by SDEs, the indirect approach uses a GRG method for solving the optimality conditions obtained from the PDE-constrained optimization problem. The GRG method eliminates the state and co-state variables at each iteration of the optimization by solving the corresponding PDEs, and the gradient, with respect to the control, of the macroscopic cost functional is obtained [15, 16].

Since the agent PDF is approximated by a mixture of Gaussians for the direct optimization approach, the shape of the PDF is limited. In complex workspaces, the optimal PDF shape might be irregular, and in these cases, the direct method will be hindered. However, since the indirect approach uses harmonic functions to approximate the PDFs, it has a greater level of flexibility in its solution, and as a result, might be capable of achieving a better solution than the direct method.

To demonstrate this, the optimal state and control trajectories for the problem described in Section 7.2.2 are computed using the DOC approach with the indirect optimization algorithm, described in Section 6.2, for the case described both with and without the added environmental effects. The optimal evolution of the agent PDF for the case without currents is plotted in Figure 7.13, and the same solution is

shown in Figure 7.14 with the microscopic states of $N = 500$ agents that are control by the DOC method. The same is done for the case with currents, where the optimal time-varying agent PDF is shown in Figure 7.15, and the solution with the resulting $N = 500$ agent states and current vector field superimposed is presented in Figure 7.16. As with the direct method, it is shown that the indirect DOC approach exploits the environment to find the optimal agent trajectories. The optimal PDF evolution is also seen to form shapes that bend around corners and are more efficient than those of the Gaussian mixture model.

As an additional comparison, a stochastic gradient descent algorithm described in [58], is simulated for the same problem. Stochastic gradient descent is an approach for obtaining the control laws for a system of robots in stochastic scenarios, where the principle underlying the algorithm is to express a goal configuration of the system as a minimum of an objective function and then to apply a gradient descent on that function as a motion plan from the initial to the goal configuration. This method follows the same idea as classical potential field techniques for feedback motion planning, where a gradient descent is performed on a function to compute a path from an initial state to a goal configuration, and where the agents have local control policies that are not coupled with other agents. Stochastic gradient descent can also typically be implemented in environments with uncertain dynamics or measurements. For this example, the goal states of the agents were sampled from the target distribution, g . The trajectories determined using the stochastic gradient descent approach for the case with currents and $N = 500$ agents is shown in Figure 7.17.

The performance of the direct DOC, indirect DOC, and stochastic gradient descent approach for the example with $N = 500$ agents and a current velocity field are shown in Table 7.1 and Figure 7.18. Due to its ability to represent a greater range of PDF shapes, the indirect DOC approach was able to outperform the direct method

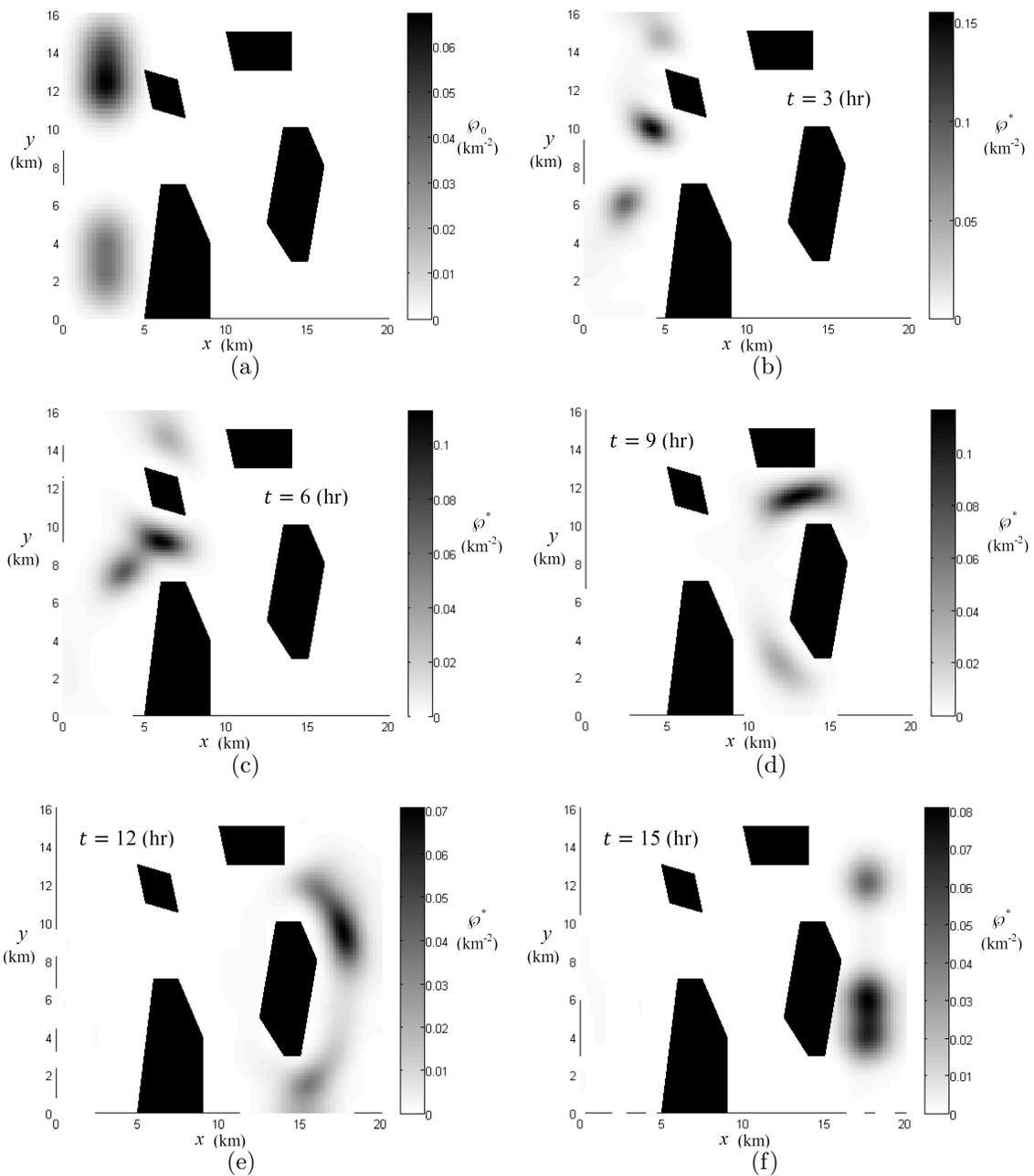


FIGURE 7.13: The evolution of the optimal agent PDF φ^* for the case with *no currents* is computed with the *indirect* DOC algorithm and plotted at six instants in time.

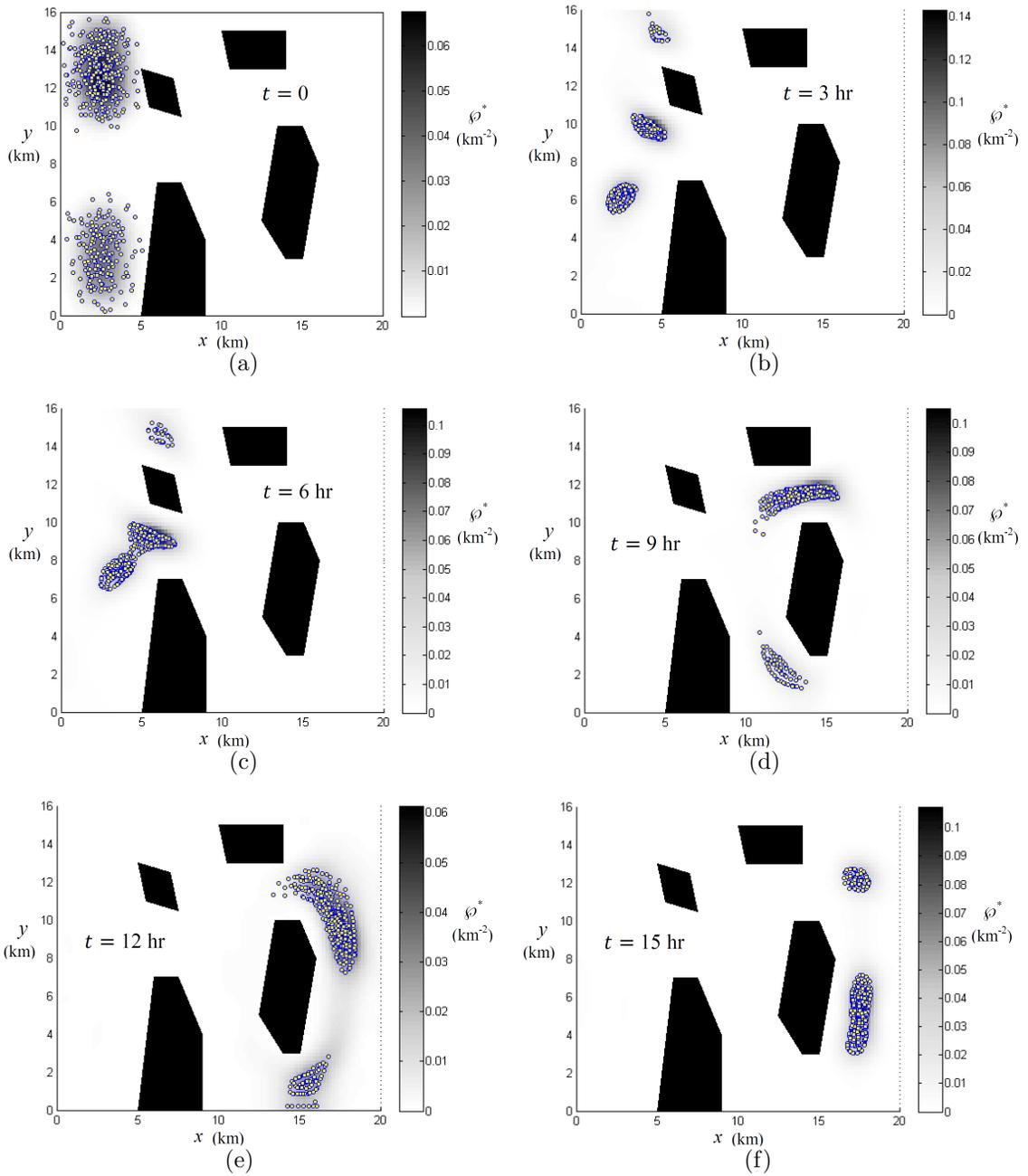


FIGURE 7.14: The evolution of the optimal agent PDF ϕ^* and the microscopic agent trajectories for the case with *no currents* is computed with the *indirect* DOC algorithm and plotted at six instants in time.

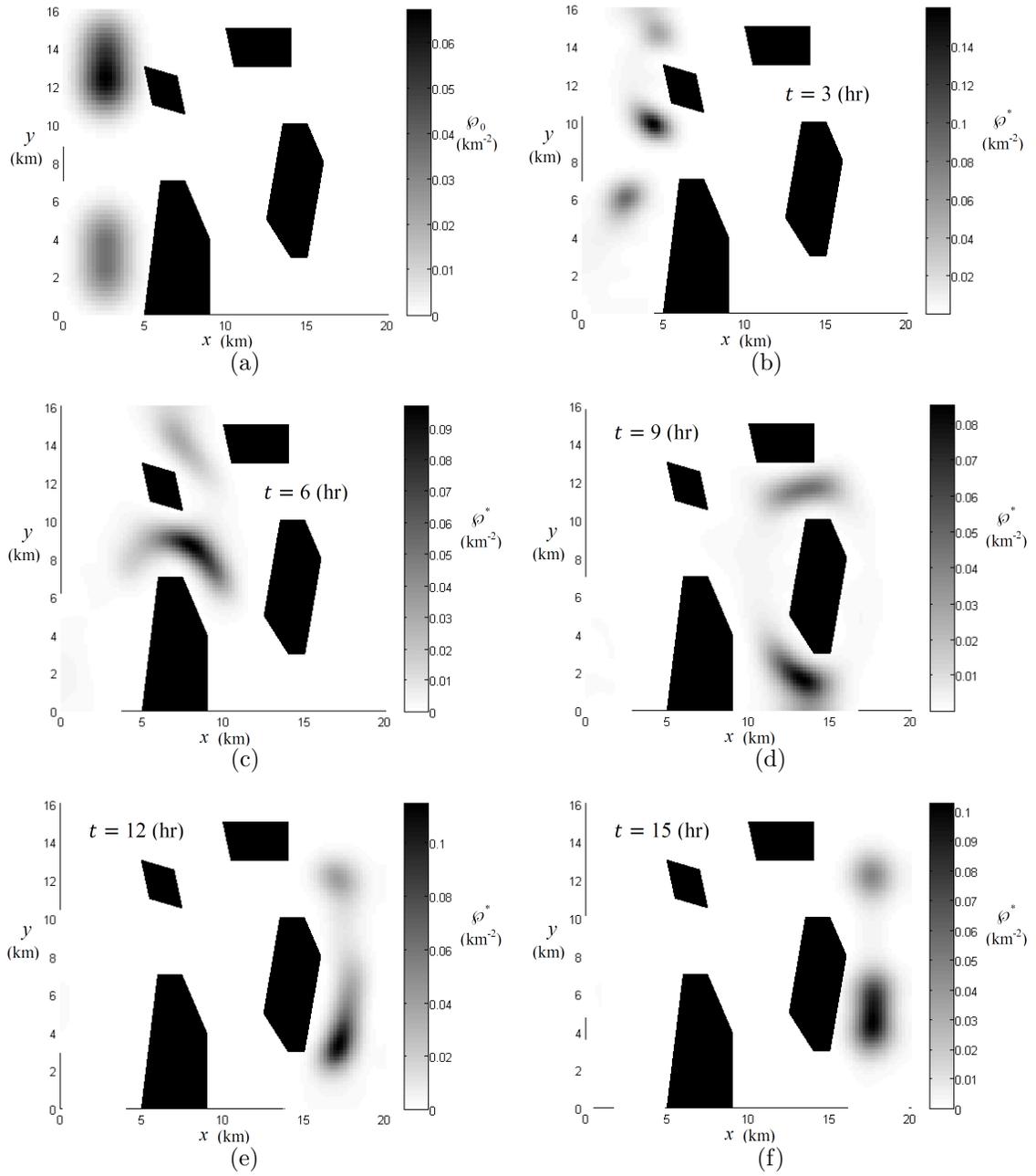


FIGURE 7.15: The evolution of the optimal agent PDF φ^* for the case with *currents* is computed with the *indirect* DOC algorithm and plotted at six instants in time.

