2014

Optimal control approaches for persistent monitoring problems.

Lin, Xuchao

Boston University

http://hdl.handle.net/2144/11119

Boston University
BOSTON UNIVERSITY
COLLEGE OF ENGINEERING

Dissertation

OPTIMAL CONTROL APPROACHES FOR PERSISTENT MONITORING PROBLEMS

by

XUCHAO LIN
B.Eng., Zhejiang University, 2009
M.S., Boston University, 2012

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
2014
To Father, Mother and Peiyuan
Acknowledgments

Throughout my five years as a PhD student in the Control of Discrete-Event Systems (CODES) Lab in Boston University, many people have provided me support and assistance, without whom this dissertation would not exist.

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Christos G. Cassandras for guiding me throughout the whole research process. His patience in listening to my immature thoughts and helping me integrate and lead them in the right direction is the foundation of my research progress. I will always be indebted to him for what I learnt from his leadership, professionalism and meticulousness of work, all of which will be invaluable treasures to my entire life.

I also sincerely thank my thesis committee members, Prof. Calin Belta, Prof. Ioannis Paschalidis, Prof. Mac Schwager and Prof. Pirooz Vakili for their time and constructive comments on my thesis and research. I would like to thank Prof. Venkatesh Saligrama and Prof. David Castanon for their help and guidance during earlier years of my PhD study. I would also like to show my gratitude to Ms. Elizabeth Flagg and Ms. Cheryl Stewart for what have made my academic journey at Boston University “a breeze” without administrative hurdles.

In addition, during my stay in Boston University, I have received much help from my fellow colleagues. The first to thank are the seniors: Tao Wang, Xu Chu Ding, Yanfeng Geng, Ali Kebarighotbi, Jiefu Zheng and Jing Qian. With their own experiences, they have helped me find the right track to accomplish my study. I also want to thank Yasaman Khazaeni, Julia Fleck, Sepideh Pourazarm, Xinmiao Sun, Xiaodong Lan, Bowen Zhang, Jing Wang, Cong Wei, Jizong Shang, Ruoseng Xiong and Yuan Zheng. We shared research ideas, thoughts of life and had fun together.

Last but not least, I owe my deepest gratitude to my father Jingzong Lin and my mother Lijuan Fu. They gave me a healthy body and mind full of curiosity, which are my most valuable assets. Throughout my life, they continuously provide me with material and mental support without any reservation, and give me their love unconditionally. I also want
to thank my beautiful, lovely wife - Peiyuan Wang. She joined my life two years ago, after which her constant accompany with shared minds, love and food gives me a firm support to fight for whatever I want. Her angel-like smile always reminds me how lucky I am. I dedicate this dissertation to them.
OPTIMAL CONTROL APPROACHES FOR PERSISTENT MONITORING PROBLEMS

XUCHAO LIN

Boston University, College of Engineering, 2014

Major Professor: Christos G. Cassandras, PhD,
Professor of Systems Engineering,
Professor of Electrical and Computer Engineering

ABSTRACT

Persistent monitoring tasks arise when agents must monitor a dynamically changing environment which cannot be fully covered by a stationary team of available agents. It differs from traditional coverage tasks due to the perpetual need to cover a changing environment, i.e., all areas of the mission space must be visited infinitely often. This dissertation presents an optimal control framework for persistent monitoring problems where the objective is to control the movement of multiple cooperating agents to minimize an uncertainty metric in a given mission space. In an one-dimensional mission space, it is shown that the optimal solution is for each agent to move at maximal speed from one switching point to the next, possibly waiting some time at each point before reversing its direction. Thus, the solution is reduced to a simpler parametric optimization problem: determining a sequence of switching locations and associated waiting times at these switching points for each agent. This amounts to a hybrid system which is analyzed using Infinitesimal Perturbation Analysis (IPA), to obtain a complete on-line solution through a gradient-based algorithm. IPA is a method used to provide unbiased gradient estimates of performance metrics with respect to various controllable parameters in Discrete Event Systems (DES) as well as in Hybrid Systems (HS). It is also shown that the solution is robust with respect to the uncertainty model used, i.e., IPA provides an unbiased estimate of the gradient without any detailed knowledge of how uncertainty affects the mission space.

In a two-dimensional mission space, such simple solutions can no longer be derived.
An alternative is to optimally assign each agent a linear trajectory, motivated by the one-dimensional analysis. It is proved, however, that elliptical trajectories outperform linear ones. With this motivation, the dissertation formulates a parametric optimization problem to determine such trajectories. It is again shown that the problem can be solved using IPA to obtain performance gradients on line and obtain a complete and scalable solution. Since the solutions obtained are generally locally optimal, a stochastic comparison algorithm is incorporated for deriving globally optimal elliptical trajectories. The dissertation also approaches the problem by representing an agent trajectory in terms of general function families characterized by a set of parameters to be optimized. The approach is applied to the family of Lissajous functions as well as a Fourier series representation of an agent trajectory. Numerical examples indicate that this scalable approach provides solutions that are near-optimal relative to those obtained through a computationally intensive two point boundary value problem (TPBVP) solver. In the end, the problem is tackled using centralized and decentralized Receding Horizon Control (RHC) algorithms, which dynamically determine the control for agents by solving a sequence of optimization problems over a planning horizon and executing them over a shorter action horizon.
# Contents

1 Introduction ................................................. 1
   1.1 Persistent Monitoring Problem Basics .................. 1
   1.2 Optimal Control Approaches ............................. 6
   1.3 Infinitesimal Perturbation Analysis .................... 9
   1.4 Receding Horizon Control ............................... 10
   1.5 Contributions of The Dissertation ..................... 12
   1.6 Dissertation Organization .............................. 13

2 1-dimensional Persistent Monitoring Problem ............. 16
   2.1 1D Persistent Monitoring Problem Formulation .......... 17
   2.2 Optimal Control Solution ................................ 20
   2.3 Infinitesimal Perturbation Analysis (IPA) ............... 30
      2.3.1 Single-agent solution with $a = 0$ and $b = L$ .......... 30
      2.3.2 Multi-agent solution where $a \geq 0$ and $b \leq L$ .......... 37
   2.4 Objective Function Optimization ........................ 43
   2.5 Numerical Experiments ................................... 45
   2.6 Soft constraint case with $R_i \leq R_i^{\text{max}}$ ............ 49
      2.6.1 Objective Function Gradient Evaluation ............... 52
      2.6.2 Numerical Experiments ............................... 54
   2.7 Summary .................................................. 55

3 Elliptical Trajectories for 2-dimensional Persistent Monitoring Problem 57
   3.1 2D Persistent Monitoring Problem Formulation ............ 59
   3.2 Optimal Control Solution ............................... 61
   3.3 Linear vs Elliptical Agent Trajectories ................... 66
3.4 Optimal Elliptical Trajectories .............................................. 74
  3.4.1 Infinitesimal Perturbation Analysis (IPA) ......................... 75
  3.4.2 Objective Function Optimization ..................................... 79
3.5 Stochastic Comparison Algorithm for global optimality ............. 79
3.6 Numerical Experiments ..................................................... 81
3.7 Summary ........................................................................ 83

4 Trajectories Characterized by General Functions for 2-dimensional Persistent Monitoring Problem 87
  4.1 Trajectory Optimization Using General Function Families Formulation ... 88
  4.2 Lissajous and Fourier Series Representations of Agent Trajectories .... 93
  4.3 Numerical Experiments .................................................... 98
  4.4 Summary ................................................................... 101

5 Receding Horizon Control for Persistent Monitoring Problem 102
  5.1 RHC for 1D Persistent Monitoring Problem .......................... 103
  5.2 RHC for 2D Persistent Monitoring Problem .......................... 107
  5.3 Numerical Experiments .................................................... 109
    5.3.1 RHC for 1-dimensional Mission Space ......................... 109
    5.3.2 RHC for 2-dimensional Mission Space ......................... 110
  5.4 Summary ................................................................... 113

6 Conclusions and Future Work ................................................. 114

References ......................................................................... 117

Curriculum Vitae ............................................................... 122
List of Figures