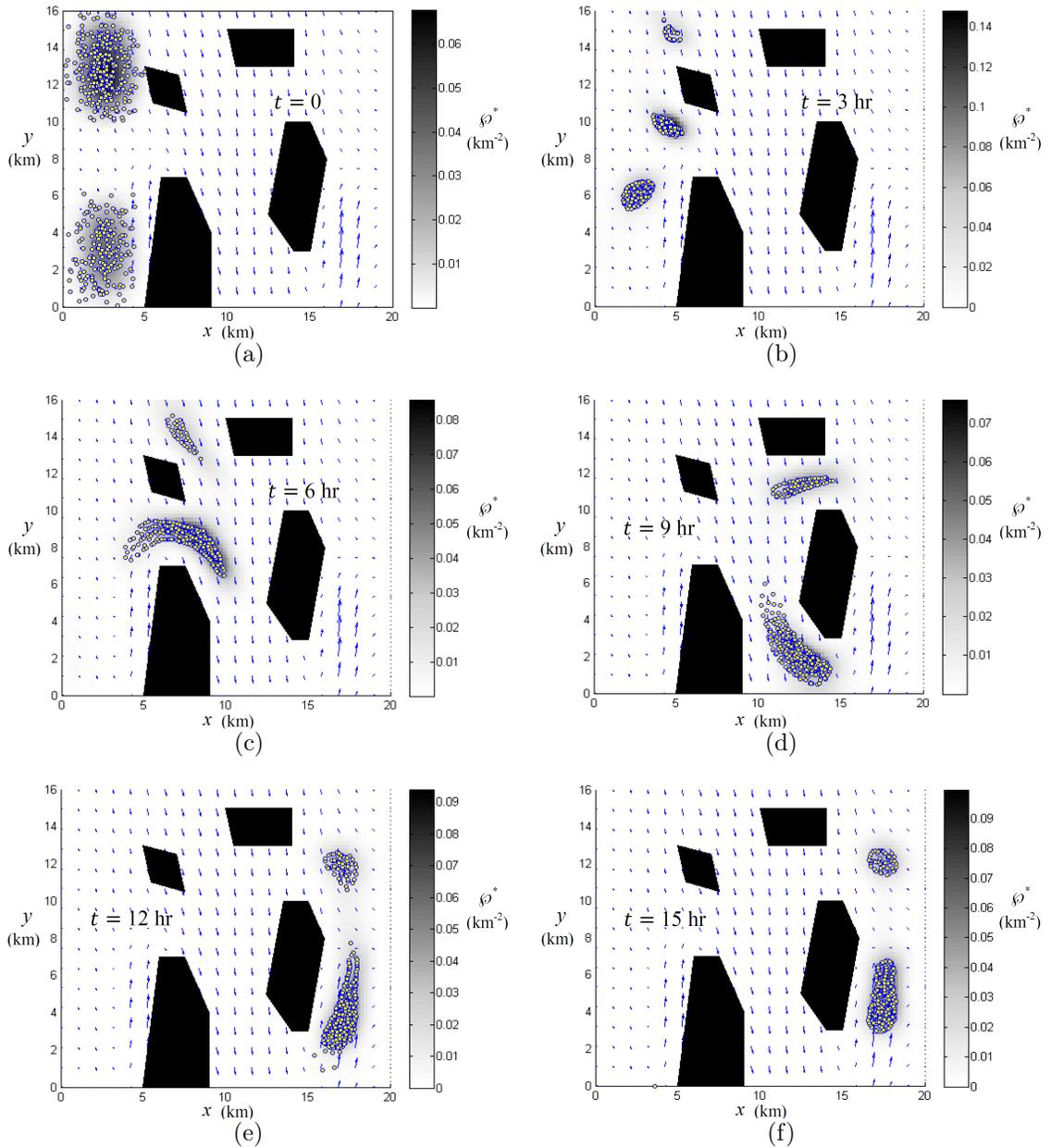


FIGURE 7.16: The evolution of the optimal agent PDF φ^* and the microscopic agent trajectories for the case with *currents* is computed with the *indirect* DOC algorithm and plotted at six instants in time.

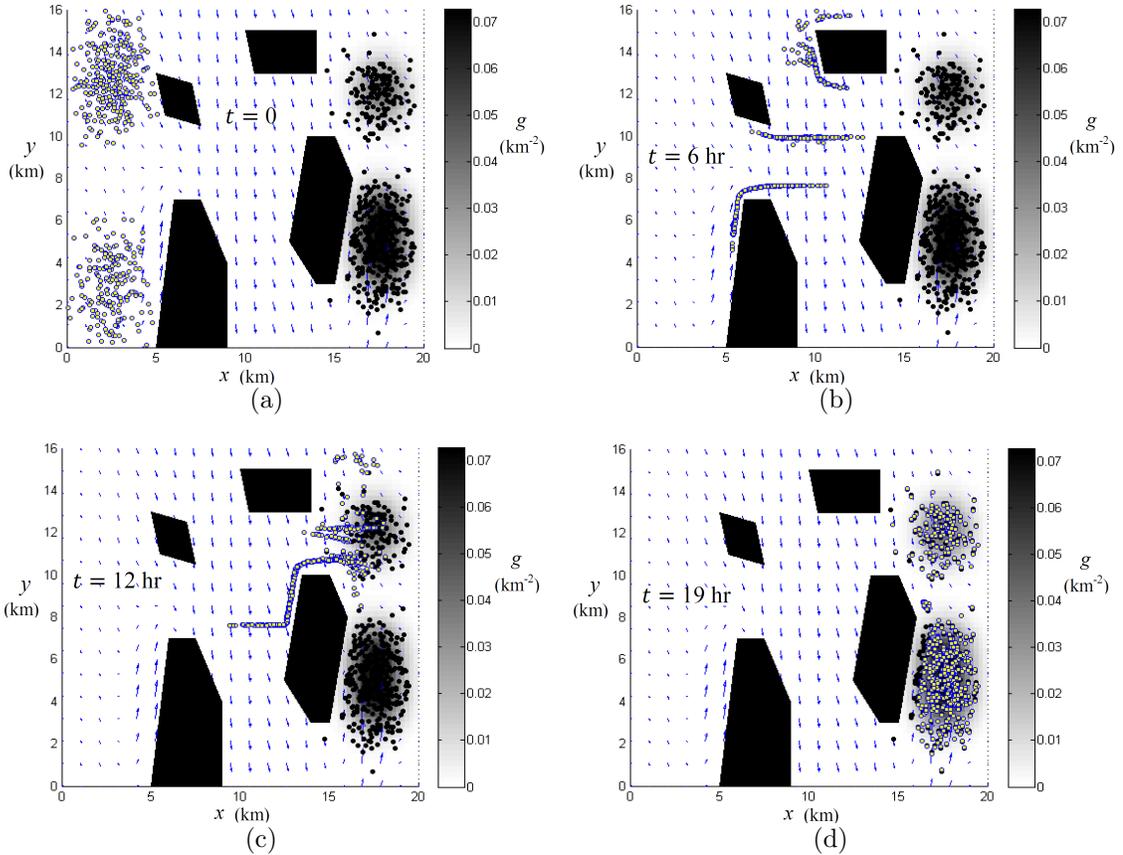


FIGURE 7.17: The microscopic agent trajectories for the case with currents is computed with the stochastic gradient descent algorithm and plotted at six instants in time.

in this case. The indirect method also has a much shorter runtime due to its use of an analytical gradient of the cost function. In contrast, the direct method must compute finite difference approximations of the gradient, leading to many more cost function evaluations during the optimization. The stochastic gradient descent approach has a very short runtime, but since the agents' trajectories are not optimized over the time interval and the agents' dynamics are decoupled, the method's performance is much worse than both DOC algorithms. Also note that the trajectories determined from the DOC method reach the target distribution in less time than those found with stochastic gradient descent.

Table 7.1: Performance comparison of trajectory planning approaches

	Integral Cost, J	Run Time (hr)	Peak Memory (Gb)
Direct DOC	78.79	58.31	3.751
Indirect DOC	67.16	11.44	4.622
Stochastic Gradient Descent	134.98	0.095	0.021

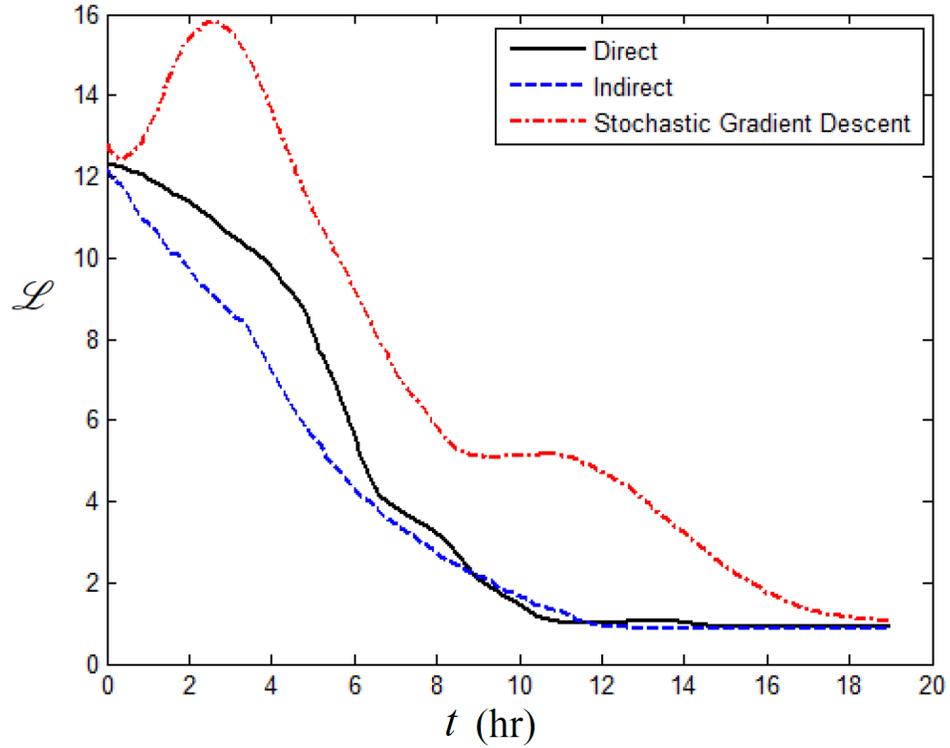


FIGURE 7.18: Comparison of the Lagrangian function evaluated over the time period $(T_0, T_f]$ for the direct DOC, indirect DOC, and stochastic gradient descent approaches.

A Distributed Optimal Control Approach to Sensor Network Trajectory Optimization

The novel distributed optimal control methodology discussed in the previous chapters is demonstrated on a cooperative track detection problem in sensor networks, where the fundamental objective is to compute the agent trajectories that optimize the probability that a target track through a rectangular region of interest (ROI) will be detected by various independent sensors at several times. Cooperative track detection is known to be well-suited to systems where no knowledge about the targets is given *a priori*, and the sensors are likely to report false alarms while opportunities for correct target detections are infrequent. Cooperative track detection fuses multiple closest-point-of-approach (CPA) detections from different sensors to confirm detections, and the tracks of an unknown number of targets can be constructed from multiple consecutive frames of observations provided by low-cost sensors utilizing multiple hypothesis tracking (MHT) [39] or geometric invariants [84] algorithms. However, there are no current methods capable of optimizing the objectives of large sensor networks subject to nonlinear vehicular dynamics and time-varying environ-

mental states. It is shown in this dissertation that distributed optimal control is effective even with such difficult conditions.

8.1 Problem Formulation

This paper considers the problem of optimizing the state and control trajectories of a network of N mobile sensors used to detect a moving target in a rectangular ROI, $\mathcal{W} = [0, L] \times [0, L] \subset \mathbb{R}^2$, during a fixed time interval, $t \in (T_0, T_f]$, where T_0 and T_f are both given. The ROI may be populated with obstacles, and each sensor is mounted on a robot or vehicle whose motion is governed by a small system of ODEs,

$$\dot{\mathbf{x}}_i(t) = \mathbf{f}[\mathbf{x}_i(t), \mathbf{u}_i(t), t], \quad \mathbf{x}_i(T_0) = \mathbf{x}_{0_i} \quad (8.1)$$

where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^n$ is the vehicle state, \mathcal{X} is an n -dimensional state space, $\mathbf{u}_i \in \mathcal{U} \subset \mathbb{R}^m$ is the vehicle control vector, and \mathcal{U} is the space of m admissible control inputs of the i^{th} mobile sensor.

It is assumed that the system performance is a function of the macroscopic network state, $\mathbf{X} \in \mathbb{R}^\ell$, where $\ell < n$. The macroscopic state, \mathbf{X} , represents the macroscopic dynamic condition of the sensor network, and is obtained via a restriction operator, $\wp : \mathcal{X} \times \mathbb{R} \rightarrow \mathbb{R}$, which maps the microscopic sensor state vectors into the macroscopic description, such that $\mathbf{X}(t) = \wp[\mathbf{x}(t), t]$. In this paper, the restriction operator is the PDF of $\mathbf{x} = [x \ y]^T \in \mathcal{W}$, and, thus, \wp also represents the density of sensors in \mathcal{W} at time t .

The sensors are assumed to be omnidirectional, with a constant effective range, $r \in \mathbb{R}$, defined as the maximum range at which the received signal exceeds a desired threshold [8]. Then, the field-of-view (FOV) of every sensor can be modeled by a disk $\mathcal{C}(\mathbf{x}, r) \subset \mathbb{R}^2$, with center at \mathbf{x} , and radius r (Fig. 8.1). The complexity of the assumed spatio-temporal track model is a function of the size of the ROI, and of the expected target dynamics [27]. In this paper, it is assumed that the target moves at

constant speed, $v_T > 0$, and heading, $\theta_T \in [0, 2\pi)$, such that every possible target track can be represented by a ray or half line, $\mathcal{R}_{\theta_T}(\mathbf{x}_{T_0}) \in \mathbb{R}_+^2$, with slope $\tan(\theta_T)$, and origin \mathbf{x}_{T_0} (Fig. 8.1). However, the same approach can also be extended to Markov motion models [81], as will be shown in a separate paper. Then, the probability of a target detection by a sensor at \mathbf{x} can be described by the Boolean detection model

$$P_d[\mathbf{x}(t), t] = \begin{cases} 1, & \text{if } \mathcal{R}_{\theta_T}(\mathbf{x}_{T_0}) \cap \mathcal{C}(\mathbf{x}, r) \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \quad (8.2)$$

A target track, $\mathcal{R}_{\theta_T}(\mathbf{x}_{T_0})$, is said to be detected when it can be formed from multiple independent detections obtained by the sensor network over time [84, 46, 43, 24]. For example, the event-based algorithm developed in [84] can be used to determine all possible target tracks based on multiple closest-point-of-approach (CPA) detections obtained by a proximity network. The number of detections required per track is represented by a constant parameter k , such that $1 \leq k \leq N$, and its value is decided based on the level of confidence required by the sensor system [84]. Then, the *probability of track detection* is defined as the probability of obtaining k independent detections when a target is present in the ROI.

The problem considered in this paper is to compute collision-free state and control trajectories, \mathbf{x}^* and \mathbf{u}^* , for all N sensors, such that the probability of track detection in \mathcal{W} is maximized, and the energy consumption is minimized, subject to the microscopic agent dynamics (8.1). The trajectory optimization problem described in this section can be formulated as a DOC problem in which the network performance depends on the sensors' PDF, as show in the next section.

8.2 Cooperative Track Detection

8.2.1 Background on Track Coverage

In surveillance systems that employ distributed sensors to detect intruding targets over a large region of interest (ROI), a track-before-detect strategy can be utilized to determine the positional information of the targets with a high level of confidence, even when false-alarms are frequent. The problem of cooperative track detection was first introduced in [46], and is concerned with the probability that a target traversing through a ROI is detected by a cooperative sensor network by obtaining multiple elementary (e.g. closest-point-of-approach) detections at various moments in time. A target is stated as detected when its track can be formed or estimated by fusing multiple independent detections according to an assumed spatio-temporal model of the target tracks. With this methodology, the tracks of an unknown number of targets can be assembled from multiple consecutive frames of observations. These systems are often relatively affordable since this data can be collected by simple, low-cost (e.g. passive) proximity sensors, using multiple hypothesis tracking [39] or geometric invariants [84] algorithms.

A general omni-directional proximity sensor can be defined such that the signal received by the i^{th} sensor at \mathbf{x}_i can be described as isotropic energy attenuated by the environment according to the following power law,

$$E_i(t) = cF[\lambda_i(t)]^{-\alpha} \quad (8.3)$$

where, $\lambda_i(t)$ is the distance between the i^{th} sensor and the target at time t . The attenuation coefficient, α , and the scaling constant, c , depend on the physical mechanisms of wave propagation and on the environmental conditions. F represents the target source level, and is assumed to be independent of both time and sensor location [21, 84]. Then, a closest-point-of-approach (CPA) detection is said to take place when E_i exceeds a threshold ϑ_i , which is typically tuned by an operator [78]. At the

CPA detection time, the values of E_i and \mathbf{x}_i are reported by the i^{th} sensor to the central processor. From (8.3), the maximum range at which the i^{th} sensor can report a CPA detection, given a target source level F and a threshold ϑ_j , is calculated as,

$$r_i = (cF)^{1/\alpha} \vartheta_i \quad (8.4)$$

Thus if the effects of any sensor vehicle propulsion on sensing are neglected, the value of r_i can be estimated from the environmental conditions, and can be assumed known and constant for all $i = 1, \dots, n$ [7, 82].

The probability of cooperative track detection, also known as track coverage [7], is a useful metric for sensor network performance that can be used as an objective function for optimization. When n is very large, it can be derived by the distributed search approach presented in [82]. This approach assumes that all n sensors have the same range, $r_i = r$ for all i , and that the probability of detection of the i^{th} sensor is equal to one everywhere inside their field-of-view, and is equal to zero elsewhere.

The dimensions of the ROI, \mathcal{W} , and the time interval $[T_0, T_f]$ are chosen such that the target vehicles can be assumed to move at a constant speed V and heading θ , and to maintain a constant source amplitude. After a minimum of k detections are obtained from k distinct sensors in the network, the values of E_i and \mathbf{x}_i are fused by a central processor to estimate the target track [84]. The number of required target detections k depends on the false-alarm rate, on the measurement errors, and on the track accuracy required by the surveillance system [84].

The sensors' state and the targets' speeds, headings, and initial positions are considered as random variables described by the joint PDFs $f_{\mathbf{x}}(\mathbf{x}_i, t)$, $f_V(V, t)$, $f_{\theta}(\theta, t)$, and $f_T(\mathbf{x}_{T_0}, t)$, respectively. The PDF of the sensors' state is a function of time, because the sensors move to optimize their probability of track detection. The PDFs of the target track's parameters are assumed to be known functions of time that are obtained from the aforementioned target-tracking algorithms [39, 84]. Then, the

detection region $\Omega_T \subset \mathcal{W}$ can be grown isotropically from the target track,

$$\mathbf{x}_T(t) = \mathbf{x}_{T_0} + V[\cos \theta \ \sin \theta]^T dt \quad (8.5)$$

over a time differential $dt \subset [T_0, T_f]$, where $\mathbf{x}_T(T_0) = \mathbf{x}_{T_0} \in \mathcal{W}$. Let the event $D_i = \{1, 0\}$ represent the set of all possible mutually-exclusive outcomes corresponding to sensor i reporting (1) or not reporting (0) a target detection. Then, assuming the targets are distributed uniformly in \mathcal{W} , the probability of a detection being reported by sensor i is given by a spatial Poisson process,

$$\Pr\{D_i = 1 \mid \mathbf{x}_T(t) \in \mathcal{W}\} = 1 - e^{-\phi_t} \quad (8.6)$$

where,

$$\phi_t(\mathbf{x}_{T_0}, V, \theta) = \int_{T_0}^{T_f} \int_{\Omega_T(\mathbf{x}_{T_0}, \theta, V dt)} f_{\mathbf{x}}(\mathbf{x}_i, t) d\mathbf{x} dt \quad (8.7)$$

is the coverage factor for a sensor sampled from $f_{\mathbf{x}}(\mathbf{x}_i, t)$, and with a detection region Ω_T . The coverage factor of a spatial Poisson process is defined as the expected value of the number of points that fall in a small region or subset of a Euclidian space. Where, every point that falls into this region corresponds to a detection event $D_i = 1$.

In a network of n sensors, the set of events $\{D_1, \dots, D_n\}$ is reported to the central processor to attempt to form a target track, and a successful track detection is declared when $\sum_{i=1}^n D_i \geq k$. Thus, the probability of a successful track detection by at least k sensors can be described using Bernoulli trials [82]. Assuming that individual detection events are statistically identical and independent, and that $\phi_t \ll 1$ and $n \gg 1$, the probability of successful track detection in \mathcal{W} can be approximated by

an integral function of the sensors' PDF,

$$\begin{aligned}
P_t &\equiv \Pr\left(\sum_{i=1}^n D_i \geq k \mid \mathbf{x}_T(t) \in \mathcal{W}\right) \\
&\approx 1 - \int_{T_0}^{T_f} \int_0^{2\pi} \int_{V_{min}}^{V_{max}} \int_{\mathcal{W}} e^{-n\phi_t(\mathbf{x}_{T_0}, V, \theta)} f_T(\mathbf{x}_{T_0}, t) \\
&\quad \times f_V(V, t) f_\theta(\theta, t) \sum_{m=0}^{k-1} \frac{[n\phi_t(\mathbf{x}_{T_0}, V, \theta)]^m}{m!} d\mathbf{x}_{T_0} dV d\theta dt
\end{aligned} \tag{8.8}$$

as shown in [82]. Where, V_{min} and V_{max} are the target's minimum and maximum speeds, respectively, and the coverage factor $\phi_t(\mathbf{x}_{T_0}, V, \theta)$ is a function of $f_{\mathbf{x}}(\mathbf{x}_i, t)$, as shown in (8.7).

8.2.2 Probabilistic Track Coverage

It was recently shown that the probability of track detection is a function of the sensors' positions that can be derived in closed form by considering the cone generated by the sensor FOV, also referred to as coverage cone [27]. For an omnidirectional FOV, $\mathcal{C}(\mathbf{x}, r)$, the coverage cone $K(\mathcal{C}, \mathbf{x}_{T_0})$ is defined as the *cone generated by* \mathcal{C} , i.e., as the set of all nonnegative combinations of the elements of \mathcal{C} , with origin \mathbf{x}_{T_0} [7]. An example of coverage cone for an omnidirectional sensor positioned at \mathbf{x} , and with an effective range r is illustrated in Fig. 8.1. Then, the coverage cone $K(\mathcal{C}, \mathbf{x}_{T_0})$ can be shown to contain all target tracks in \mathcal{W} that are detected by the sensor, and that lie between the cone extremals, characterized by lines with orientations g_1 and g_2 [7].

Since the track parameters v_T , θ_T , and \mathbf{x}_T typically are uncorrelated random variables, prior target information is provided in terms of the probability density functions (PDFs) $f_{\mathbf{x}_T}(\mathbf{x}_T)$, $f_{\theta_T}(\theta_T)$, and $f_{v_T}(v_T)$, which are routinely outputted by target tracking algorithms [5]. Then, all target tracks with origin \mathbf{x}_{T_0} can be viewed as Poisson flats that are placed in the open cone with a density $f_{\theta_T}(\theta_T)$ [27]. By viewing \mathbf{x} as a random variable with PDF \wp , the coverage cone $K(\mathcal{C}, \mathbf{x}_{T_0})$ can be

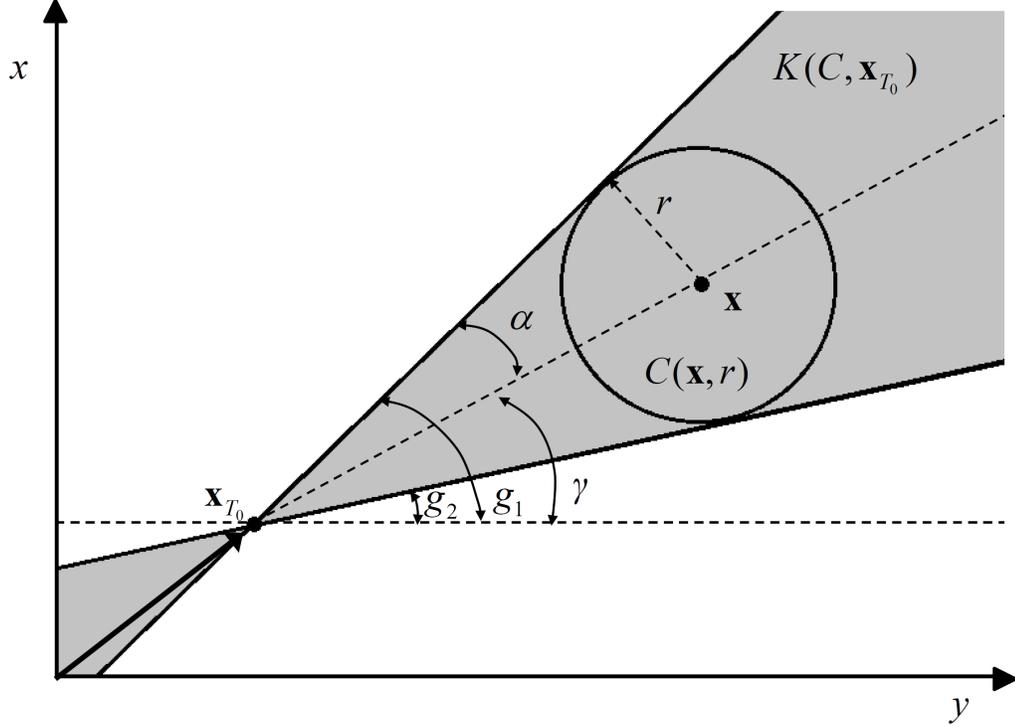


FIGURE 8.1: Coverage cone of an omnidirectional sensor at \mathbf{x} for a target track with origin \mathbf{x}_{T_0} .

considered as an interval of length 2α that is placed randomly in θ -phase space, or $\theta \in [0, 2\pi)$. As shown in detail in [27], the expected number of Poisson flats that fall in $K(\mathcal{C}, \mathbf{x}_{T_0})$ can be obtained by writing the cone extremals in rectangular coordinates,

$$g_{1,2}(\mathbf{x}, \mathbf{x}_{T_0}) = \begin{cases} \{0, 2\pi\}, & \text{if } \|\mathbf{x} - \mathbf{x}_{T_0}\| \leq r \\ \gamma(\mathbf{x}, \mathbf{x}_{T_0}) \pm \alpha(\mathbf{x}, \mathbf{x}_{T_0}), & \text{otherwise} \end{cases} \quad (8.9)$$

where

$$\gamma(\mathbf{x}, \mathbf{x}_{T_0}) \triangleq \sin^{-1}[(y - y_T) / \|\mathbf{x} - \mathbf{x}_{T_0}\|] \quad (8.10)$$

$$\alpha(\mathbf{x}, \mathbf{x}_{T_0}) \triangleq \sin^{-1}(r / \|\mathbf{x} - \mathbf{x}_{T_0}\|). \quad (8.11)$$

Assuming all N sensor positions are independently and identically sampled from the PDF φ , multiple detections can be viewed as repeated Bernoulli trials. It follows

that the probability that a detection event will occur at least k times is given by,

$$\Pr \left(\sum_{i=1}^N D_i \geq k \mid \mathbf{x}_{T_0} \in \mathcal{W} \right) = 1 - \sum_{m=0}^{k-1} \frac{N!}{m!(N-m)!} \phi_s^m (1 - \phi_s)^{N-m} \quad (8.12)$$

where $D_i \in \{1, 0\}$ denotes the detection event for a sensor indexed by i , for $i = 1, \dots, N$, and the outcomes 1 and 0 represent a successful and unsuccessful detection events, respectively [27]. Let ϕ_s denote the expected number of rays with origin \mathbf{x}_{T_0} that fall in the coverage cone $K(\mathcal{C}, \mathbf{x}_{T_0})$, referred to as *coverage factor*, such that

$$\phi_s \triangleq \int_{\mathcal{W}} \wp(\mathbf{x}, t) \int_{g_1(\mathbf{x}, \mathbf{x}_{T_0})}^{g_2(\mathbf{x}, \mathbf{x}_{T_0})} f_{\theta_T}(\theta_T) d\theta_T d\mathbf{x} \quad (8.13)$$

Then, for an independent random variable \mathbf{x}_{T_0} with PDF $f_{\mathbf{x}_T}(\mathbf{x}_{T_0})$, and for $\phi_s \ll 1$ and $N \gg 1$, the probability of obtaining at least k independent target detections, or probability of track detection, is

$$\begin{aligned} P_s &= 1 - \int_{\mathcal{W}} f_{\mathbf{x}_T}(\mathbf{x}_{T_0}) e^{-N\phi_s} \sum_{m=0}^{k-1} \frac{(N\phi_s)^m}{m!} d\mathbf{x}_{T_0} \\ &\triangleq 1 - P_t(\wp) \end{aligned} \quad (8.14)$$

The complete proof, provided in [27], is based on the Poisson limit theorem.