2.1 A queueing system analog of the persistent monitoring problem. ....... 20
2.2 Hybrid automaton for each $\alpha_i$. Red arrows represent events when the control switches between 1 and 1. Blue arrows represent events when $R_i$ becomes 0. Black arrows represent all other events. .................... 32
2.3 One agent example. $L = 20, T = 400$. For each example, top plot: optimal trajectory; bottom plot: $J$ versus iterations. ...................... 47
2.4 Magnified trajectory for sub-figure (a) and (b) in Fig. 2-3, $t \in [0, 75]$. .... 48
2.5 Two agent example. $L = 40, T = 400$. Top plot: optimal trajectory. Bottom plot: $J$ versus iterations. .......................... 48
2.6 $\tau_e$ is the event time when the system enters a region violating the constraint $R_i(t) \leq R_i^{\text{max}}$ and $\tau_f$ is the event time when the system leaves this region. .... 52
2.7 Two agent example for $\beta_i = 0, i = 0, 1, \ldots, 40$. Left top plot: cost $J(\theta^i, w^i)$ as a function of iteration number with $J_{\beta=0}(\theta_{\beta=0}^*, w_{\beta=0}^*) = 54.59$; left bottom plot shows the optimal agent trajectories. Right plot: Selected $R_i(t)$ vs. time. Red lines are the $R_i^{\text{max}}$ constraint. ............... 55
2.8 Two agent example for $\beta_i = 10, i = 0, 1, \ldots, 40$. Left top plot: cost $J(\theta^i, w^i)$ as a function of iteration number with $J_{\beta=10}(\theta_{\beta=10}^*, w_{\beta=10}^*) = 530.1$, red line $J_{\beta=10}(\theta_{\beta=0}^*, w_{\beta=0}^*) = 1861$ represents the cost using the agent trajectories with $\beta_i = 0$ and cost function with $\beta_i = 10, i = 0, \ldots, 40$; left bottom plot shows the optimal agent trajectories. Right plot: Selected $R_i(t)$ vs. time. Red lines are the $R_i^{\text{max}}$ constraint. .......... 56
3.1 Two agent example for $19 \times 19$ 2-dimensional rectangular space for agents have different initial positions and diagonal initial trajectories. .......... 66

xi
3.2 The red ellipse represents the agent trajectory. The area defined by the black curve is the agent's effective coverage area. \[ \frac{ab}{\sqrt{b^2 \cos^2(\theta) + a^2 \sin^2(\theta)}} + \gamma(\theta) \] is the distance between the ellipse center and the coverage area boundary for a given \( \theta \).  

3.3 Optimal elliptical trajectories for two agents (without using the CSC algorithm).  

3.4 Optimal trajectories using TPBVP solver for two agents. Initial trajectories are red curves in Fig. 3.3(a).  

3.5 Two agent example for the deterministic environment setting using the CSC Algorithm 1 for \( Q = 300 \) trials.  

3.6 Two-agent example for a stochastic environment setting using the CSC Algorithm 1 for \( Q = 300 \) trials, where \( A_i (\Delta t) \sim U(0.195, 0.205), \Delta t \sim 0.2t \).  

3.7 Left plot: elliptical trajectories (blue curve) obtained in Fig. 3.5(a) used as initial trajectories for the TPBVP solver.  

4.1 Lissajous trajectories. Top left: initial trajectories assigned for two agents. Top right, final trajectories obtained solving Problem P4.1. Bottom: cost vs. number of iterations. \( J^* = 1612 \).  

4.2 Fourier Series trajectories. Top left: initial trajectories assigned for two agents. Top right, final trajectories obtained solving Problem P4.1. Bottom: cost vs. number of iterations. \( J^* = 1612 \).  

4.3 Elliptical trajectories. Top left: initial trajectories assigned for two agents. Top right, final trajectories obtained solving Problem P4.1. Bottom: cost vs. number of iterations. \( J^* = 1660 \).  

4.4 Trajectories by solving the TPBVP problem. Top left: initial trajectories assigned for two agents. Top right, final trajectories obtained solving the TPBVP problem. Bottom: cost vs. number of iterations. \( J^* = 1575 \).
5·1 Comparison of Receding Horizon controller with optimal control for two agent 1-dimensional persistent monitoring problem. ............................................ 110


5·4 RHC trajectories using Alg 6. $J^* = 654$. ........................................ 112
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>COV</td>
<td>Calculus of Variations</td>
</tr>
<tr>
<td>CSC</td>
<td>Continuous Stochastic Comparison Algorithm</td>
</tr>
<tr>
<td>DES</td>
<td>Discrete Event Systems</td>
</tr>
<tr>
<td>HJB</td>
<td>Hamilton Jacobi Bellman Equation</td>
</tr>
<tr>
<td>IPA</td>
<td>Infinitesimal Perturbation Analysis</td>
</tr>
<tr>
<td>MRCP</td>
<td>Maximum Reward Collection Problems</td>
</tr>
<tr>
<td>PDE</td>
<td>Partial Differential Equation</td>
</tr>
<tr>
<td>PMP</td>
<td>Pontryagin’s Maximum (Minimum) Principle</td>
</tr>
<tr>
<td>RHC</td>
<td>Receding Horizon Control</td>
</tr>
<tr>
<td>SFM</td>
<td>Stochastic Flow Model</td>
</tr>
<tr>
<td>TPBVP</td>
<td>Two Point Boundary Value Problem</td>
</tr>
<tr>
<td>TSP</td>
<td>Traveling Salesman Problem</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Persistent Monitoring Problem Basics

Enabled by recent technological advances, the deployment of autonomous agents that can cooperatively perform complex tasks is rapidly becoming a reality. In particular, there has been considerable progress reported in the literature on robotics and sensor networks regarding cooperative control (Chandler et al., 2001; Clough, 2002; Zhang and Leonard, 2010) coverage control (Rekleitis et al., 2004; Cortes et al., 2004; Li and Cassandras, 2006), surveillance (Girard et al., 2005; Grocholsky et al., 2006; Chen et al., 2012), path planning (Ulusoy et al., 2012) and environmental sampling (Smith et al., 2012; Paley et al., 2008) missions. Coverage control is the process of controlling the movement of multi agents and ultimately assigning them to target points so as to maximize the total reward collected by visiting points in the mission space within a given operation time. The environment is allow to be changing, such as new target points may show up during the control process, but normally the environment isn’t modeled to change rapidly and continuously. As long as the system is stable and no new target shows up, autonomous agents will converge to the static positions which are the optimal coverage positions for the system. However, as new target or new obstacles comes into the environment, original converged positions may not be optimal and agents will move to the new static optimal positions again.

On the contrary, environment in persistent monitoring is defined to change constantly and sometimes continuously. Compared to the environment needed to be covered, the sensing area, or so called footprint, is small and it’s impossible for agents to cover the whole environment at any static positions. In addition, in order to avoid the uninteresting case where there is a large number of agents who can adequately cover the mission space,
we assume that for any agents formation, there always exist some point that cannot be covered by any agent. Thus, autonomous agents must move and return to every point in the environment periodically to keep their coverage or observation current. Persistent monitoring is defined as the problem of designing the optimal trajectories and finding the optimal movement along those trajectories. In our research, we are interested in generating 1) optimal control, and 2) receding horizon control strategies for persistent monitoring tasks; these arise when agents must monitor a dynamically changing environment which cannot be fully covered by a stationary team of available agents. Persistent monitoring differs from traditional coverage tasks due to the perpetual need to cover a changing environment, i.e., all areas of the mission space must be visited infinitely often. The main challenge in designing control strategies in this case is in balancing the presence of agents in the changing environment so that it is covered over time optimally (in some well-defined sense) while still satisfying sensing and motion constraints. Examples of persistent monitoring missions include surveillance, patrol missions with unmanned vehicles, and environmental applications where routine sampling of an area is involved.

Persistent monitoring problem is also similar to Data Harvesting Problems, or Maximum Reward Collection Problems (MRCP) where agents are traveling in the mission space to collect time-dependent rewards associated with certain number of targets in an uncertain environment. There usually exists a special target called the sink or the base station, where agents have to reach periodically to unload the data or reward (Pennesi and Paschalidis, 2010; Moazzez-Estanjini et al., 2012; Moazzez-Estanjini et al., 2013; Khazaeni and Cassandra, 2014). In a deterministic setting with equal target rewards, a one-agent MRCP is an instance of a Traveling Salesman Problem (TSP) (Salz, 1966; Applegate et al., 2011). MRCP is a combinatorial problems for which globally optimal solutions are found through integer programming algorithms.