The probability of track detection, P_s , is a performance function that represents the track coverage for a probabilistic sensor network with a distribution, \wp , and multiple targets with tracking parameters with uncertainties described by the PDFs $f_{\mathbf{x}_T}$ and f_{θ_T} , that are possibly nonuniform. It can be seen from (8.14), that the network track-coverage performance is a function of the sensor distribution, \wp , and, consequently, of the sensor control input, \mathbf{u} , by virtue of the dynamic equation (8.1). The obstacle avoidance and energy minimization objectives can also be expressed as a function of \wp and \mathbf{u} , by introducing a repulsive potential function U_{rep} generated

from the obstacles in \mathcal{W} , as shown in [28, 44]. Then, the objective function for the N mobile sensors can be expressed as an integral cost function,

$$J = \int_{T_0}^{T_f} \left[w_s P_s + \int_{\mathcal{W}} (w_r \wp U_{rep} + w_e \mathbf{u}^T \mathbf{R} \mathbf{u}) d\mathbf{x} \right] dt \quad (8.15)$$

to be minimized with respect to \wp and \mathbf{u} , and subject to the dynamic and equality constraints in (3.7)-(3.9), (3.3). The constant weights w_s , w_r , and w_e , are chosen by the user based on the desired tradeoff between the sensing, obstacle-avoidance, and energy objectives, and \mathbf{R} is a diagonal positive-definite matrix.

Because the network performance (8.15) is a function of the macroscopic network state, the DOC approach proposed in [28] can be applied to determine the optimal sensor state and control trajectories, as explained in the following section.

8.3 Numerical Simulations and Results

Two numerical examples are presented to demonstrate the effectiveness of the DOC methodology described in the previous sections. The first example considers a network of $N = 100$ sensors that are each installed on a vehicle with nonlinear unicycle kinematics,

$$\dot{x} = v \cos \theta \quad \dot{y} = v \sin \theta \quad \dot{\theta} = \omega \quad (8.16)$$

and deployed in an obstacle-free workspace $\mathcal{W} = [0, L] \times [0, L]$, with $L = 16$ km, over a time interval $(T_0, T_f]$, with $T_0 = 0$ and $T_f = 15$ hr. The sensor configuration, $\mathbf{q} = [x \ y \ \theta]^T$, consists of the x, y -coordinates, and heading angle θ . The sensor control vector is $\mathbf{u} = [v \ \omega]^T$, where v is the linear velocity, and ω is the angular velocity. The sensors are assumed to have constant linear velocities of $v = 0.5$ km/hr, and maximum angular velocities of $\omega_{max} = 0.52$ rad/s, such that $\omega \in [-\omega_{max}, +\omega_{max}]$. The microscopic feedback control law is defined to minimize the potential function, U , in (5.1), and is obtained from the negative gradient of U . For sensors described

by the nonlinear unicycle model (8.16), the microscopic control law is,

$$\mathbf{u} = [v \quad Q(\theta, \phi)]^T \quad (8.17)$$

where $\phi = -\nabla U$, and,

$$Q(\cdot) = \{a(\theta) - a[\Theta(\phi)]\} \text{sgn}\{a[\Theta(\phi)] - a(\theta)\}$$

is the minimum differential between the sensor's actual heading angle θ and the desired heading angle $\Theta(\phi)$, $\text{sgn}(\cdot)$ is the sign function, and $a(\cdot)$ is an angle wrapping function [44].

It is assumed that the sensors are deployed in \mathcal{W} with an initial distribution \wp_0 , and, thus, at $t = T_0$ they are located at a set of initial positions sampled from \wp_0 . The number of independent elementary detections required to declare a target track detection is chosen to be $k = 2$. The PDF of the initial target position, $f_{\mathbf{x}_T}(\mathbf{x}_{T_0})$, plotted in Fig. 8.2, is a two-dimensional Gaussian mixture,

$$f_{\mathbf{x}_T}(\mathbf{x}_{T_0}) = \sum_{\ell=1}^2 \frac{1}{(2\pi)^{n/2} \det(\Sigma)^{1/2}} e^{[-(1/2)(\mathbf{x}_{T_0} - \boldsymbol{\mu}_\ell)^T \Sigma^{-1} (\mathbf{x}_{T_0} - \boldsymbol{\mu}_\ell)]} \quad (8.18)$$

with means $\boldsymbol{\mu}_1 = [2 \ 2]^T$ km and $\boldsymbol{\mu}_2 = [3 \ 4]^T$ km, and covariance $\Sigma = \mathbf{I}_2$, where \mathbf{I}_n denotes an $n \times n$ identity matrix. The heading PDF, $f_{\theta_T}(\theta_T)$, shown in Fig. 8.3, is

$$f_{\theta_T}(\theta_T) = \begin{cases} 1.92 \text{ (rad}^{-1}\text{)}, & \text{if } 0 \leq \theta_T \leq 0.52 \text{ (rad)} \\ 0, & \text{otherwise} \end{cases} \quad (8.19)$$

For simplicity, the target speed is constant and equal to $v_T = 0.2$ km/hr. Then, the evolution of the target PDF $f_{\mathbf{x}_T}(\mathbf{x}_T)$ over time can be computed from $f_{\mathbf{x}_T}(\mathbf{x}_{T_0})$, $f_{\theta_T}(\theta_T)$, and v_T , and is plotted in Fig. 8.4.

The cost function weights are $w_s = 1$, $w_r = 0$, and $w_e = 0.1$, based on the relative importance of the sensing, obstacle-avoidance, and energy objectives, respectively.

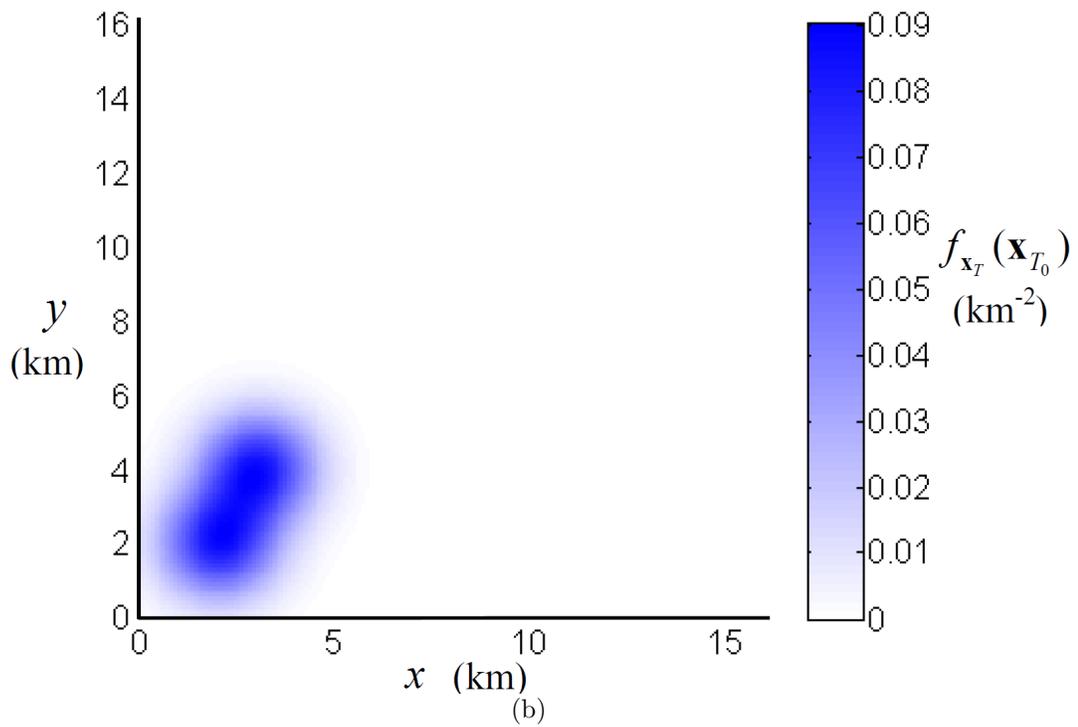
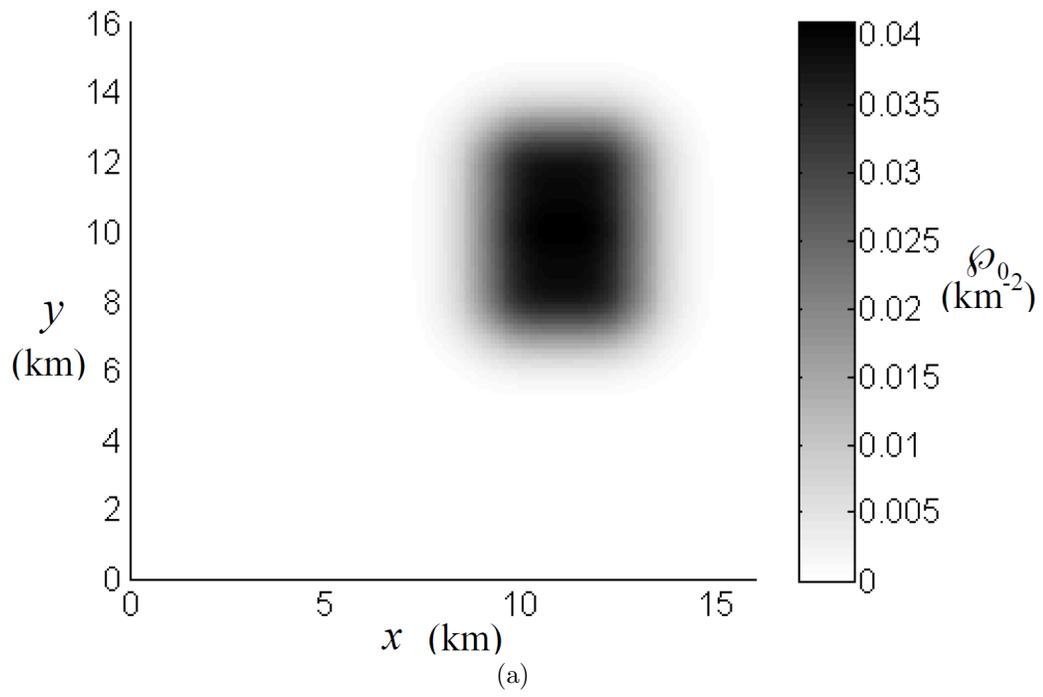


FIGURE 8.2: Initial sensor distribution (a) and PDF of initial target position (b) for obstacle-free example.

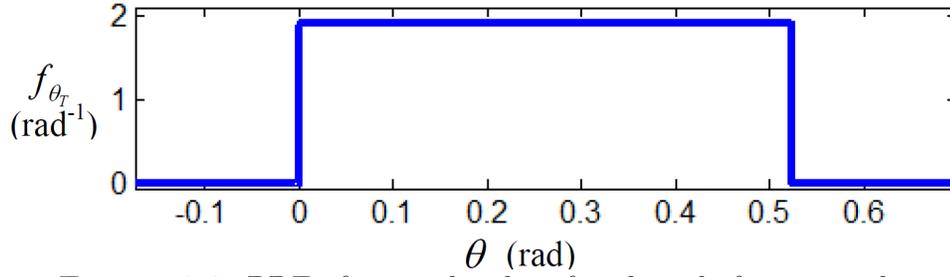


FIGURE 8.3: PDF of target heading for obstacle-free example.

In this example, $w_r = 0$ because the ROI contains no obstacles. The number of mixture components is $z = 6$, and the state space is discretized using $X = 900$ collocation points, and $\Delta t = 1$ hr, such that $K = 15$. The optimal sensor distribution φ^* obtained by the DOC method is plotted in Fig. 8.5 at four sample moments in time, and the corresponding instantaneous cost (or Lagrangian) is plotted in Fig. 8.6 for

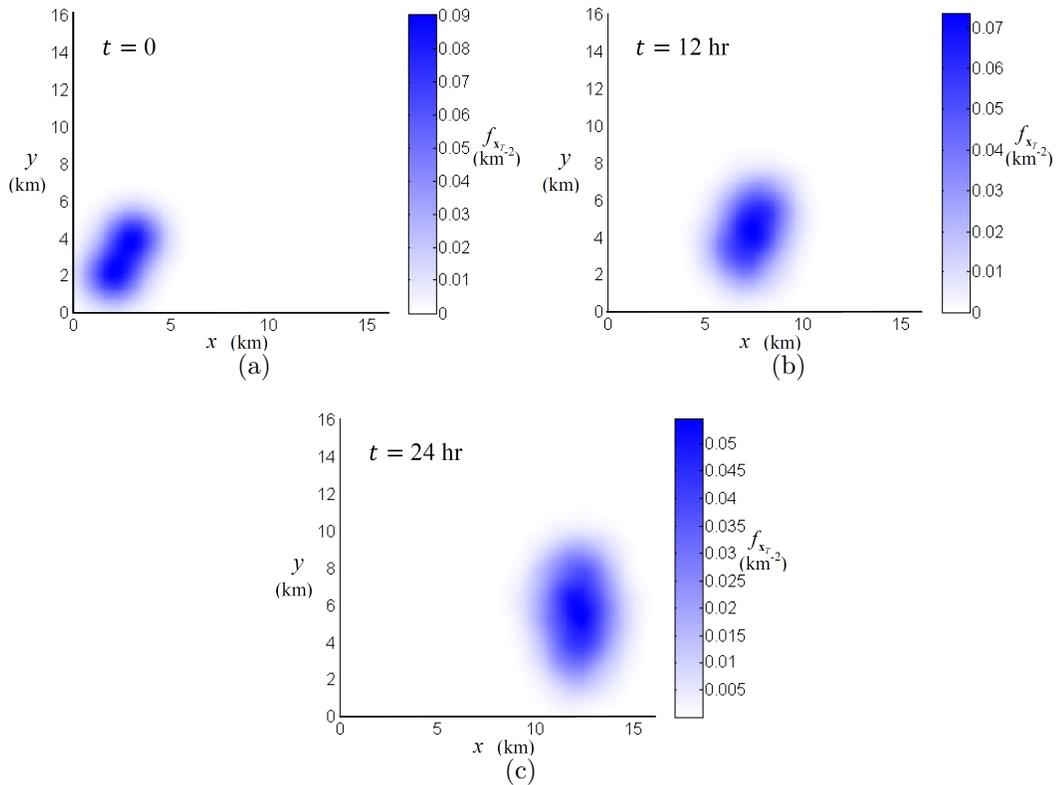


FIGURE 8.4: Evolution of target PDF at three instants in time for obstacle-free example in Figs. 8.2-8.3.

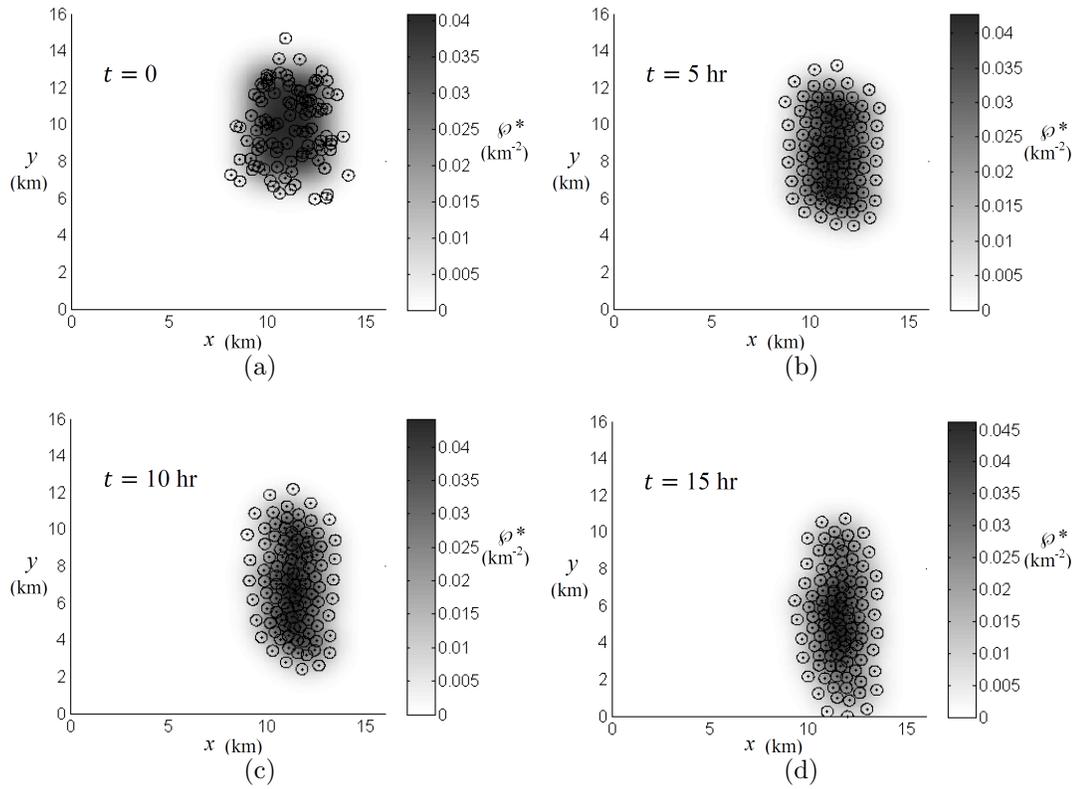


FIGURE 8.5: Evolution of optimal sensor PDF, φ^* , microscopic state (black dots), and FOVs (black circles) at four instants in time, for obstacle-free example in Figs. 8.2-8.3, and $N = 100$ sensors.

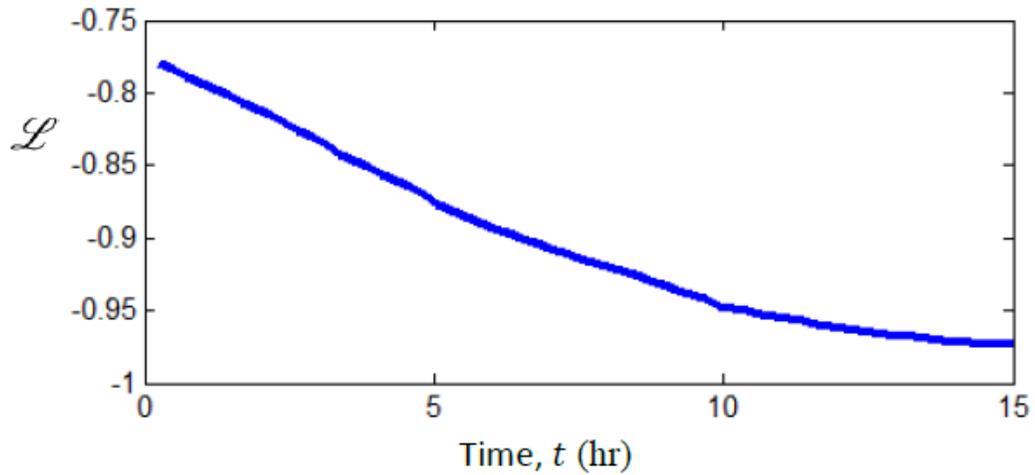


FIGURE 8.6: Time-history of cost function Lagrangian for DOC solution in Fig. 8.5.

the entire interval $(T_0, T_f]$. From φ^* , the sensors state and control trajectories are obtained by integrating the closed-loop microscopic dynamic equation (8.1) numerically, using the DOC feedback control law in (7.5), and a time step $\delta t = 3$ s. The instantaneous sensors state and FOV are superimposed on φ^* in Fig. 8.5 to illustrate the sensors positions along the optimal trajectories over time. It can be seen from Fig. 8.6 that as the Lagrangian decreases over time, as the sensor network moves to accomplish its objectives of optimizing the probability of track detection (8.14) and minimizing the energy consumption associated with control usage.

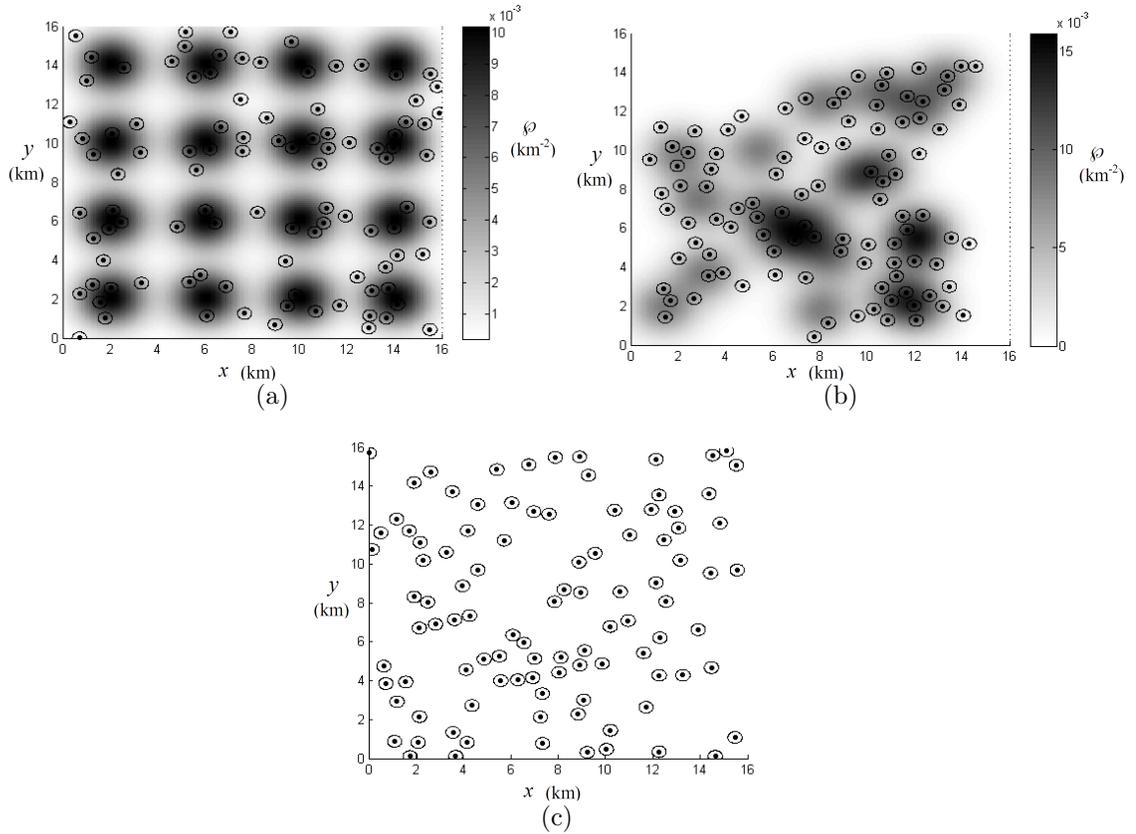


FIGURE 8.7: Grid (a), random (b), and uniform (c) sensor distributions and sampled sensor positions (black dots), and FOVs (black circles).

The performance of the DOC method is also compared to three existing sensor network deployment strategies known as grid, random, and uniform deployments

[56], plotted in Fig. 8.7. The grid deployment is obtained by sampling a Gaussian mixture PDF with support \mathcal{W} , and $z = 16$ components arranged in a grid formation. The random deployment is similarly obtained but uses a Gaussian mixture in which the means are randomly generated over \mathcal{W} , and the uniform deployment is obtained by sampling a uniform distribution with support \mathcal{W} . For all three strategies the sensor positions are obtained using finite-mixture sampling [56], and are assumed to be constant over time. Subsequently, the cost function in (8.15) and the actual number of target track detections were evaluated for each strategy, and averaged over twenty simulations. The results, summarized in Table 8.1, show that sensors deployed by the DOC method outperform other deployment strategies.

Table 8.1: Performance comparison of sensor deployment strategies.

Deployment Strategy	Cost Function, J	Track Detections ($k = 2$)
DOC	-13.801	998
Uniform	-5.034	488
Grid	-6.254	549
Random	-3.951	454

The second example considers a network of $N = 250$ sensors with nonlinear unicycle kinematics in (8.16). In this case, however, the sensors are deployed in an obstacle-populated workspace $\mathcal{W} = [0, L] \times [0, L]$ shown in Fig. 8.8, with $L = 16$ km, $T_0 = 0$, and $T_f = 15$ hr. The number of independent elementary detections required to declare a target track detection is $k = 3$, and the initial sensor distribution, φ_0 , is shown in Fig. 8.8.a. The PDF of the initial target position is plotted in Fig. 8.8, and is modeled by the Gaussian mixture,

$$f_{\mathbf{x}_{T_0}}(\mathbf{x}_{T_0}) = \sum_{\ell=1}^3 \frac{w_\ell}{(2\pi)^{n/2} \det(\Sigma_\ell)^{1/2}} e^{[-(1/2)(\mathbf{x}_{T_0} - \boldsymbol{\mu}_\ell)^T \Sigma_\ell^{-1} (\mathbf{x}_{T_0} - \boldsymbol{\mu}_\ell)]} \quad (8.20)$$

with means $\boldsymbol{\mu}_1 = [10 \ 6]^T$ km, $\boldsymbol{\mu}_2 = [10 \ 9]^T$ km, and $\boldsymbol{\mu}_3 = [12.5 \ 7.5]^T$ km, and covariances $\Sigma_1 = \mathbf{I}_2$, $\Sigma_2 = 1.5 \cdot \mathbf{I}_2$, and $\Sigma_3 = 3 \cdot \mathbf{I}_2$. The mixing proportions are

$w_1 = 0.2$, $w_2 = 0.2$, and $w_3 = 0.6$. The PDF of the target heading distribution, shown in Fig. 8.9, is

$$f_{\theta_T}(\theta_T) = \begin{cases} 1.43 \text{ (rad}^{-1}\text{)}, & \text{if } 0.70 \leq \theta_T \leq 1.43 \text{ (rad)} \\ & \text{or } 2.18 \leq \theta_T \leq 2.53 \text{ (rad)} \\ 0, & \text{otherwise} \end{cases} \quad (8.21)$$

As in the first example, the target speed is assumed constant and equal to $v_T = 0.2$ km/hr.