We address the 1-dimensional and 2-dimensional persistent monitoring problem by proposing an optimal control framework to drive agents so as to minimize a metric of uncertainty over the environment. In coverage control (Cortes et al., 2004; Li and Cas-
sandras, 2006), it is common to model knowledge of the environment as a non-negative density function defined over the mission space, and usually assumed to be fixed over time. However, since persistent monitoring tasks involve dynamically changing environments, it is natural to extend this model to a function of both space and time to capture uncertainty in the environment. We assume that uncertainty at a point grows in time with a certain rate (could be fixed or random process), if it is not covered by any agent sensors. To model sensor coverage, we define a probability of detecting events at each point of the mission space by agent sensors. Thus, the uncertainty of the environment decreases with a rate proportional to the event detection probability, i.e., the higher the sensing effectiveness is, the faster the uncertainty is reduced.

While it is desirable to track the value of uncertainty over all points in the environment, this is generally infeasible due to computational complexity and memory constraints. Motivated by polling models in queueing theory, e.g., spatial queueing (Bertsimas and Van Ryzin, 1993; Cooper, 1981), and by stochastic flow models (Sun et al., 2004; Yu and Cassandras, 2004), we assign sampling points of the environment to be monitored persistently (this is equivalent to partitioning the environment into a discrete set of regions.) We associate to these points uncertainty queues” which are visited by one or more servers”. The growth in uncertainty at a sampling point can then be viewed as a flow into a queue, and the reduction in uncertainty (when covered by an agent) can be viewed as the queue being visited by mobile servers as in a polling system. Moreover, the service flow rates may or may not depend on the distance of the sampling point to nearby agents. The service flow rates can decay linearly or exponentially as the distance between agent and sampling point increases, or the service rate can maintain constant as long as the sampling point is covered by any one of the agent. From this point of view, we aim to control the movement of the servers (agents) so that the total accumulated uncertainty queue” content is minimized.

Control and motion planning for agents performing persistent monitoring tasks have been studied in the literature. In (Rekleitis et al., 2004) the focus is on sweep coverage problems, where agents are controlled to sweep an area. In (Smith et al., 2012; Nigam and
Kroo, 2008) a similar metric of uncertainty is used to model knowledge of a dynamic environment. In (Nigam and Kroo, 2008), the sampling points in a 1-dimensional environment are denoted as cells, and the optimal control policy for a two-cell problem is given. Problems with more than two cells are addressed by a heuristic policy. In (Smith et al., 2012), the authors proposed a stabilizing speed controller for a single agent so that the accumulated uncertainty over a given path in the environment is bounded, along with an optimal controller that minimizes the maximum steady-state uncertainty, assuming that the agent travels along a closed path and does not change direction. In (Soltero et al., 2012), a curve shaping algorithm is combined with a speed controller to produce guaranteed persistent monitoring trajectories in unknown dynamics environment. In (Lan and Schwager, 2013), a new randomized path planning algorithm is proposed to find a periodic trajectory for sensing robot to best estimate a time-changing Gaussian Random Field in its environment.

The persistent monitoring problem is also related to robot patrol problems, where a team of robots are required to visit points in the workspace with frequency constraints (Agmon et al., 2008; Hokayem et al., 2008; Elmaliach et al., 2008; Elmaliach et al., 2007; Agmon et al., 2012).

Our ultimate goal is to optimally control a team of cooperating agents in a two or three-dimensional environment. The contribution of our research is to take a first step toward this goal by formulating and solving an optimal control problem for a team of agents moving in an 1-dimensional mission space described by an interval $[0, L] \subset \mathbb{R}$ in which we minimize the accumulated uncertainty over a given time horizon and over an arbitrary number of sampling points. Even in this simple case, determining a complete explicit solution is computationally hard, as seen in (Cassandras et al., 2011) where the single-agent case was first considered. However, we show that the problem can be reduced to a parametric optimization problem. In particular, the optimal trajectory of each agent is to move at full speed until it reaches some switching point, dwell on the switching point for some time (possibly zero), and then switch directions. In addition, we prove that all agents should never reach the end points of the mission space $[0, L]$. Thus, each agent’s optimal
trajectory is fully described by a set of switching points \( \{\theta_1, \ldots, \theta_K\} \) and associated waiting times at these points, \( \{w_1, \ldots, w_K\} \). As a result, we show that the behavior of the agents operating under optimal control is described by a hybrid system. This allows us to make use of generalized Infinitesimal Perturbation Analysis (IPA), as presented in (Cassandras et al., 2010; Wardi et al., 2010), to determine gradients of the objective function with respect to these parameters and subsequently obtain optimal switching locations and waiting times that fully characterize an optimal solution. It also allows us to exploit robustness properties of IPA to extend this solution approach to a stochastic uncertainty model.

Our analysis establishes the basis for extending this approach to a 2-dimensional mission space, where the objective is still to control the trajectories of multiple cooperating agents to minimize an uncertainty metric in 2-dimensional mission space. Using a similar analysis to the 1-dimensional case, we find that we can no longer identify a parametric representation of optimal agent trajectories. A complete solution requires a computationally intensive process for solving a Two Point Boundary Value Problem (TPBVP) making any on-line solution to the problem infeasible. Motivated by the simple structure of the 1-dimensional problem, it has been suggested to assign each agent a linear trajectory for which the explicit 1-dimensional solution can be used. One could then reduce the problem to optimally carrying out this assignment. However, in a 2-dimensional space, it is not obvious that a linear trajectory is a desirable choice. Indeed, we formally prove that an elliptical agent trajectory outperforms a linear one in terms of the uncertainty metric we are using. Motivated by this result, we formulate a 2-dimensional persistent monitoring problem as one of determining optimal elliptical trajectories for a given number of agents, noting that this includes the possibility that two or more agents share the same trajectory. We show that this problem can be solved using similar IPA techniques as in our 1-dimensional analysis. In particular, we use IPA to determine on line the gradient of the objective function with respect to the parameters that fully define each elliptical trajectory (center, orientation and length of the minor and major axes). This approach is scalable in the number of observed events, not states, of the underlying hybrid system characterizing the persistent monitoring.
process, so that it is suitable for online implementation. However, the standard gradient-based optimization process we use is generally limited to local, rather than global optimal solutions. Thus, we adopt a stochastic comparison algorithm from the literature (Bao and Cassandras, 1996) to overcome this problem.

We further approach the problem by representing an agent trajectory in terms of general function families characterized by a set of parameters that we can optimize. We seek to optimize the set of parameters given the same persistent monitoring objective function. In particular, we study two families of functions: Lissajous functions (Cundy and Rollett, 1989) and a Fourier series representation of a trajectory. Motivated by the simple oscillatory optimal trajectory structure in the 1-dimensional problem, we consider Lissajous functions because of their property to systematically describe complex harmonic motion in a 2-dimensional space. Trajectories based on a Fourier series representation, on the other hand, are used to approximate any arbitrary trajectory and are more suitable when the mission space is irregular (i.e., its shape is complex or the weights and distribution of sampling points in the mission space are inhomogeneous). We derive suitable parameterizations for both trajectory representations and show that the problem of determining optimal parameters can be explicitly solved using similar IPA techniques as in our 1-dimensional analysis and the 2-dimensional analysis limited to elliptical trajectories. This is done, again through IPA gradients of the objective function with respect to these parameters evaluated online so as to adjust them towards optimality.

In a broader context, our approach brings together optimal control, hybrid systems, and perturbation analysis techniques in solving a class of problems which, under optimal control, can be shown to behave like hybrid systems characterized by a set of parameters whose optimal values deliver a complete optimal control solution.

1.2 Optimal Control Approaches

Optimal control theory is a mathematical optimization method for deriving control policies. A control problem includes a cost functional that is a function of state and control variables.
Optimal control deals with the problem of finding a control law for a given system such that a certainty optimality criterion for the cost function is achieved (Pontryagin, 1962). It is a set of differential equations describing the paths of the control variables that minimize the cost functional. If we formulate these problems in discrete form by dividing time (or distance) into a finite number of intervals, then they may be solved by the conventional techniques such as Lagrange's method and nonlinear programming. However, when it comes to the optimization over continuous time, more technical difficulties are incurred. In the continuous-time model, the number of decision variables is no longer finite: since decisions may be taken at each instant of time, there is a continuously infinite number of decision variables. The rigorous treatment of optimization in an infinite-dimensional space requires the use of advanced mathematics techniques other than the optimization methods mentioned above.