The cost function weights are $w_s = 1$, $w_r = 0.02$, and $w_e = 0.1$, to include the obstacle-avoidance objective. The number of mixture components is $z = 6$, and the state space is discretized into $X = 900$ collocation points, and $\Delta t = 1$ hr, such that $K = 15$. The optimal sensor PDF obtained by the DOC method is plotted in Fig. 8.11, along with the sensors microscopic state and FOVs, at four instants in time. From the time-history of the Lagrangian, evaluated at each time step and plotted in Fig. 8.12, it can be seen that the instantaneous cost is reduced over time, as the sensor network maximizes its probability of track detection (8.14), avoid obstacles, and minimizes the energy consumption.

A performance comparison was also completed for this case, where grid, random, and uniform deployment strategies, plotted in Fig. 8.13, were modified for the workspace \mathcal{W} containing geometric obstacles. The grid formation has $z = 17$ Gaussian mixture components, where the components overlapping the obstacles were removed, and the random deployment uses $z = 20$ components with means randomly generated with the constraint that they be at least a distance r from the nearest obstacle. The uniform deployment uses a sensor distribution defined uniformly over \mathcal{W} . As with the previous example, the sensor states are obtained via finite-mixture sampling from each distribution and are assumed to be constant over time. The sample positions are also required to be at a distance greater than r from the nearest obstacle.

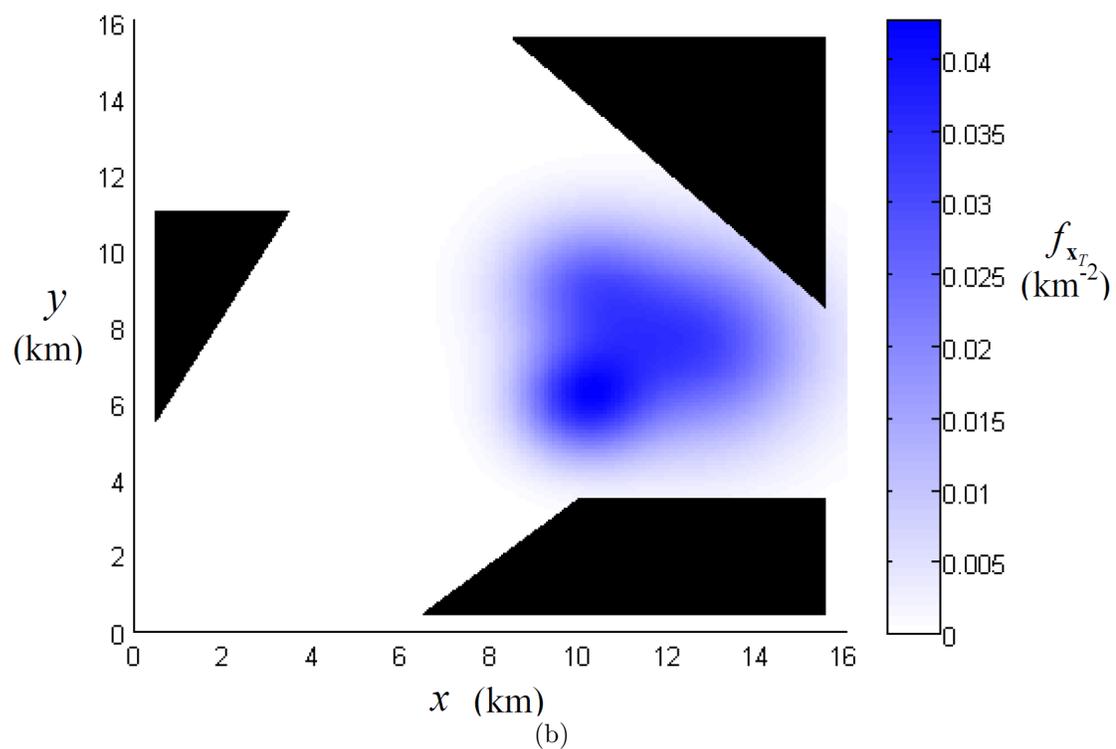
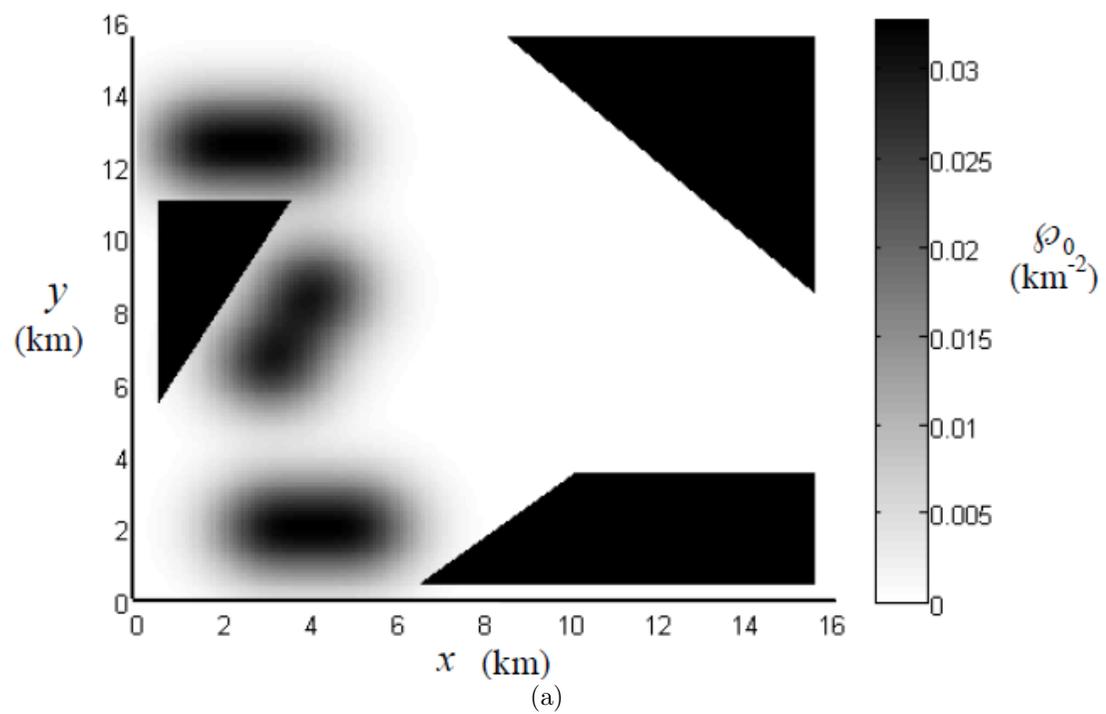


FIGURE 8.8: Initial sensor distribution (a) and PDF of initial target position for example with obstacles (solid black).

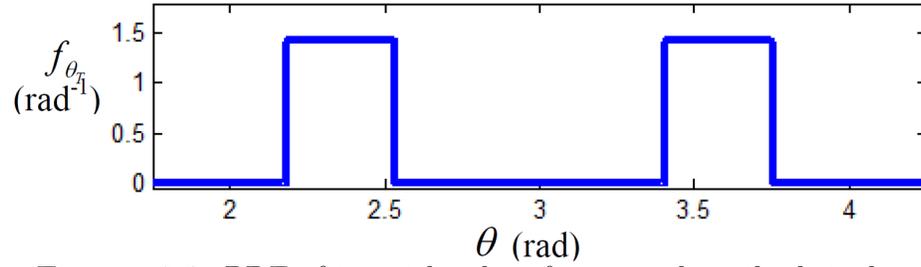


FIGURE 8.9: PDF of target heading for example with obstacles.

In addition, a stochastic gradient approach [58] is simulated for comparison. Stochastic gradient methods use a similar concept as classical potential function methods for feedback motion planning, where a gradient descent is performed on a function to compute a path from an initial state to a goal configuration, and they can typically be implemented in environments with uncertain dynamics or measurements.

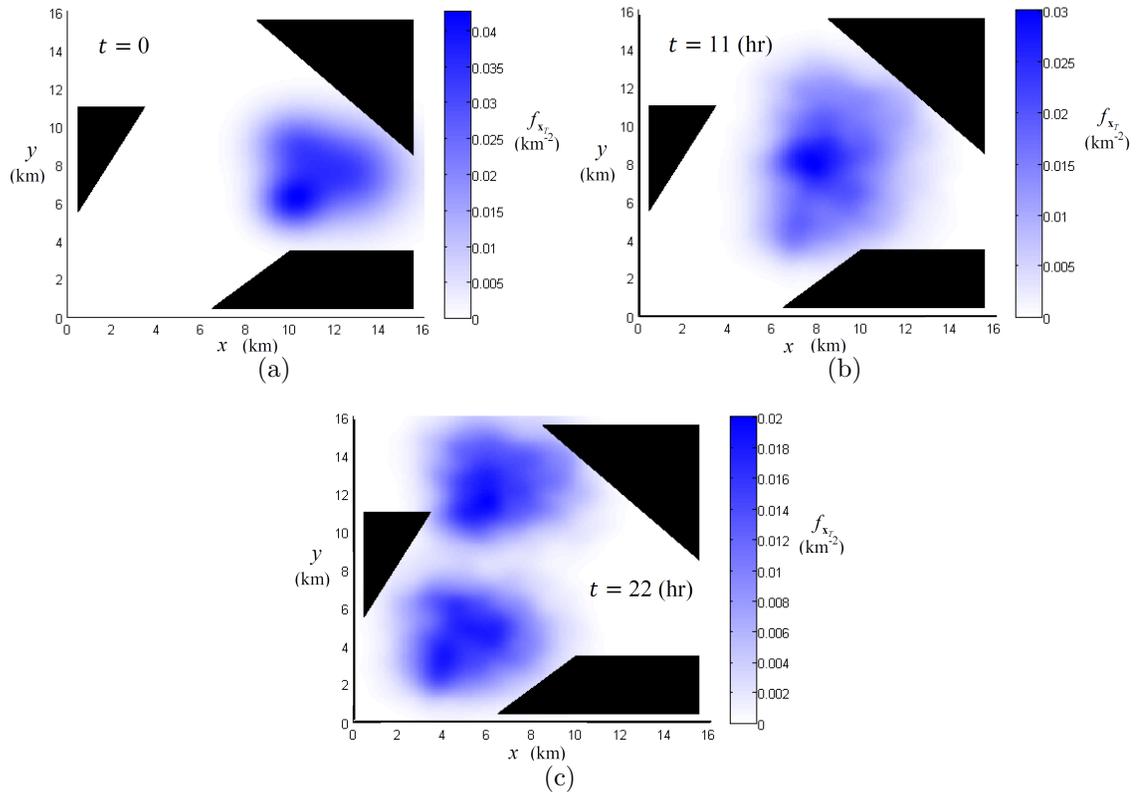


FIGURE 8.10: Evolution of target PDF at three instants in time for example with obstacles (solid black) in Figs. 8.8-8.9.

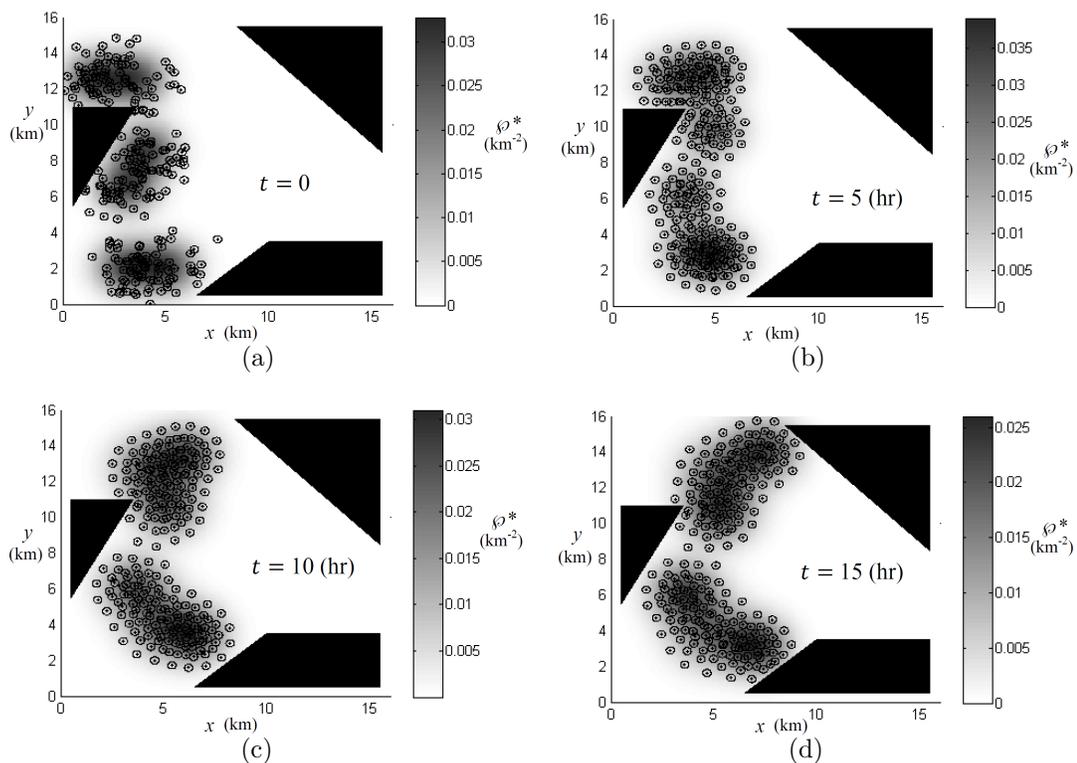


FIGURE 8.11: Evolution of optimal sensor PDF, φ^* , microscopic state (black dots), and FOVs (black circles) at four instants in time, for example with obstacles (solid black) in Figs. 8.8-8.9.