The optimal control can be derived using Pontryagin's maximum principle (PMP) (a necessary condition also known as Pontryagin's minimum principle or simply Pontryagin's Principle (Ross, 2009)), or by solving the Hamilton Jacobi Bellman equation (HJB) (a sufficient condition). Calculus of variations (COV) is the oldest approach that deals with the interior solution. But in most applications, as it turned out, decision variables are often bounded. PMP, which was developed to deal with such cases, states that the Hamiltonian must be minimized over all permissible controls. The result is first successfully applied into the minimum time problem when the input control is constrained. For the fixed final time problem where the Hamiltonian doesn't explicitly depend on time as our optimal control framework for persistent monitoring problem, the Hamiltonian maintains a constant value. When satisfied along a trajectory, Pontryagin's minimum principle is a necessary condition for an optimum. Hamilton Jacobi Bellman equation is a result of the theory of dynamic programming which was pioneered in the 1950s by Richard Bellman and coworkers (Bellman, 1957). HJB equation, which can be generated to stochastic systems as well, when solved over the whole of state space, is a necessary and sufficient condition for an optimum.

Dynamic programming exploits the recursive nature of the optimal control problem.
Many problems including those treated by the PMP and HJB have the property that the optimal policy from any arbitrary time on depends only on the state of the system at that time and does not depend on the paths that the decision variables have taken up to that time. In such cases, the optimal value of the objective function beyond time $t$ can be considered as a function of the state of the system at time $t$. This function is called the value function. The value function yields the value which the best possible performance achieves from $t$ to the end of the interval. The dynamic programming approach solves the optimization problem by first defining the value function.

In persistent monitoring problem, we are interested in controlling single or multi-agent in one or 2-dimensional environment to optimize a well defined monitoring performance, so optimal control method becomes our major tool to conduct analysis. First, we aim to find the properties of the optimal solution for 1-dimensional multi-agent persistent monitoring problem. Accordingly, the problem is determined in a deterministic environment setting and then we find that the optimal control structure for the deterministic environment can be well extended to an stochastic environment, where the increasing rate of the uncertainty value for sampling points follows a stationary random process. As for deterministic optimal control, we use the PMP, which provides a necessary condition for the optimal control, to show the optimal variables can only take discrete values: either move with full speed or stop for every agent. Therefore, we turn the optimal control problem into a parameterized optimization problem.

Next we extend to the 2-dimensional problem and use PMP to prove that the optimal control solution is for all agents to move at full speed. We are left with the task of determining the optimal heading. This can be accomplished by solving the standard TPBVP (Press et al., 2007) involving forward and backward integrations of the state and costate equations to evaluate the partial derivative of the Hamiltonian with respect to control variablse after each such iteration and using a gradient descent approach until the objective function converges to a (local) minimum. Shooting method is a classical numerical method used in solving TPBVP. We choose values for all of the dependent variables at one boundary and
these values must be consistent with any boundary conditions for that boundary. Clearly this is a computationally intensive process which scales poorly with the number of agents and the size of the mission space. Another disadvantages of solving TPBVP problem is that it can only give us numerical results.

1.3 Infinitesimal Perturbation Analysis

The study of hybrid systems is based on a combination of modeling frameworks originating in both time-driven and event-driven dynamics systems and resulting in hybrid automata. In a hybrid automaton (Cassandras and Lafortune, 2009), discrete events cause transitions from one discrete state (or mode) to another. While operating at a particular mode, the system's behavior is described by differential equations. In a stochastic setting, such frameworks are augmented with models for random processes that affect either the time-driven dynamics or the events causing discrete state transitions or both. The optimization of hybrid systems is generally hard because of the absence of closed-form expressions capturing the dependence of interesting performance metrics on controllable parameters (Cassandras et al., 2010). In the early 1980s, it is discovered that we can efficiently extract sensitivities of various performance metrics with respect to certain types of control parameters from state trajectories during each particular mode or between modes transitions. This has led to the development of a theory for infinitesimal perturbation analysis (IPA). Using IPA, we can obtain unbiased gradient estimates of performance metrics that can be incorporated into standard gradient-based algorithms for optimization purposes (Ho and Cao, 1991; Glasserman, 1991). However, IPA estimates become biased (hence unreliable for control purposes) when dealing with various aspects of DES that cause significant discontinuities in sample functions of interest. Such discontinuities normally arise when a parameter perturbation causes the order in which events occur to be affected and this event order change may violate a basic commuting condition (Glasserman, 1991; Holtzman, 1992). When this happens, one must resort to significantly more complicated methods for deriving unbiased estimates.