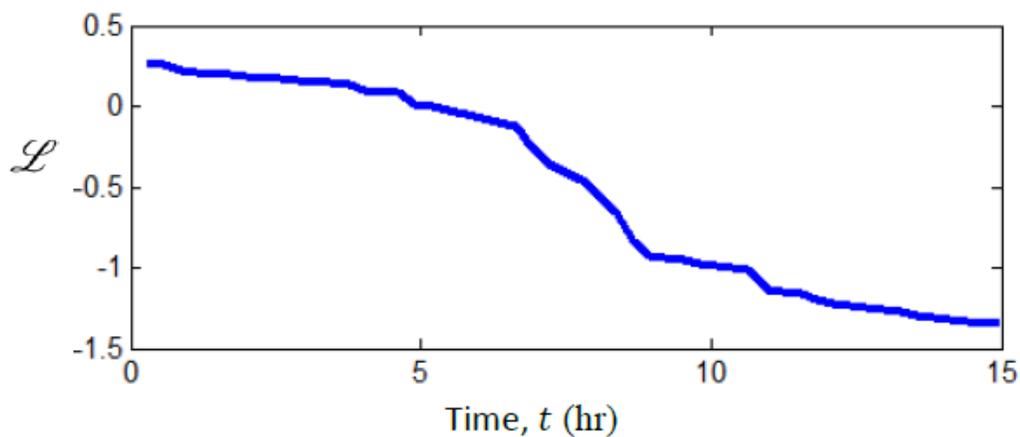


FIGURE 8.12: Time-history of cost function Lagrangian for DOC solution in Fig. 8.11.

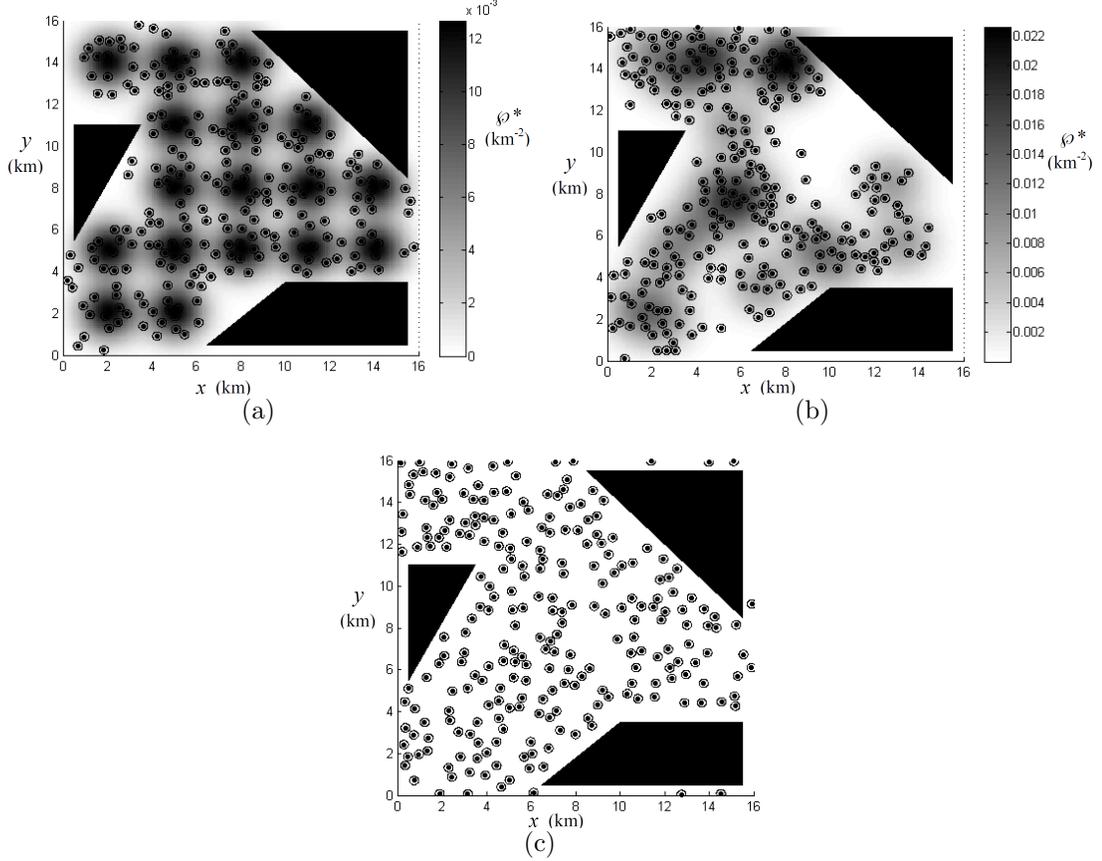


FIGURE 8.13: Grid (a), random (b), and uniform (c) sensor distributions with sampled sensor positions (black dots), FOVs (black circles), and geometric obstacles (solid black).

For this example, the goal states are sampled from a time-invariant goal sensor PDF, plotted in Fig. 8.14, that minimizes the cost function (8.15) over only the instant in time $t = T_f$ and without the dynamic constraints. The initial sensor states are sampled from φ_0 . Each sensor seeks to move toward the closest goal state, \mathbf{s} , that is not already occupied by another sensor. Adding a repulsive potential term, U_{rep} , for obstacle avoidance to the state update law given in [58], a microscopic feedback control law for use with the vehicle dynamics from (8.16) can be formulated as (7.5) with $U = w_a \|\mathbf{s} - \mathbf{q}\| + w_b U_{rep}$, where $w_a = 1$ and $w_b = 2.5$ are weighting constants. The states of the sensors simulated using the stochastic gradient method are plotted in Fig. 8.15 at three instants in time, along with the goal sensor PDF and static goal

microscopic states.

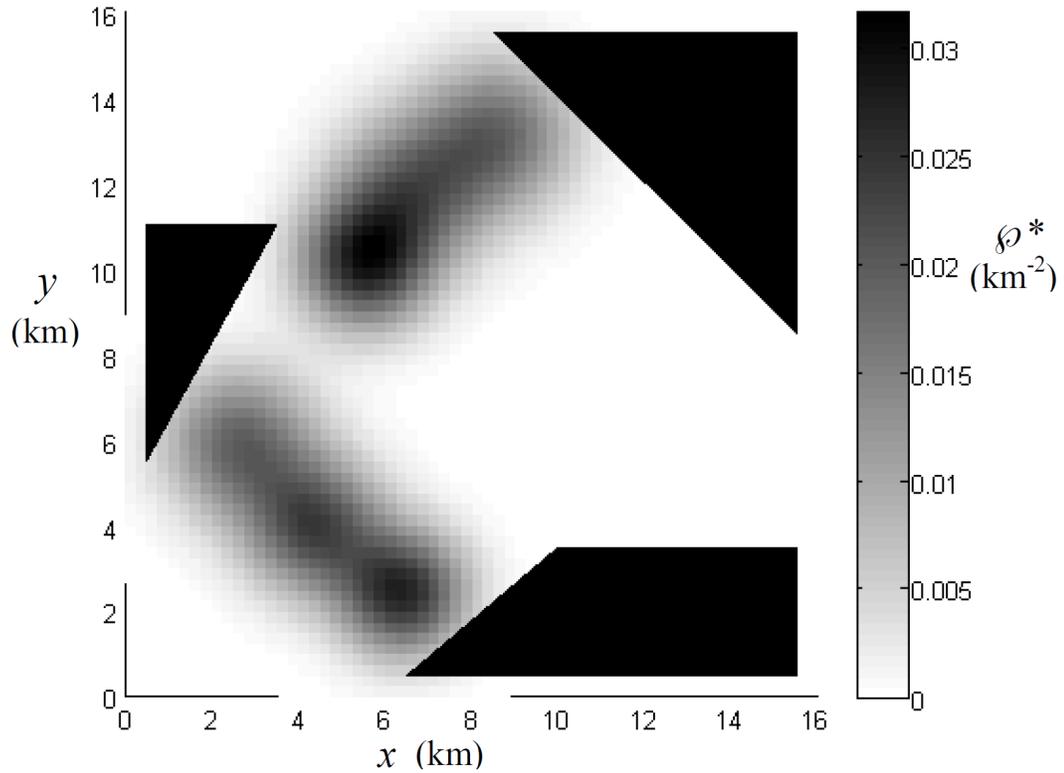


FIGURE 8.14: Goal sensor PDF from which the goal microscopic states were sampled for the stochastic gradient method.

The cost function (8.15) and the actual number of target track detections were computed, averaging over twenty simulations for each method. To evaluate (8.15) for the stochastic gradient method, a PDF was constructed from the microscopic sensor states at each timestep using kernel density estimation with a standard Gaussian kernel. The results are summarized in Table 8.2, and the DOC approach is shown to outperform the other strategies.

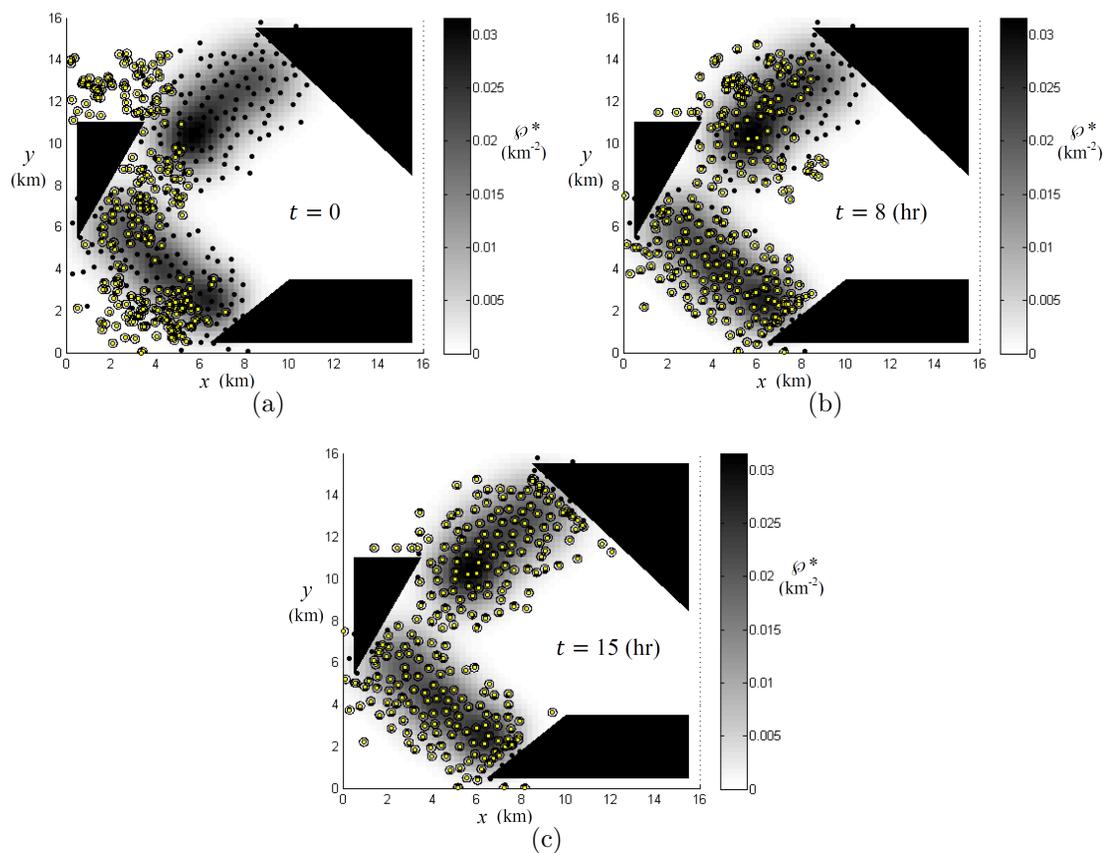


FIGURE 8.15: States of $N = 250$ sensors (solid yellow circles) with FOVs (black circles) controlled using the stochastic gradient approach and plotted at three instants in time. The sensors travel toward goal states (black dots) sampled from a goal sensor PDF (plotted on background).

Table 8.2: Performance comparison of sensor deployment strategies for example case with obstacles.

Deployment Strategy	Cost Function, J	Track Detections ($k = 3$)
DOC	-5.956	933
Stochastic Gradient	-0.884	847
Uniform	3.196	387
Grid	2.639	421
Random	2.844	395

Conclusions

This dissertation presents a novel distributed optimal control problem formulation that is applicable to multiscale dynamical systems comprised of numerous interacting systems, or agents, that together give rise to coherent macroscopic behaviors, or coarse dynamics, that can be modeled by partial differential equations on larger spatial and time scales. The DOC methodology seeks to obtain optimal agent state and control trajectories by representing the system's performance as an integral cost function of the macroscopic state, which is optimized subject to the agents' dynamics. The macroscopic state is identified as a time-varying probability density function to which the states of the individual agents can be mapped via a restriction operator. Optimality conditions for the DOC problem are derived analytically, and the optimal trajectories of the macroscopic state and control are computed using direct and indirect optimization algorithms. Feedback microscopic control laws are then derived from the optimal macroscopic description using a potential function approach.

The DOC approach is demonstrated numerically through benchmark multi-agent trajectory optimization problems, where large systems of agents were given the ob-

jectives of traveling to goal state distributions, avoiding obstacles, maintaining formations, and minimizing energy consumption through control. Comparisons are provided between the direct and indirect optimization techniques, as well as existing methods from the literature, and a computational complexity analysis is presented. The indirect method is demonstrated to have a higher performance and shorter runtime than the direct approach due to its greater flexibility of PDF approximations and its availability of an analytical gradient of the cost function. The methodology is also applied to a track coverage optimization problem for the control of distributed networks of mobile omnidirectional sensors, where the sensors move to maximize the probability of track detection of a known distribution of mobile targets traversing a region of interest (ROI). Through extensive simulations, DOC is shown to outperform several existing sensor deployment and control strategies. Furthermore, the computation required by the DOC algorithm is proven to be far reduced compared to that of classical, direct optimal control algorithms.

The recommended future work is as follows. A fully decentralized DOC methodology can be developed that includes a decentralized optimization algorithm in addition to the decentralized microscopic control law. Systems with heterogeneous agents can be investigated, where several classes of agents may exist, each governed by unique dynamic models. Non-conservative problems can be considered, where the introduction and removal of agents is a property of the environment or part of the control. The applicability of DOC for varying number of agents is also an area of interest, where the effectiveness of the DOC methodology for small numbers of agents can be evaluated. Through alternative numerical methods, such as two-stage approaches or approximate dynamic programming, the numerical properties of the optimization algorithms can be improved, including the avoidance of local minima, lowering runtime, and reducing memory required. DOC can be applied to a number of applications, such as traffic control or resource allocation in response to disasters.

Bibliography

- [1] E. U. Acar. Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods. *International Journal of Robotic Research*, 22:7–8, 2003.
- [2] A. Alvarez, A. Caiti, and R. Onken. Evolutionary path planning for autonomous underwater vehicles in a variable ocean. *IEEE Journal of Oceanic Engineering*, 29(2):418–429, 2004.
- [3] N. Ayanian and V. Kumar. Abstractions and controllers for groups of robots in environments with obstacles. *Proc. IEEE International Conference on Robotics and Automation, Anchorage, Alaska*, 2010.
- [4] K. Bajer. Hamiltonian formulation of the equations of streamlines in three-dimensional steady flows. *Chaos, Solitons and Fractals*, 4(6):895–911, 1994.
- [5] Y. Bar-Shalom and W. D. Blair. *Multitarget-Multisensor Tracking: Applications and Advances, Vol. III*. Artech House, 2000.
- [6] J. Barraquand, B. Langlois, and J. C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(2):224–241, 1992.
- [7] K. C. Baumgartner and S. Ferrari. A geometric transversal approach to analyzing track coverage in sensor networks. *IEEE Transactions on Computers*, 57(8):1113–1128, 2008.
- [8] K. C. Baumgartner, S. Ferrari, and A.V. Rao. Optimal control of an underwater sensor network for cooperative target tracking. *IEEE Journal of Oceanic Engineering*, 34(4):678–697, Oct. 2009.
- [9] K. C. Baumgartner, S. Ferrari, and T. A. Wettergren. Robust deployment of dynamic sensor networks for cooperative track detection. *IEEE Sensors*, 9(9):1029–1048, 2009.
- [10] M. Bennewitz, W. Burgard, and S. Thrun. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robots and Autonomous Systems*, 41(2):89–99, 2002.

- [11] S. Berman, A. Halasz, M. A. Hsieh, and V. Kumar. Navigation-based optimization of stochastic strategies for allocating a robot swarm among multiple sites. *Proc. Conference on Decision and Control*, pages 4376–4381, 2008.
- [12] S. Berman, A. Halasz, M. A. Hsieh, and V. Kumar. Optimized stochastic policies for task allocation in swarms of robots. *IEEE Transactions on Robotics*, 25(4):927–937, 2009.
- [13] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 2007.
- [14] J. T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998.
- [15] L. T. Biegler. *Large-Scale PDE-Constrained Optimization*. Lecture Notes in Computational Science and Engineering, 30. Springer Verlag, 2003.
- [16] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur Methods for PDE-Constrained Optimization. Part I: The Krylov-Schur Solver. *SIAM Journal on Scientific Computing*, 27(2):687–713, 2005.
- [17] J. P. Boyd. *Chebyshev and Fourier Spectral Methods, II Ed*. Dover, New York, NY, 2001.
- [18] S. J. Buckley. Fast motion planning for multiple moving robots. *Proceedings of IEEE International Conference on Robotics and Automation*, pages 322–326, 1989.
- [19] C. C. Cheah, S. P. Hou, and J. J. E. Slotine. Region-based shaped control for a swarm of robots. *Automatica*, 45:2406–2411, 2009.
- [20] H. Choset. Coverage for robotics: A survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):113–126, 2001.
- [21] M. Chu, H. Haussecker, and F. Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *International Journal of High Performance Computing Applications*, 16(3):293–313, 2002.
- [22] J. Cortes, C. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.
- [23] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., 1991.
- [24] H. Cox. Cumulative detection probabilities for a randomly moving source in a sparse field of sensors. In *Proc. Asilomar Conference*, pages 384–389. 1989.

- [25] R. Daily and D. M. Bevly. Harmonic potential field path planning for high speed vehicles. *Proc. American Control Conference, Seattle, WA*, pages 4609–4614, 2008.
- [26] C. Ferrari, E. Pagello, J. Ota, and T. Arai. Multirobot motion coordination in space and time. *Robotics and Autonomous Systems*, 25:219–229, 1998.
- [27] S. Ferrari, G. Zhang, and T. A. Wettergren. Probabilistic track coverage in cooperative sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 40(6), 2010.
- [28] G. Foderaro, S. Ferrari, and T. A. Wettergren. Distributed optimal control for multi-agent trajectory optimization. *Automatica, in press*, 2013.
- [29] C. Fox. *An Introduction to the Calculus of Variations*. Dover Publications, Inc., New York, 1987.
- [30] V. Gazi and K. M. Passino. Stability analysis of social foraging swarms. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(1):539–557, 2004.
- [31] S. Ge and Y. Cui. New potential functions for mobile robot path planning. *IEEE Transactions on Robotics and Automation*, 16(5):615–620, 2000.
- [32] I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Prentice-Hall, Englewood Cliffs, NJ, 1963.
- [33] H. Gonzales-Banos and J. C. Latombe. Robot motion planning with uncertainty in control and sensing. *17th Annual Symposium on Computational Geometry*, 2001.
- [34] Y. Guo and L. E. Parker. A distributed and optimal motion planning approach for multiple mobile robots. *IEEE International Conference on Robotics and Automation*, 2002.
- [35] J. E. Hopcroft, J. T. Schwartz, and M. Sharir. On the complexity of motion planning for multiple independent objects; PSPACE-hardness of the “warehouseman’s problem”. *The International Journal of Robotics Research*, 3(4):76–88, 1984.
- [36] Y. Hu, H. Chen, J. Lou, and J. Li. Distributed density estimation using non-parametric statistics. *International Conference on Distributed Computing Systems*, 2007.
- [37] T. Inanc, S. C. Shadden, and J. E. Marsden. Optimal trajectory generation in ocean flows. In *Proceedings of the 2005 American Control Conference*, pages 674–679, Portland, OR, 2005.

- [38] K. Kant and S. W. Zucker. Toward efficient trajectory planning: the path-velocity decomposition. *The International Journal of Robotics Research*, 5(3):72–89, 1986.
- [39] G. Van Keuk. Sequential track extraction. *IEEE Transactions on Aerospace and Electronic Systems*, 34(4):1135–1148, 1998.
- [40] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg, and C. Theodoropoulos. Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1(4):715–762, 2003.
- [41] D. E. Kirk. *Optimal Control Theory: An Introduction*. Prentice-Hall, Englewood Cliffs, NJ, 1970.
- [42] Z. Kone, E. G. Rowe, and T. A. Wettergren. Sensor repositioning to improve undersea sensor field coverage. In *Proceedings of OCEANS 2007*, pages 1–6, 2007.
- [43] B. O. Koopman. *Search and Screening: General Principles with Historical Applications*. Pergamon Press, 1980.
- [44] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [45] S. M. LaValle and S. A. Hutchinson. Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation*, 14:912–925, 1998.
- [46] J. P. LeCadre and G. Souris. Searching tracks. *IEEE Transactions on Aerospace and Electronic Systems*, 36(4):1149–1166, 2000.
- [47] P. F. J. Lermusiaux. Evolving the subspace of the three-dimensional multiscale ocean variability: Massachusetts bay. *Journal of Marine Systems*, 29:385–422, 2001.
- [48] X. Liao and L. Carin. Application of the theory of optimal experiments to adaptive electromagnetic-induction sensing of buried targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):961–972, 2004.
- [49] Y. Liu and K. M. Passino. Stable social foraging swarms in a noisy environment. *IEEE Transactions on Automatic Control*, 49(1):30–44, 2004.
- [50] R. C. Loxton, K. L. Teo, and V. Rehbock. Optimal control problems with multiple characteristic time points in the objective and constraints. *Automatica*, 44(11):2923–2929, 2008.

- [51] O. Babaoglu M. Jelasity, A. Montresor. Gossip-based aggregation in large dynamic networks. *ACM Transactions on Computer Systems*, 23(3):219–252, 2005.
- [52] X. Mao. *Stochastic Differential Equations and Applications*. Horwood Publishing, 1997.
- [53] S. Martinez, J. Cortes, and F. Bullo. Motion coordination with distributed information. *IEEE Control Systems Magazine*, 27(4):75–88, 2007.
- [54] A. A. Masoud. Managing the dynamics of a harmonic potential field-guided robot in a cluttered environment. *IEEE Transactions on Industrial Electronics*, 56(2):488–496, 2009.
- [55] Mathworks. *Matlab Optimization Toolbox*. [Online]. Available: <http://www.mathworks.com>, 2004. function: fmincon.
- [56] G. McLachlan. *Finite Mixture Models*. Wiley Interscience, New York, NY, 2000.
- [57] S. Megerian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Worst and best-case coverage in sensor networks. *IEEE Transactions on Mobile Computing*, 4(2):84–92, 2005.
- [58] J. Le Ny and G. J. Pappas. Sensor-based robot deployment algorithms. *IEEE Conference on Decision and Control*, pages 5486–5492, 2010.
- [59] J. Oyekan and H. Hu. Ant robotic swarm for visualizing invisible hazardous substances. *Robotics*, 2:1–18, 2013.
- [60] D. Parsons and J. Canny. A motion planner for multiple mobile robots. *IEEE International Conference on Robotics and Automation*, pages 8–13, 1990.
- [61] M. Peasgood, C. Clark, and J. McPhee. A complete and scalable strategy for coordination of multiple robots with roadmaps. *IEEE Transactions on Robotics*, 24(2):283–292, 2008.
- [62] M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. *Numerical Analysis*, 630, 1978.
- [63] H. Qi, S. Iyengar, and K. Chakrabarty. Distributed sensor networks—a review of recent research. *Journal of the Franklin Institute*, 338(6):655–68, 2001.
- [64] J. H. Reif and H. Wang. Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems*, 27(3):171–194, 1999.
- [65] J. Ren and K. Mclsaac. A hybrid-systems approach to potential field navigation for a multi-robot team. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 3875–3880, Taipei, Taiwan, 2003.

- [66] E. Rich. *Automata, Computability and Complexity*. Pearson Education, Inc., 2008.
- [67] J. Sanchirico and J. Wilen. Optimal spatial management of renewable resources: Matching policy scope to ecosystem scale. *Journal of Environmental Economics and Management*, 50:23–46, 2005.
- [68] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley-Interscience, 1992.
- [69] S. Shimoda, Y. Kuroda, and K. Iagnemma. Potential field navigation of high speed unmanned ground vehicles on uneven terrain. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 2839–2844, Barcelona, Spain, 2005.
- [70] J. R. Sibert and J. L. Nielsen. *Electronic Tagging and Tracking in Marine Fisheries*. Kluwer Academic Publisher, 2001.
- [71] J. S. Simonoff. *Smoothing Methods in Statistics*. Springer, 1996.
- [72] I. Spangelo and O. Egeland. Trajectory planning and collision avoidance for underwater vehicles using optimal control. *IEEE Journal of Oceanic Engineering*, 19(4):502–511, 1994.
- [73] M. F. M. Speetjens. Topology of advective-diffusive scalar transport in laminar flows. *Physical Review E*, 77(2), 2008.
- [74] J. R. Spletzer and C. J. Taylor. Dynamic sensor planning and control for optimally tracking target. *International Journal of Robotics Research*, 22(1):7–20, 2003.
- [75] R. F. Stengel. *Optimal Control and Estimation*. Dover Publications, Inc., 1986.
- [76] W. Sun and Y. X. Yuan. *Optimization Theory and Methods: Nonlinear Programming*. Springer Optimization and Its Applications Series. Springer Science+Business Media, LLC, 2006.
- [77] J. C. Tannehille, D. A. Anderson, and R. H. Pletcher. *Computational Fluid Mechanics and Heat Transfer*. Taylor and Francis, 1997.
- [78] R. J. Urick. *Principles of Underwater Sound*. McGraw-Hill Book Company, 3 edition, 1983.
- [79] L. Y. Wang, W. H. Gui, K. L. Teo, R. Loxton, and C. H. Yang. Time delayed optimal control problems with multiple characteristic time points: Computation and industrial applications. *Journal of Industrial and Management Optimization*, 5(4):705–718, 2009.

- [80] C. W. Warren. Multiple robot path coordination using artificial potential fields. *Proceedings of IEEE International Conference on Robotics and Automation*, pages 500–505, 1990.
- [81] H. Wei and S. Ferrari. A geometric transversals approach to analyzing the probability of track detection for maneuvering targets. *IEEE Transactions on Computers*, *in press*, 2013.
- [82] T. A. Wettergren. Performance of search via track-before-detect for distributed sensor networks. *IEEE Transactions on Aerospace and Electronic Systems*, 44(1):314–325, 2008.
- [83] T. A. Wettergren and R. Costa. Optimal placement of distributed sensors against moving targets. *ACM Transactions on Sensor Networks*, 5(3), 2009.
- [84] T. A. Wettergren, R. L. Streit, and J. R. Short. Tracking with distributed sets of proximity sensors using geometric invariants. *IEEE Transactions on Aerospace and Electronic Systems*, 40(4):1366–1374, October 2004.
- [85] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas. Graph-theoretic connectivity control of mobile robot networks. *Proceedings of the IEEE*, 99(9):1525–1540, 2011.

Biography

Greg Foderaro was born in June 1986 in Sparta, New Jersey. He attended Clemson University and earned his BS in Mechanical Engineering. From there, Greg decided to pursue his Ph.D. under the supervision of Dr. Silvia Ferrari, and he immediately joined the Laboratory for Intelligent Systems and Controls at Duke University. During his study at Duke, his research interests have focused on distributed optimal control, robot path planning, underwater sensor networks, spiking neural networks, and intelligent agents in games.