IPA produces gradient information based on one experiment performed. It uses the gra-
dient information as the computational overhead for the next execution of the controllable variables. (Dai and Ho, 1995) extends Perturbation analysis, including IPA, to derivative estimation in discrete event system on structural parameters. In (Wardi et al., 2010), an abstract framework is developed for IPA in the setting of stochastic flow models and shows its application in flow controls in single server fluid flow models. In (Hu, 1992), a strong consistency of IPA estimates which is based on convex analysis of sample paths, is proposed to solve tandem and cyclic queueing systems. In (Geng and Cassandras, 2012), a traffic light control problem is modeled as a Stochastic Flow Model (DFM) and the IPA algorithm is used to derive online gradient estimates of a cost metric with respect to the controllable green and red light cycles.

We show that the optimal control problem can be reduced to a parametric optimization problem. In particular, the optimal trajectory of each agent is to move at full speed until it reaches some switching point, dwell on the switching point for some time (possibly zero), and then switch directions. In 1-dimensional mission space, we show that agent’s optimal trajectory is fully described by a set of switching points \( \{\theta_1, \ldots, \theta_K\} \) and associated waiting times at these points, \( \{w_1, \ldots, w_K\} \). In the 2-dimensional mission space, we approximate the optimal trajectories by parameterizing elliptical, Lissajous and Fourier Series functions. Every agent’s optimal trajectories are again fully characterized by the set of parameters. As a result, we show that the behavior of the agents operating under optimal control is described by a hybrid system, which is defined as a system that is a combination of both time-driven and event-driven dynamic systems. For the one and 2-dimensional persistent monitoring optimal control problem, after showing that it can be reduced to a parameterized optimization problem, we apply IPA to obtain the unbiased gradient of the cost function with respect to the controllable parameters iteratively until it converges to the (local) minimum.

1.4 Receding Horizon Control

Receding horizon control is based on iterative, finite horizon optimization of a plant model (Zheng and Morari, 1995; Zheng and Morari, 1995; Mao and Chai, 1996). (Mayne and
Michalska, 1990) shows that when finite horizon function is continuously differentiable, the receding horizon control stabilizes a wide class of non-linear systems. At current time the system state is sampled and a cost minimizing control strategy is computed during the planning horizon. Specifically, an online or on-the-fly calculation is used to explore state trajectories that emanate from the current state and find a cost-minimizing control strategy until planning horizon time, which is typically smaller than the planning horizon time. Then the control strategy computed in the last step is implemented for a time duration called action horizon. After that the system state is sampled again. The prediction horizon keeps being shifted forward, and the calculations are repeated starting from the current state, yielding a new control and new predicted state path.

Receding horizon control is in essence a time decomposition approach. It is associated with model-predictive control, which is used to solve optimal control problems for which feedback control are hard or impossible to obtain. (Singh and Fuller, 2001) describes a receding-horizon optimal control scheme for autonomous trajectory generation and flight control of an unmanned air vehicle in urban terrain. In (Cruz et al., 2001), a receding horizon controller that can take into account the possible near-term control actions of the adversary is proposed and the RH controller can achieve the sub-optimal in the global sense. (Franco et al., 2004; Dunbar and Murray, 2004; Richards and How, 2004; Frazzoli and Bullo, 2004) study the decentralized receding horizon control in multi-agent cooperative control. (?) proposes a receding horizon controller that can generate stationary trajectories that is suitable for dynamic and uncertain environments.

We propose a receding horizon controller suitable for dynamic and uncertain environments, where off-line optimal control approach may be infeasible. In addition, for persistent monitoring problems where optimal control approach is too time consuming, receding horizon controller can provide an efficient and near optimal result. The control scheme dynamically determines agent control by solving a sequence of optimal control over a planning horizon and executing them over a shorter action horizon. For the 1-dimensional persistent monitoring problems, since we can show that the system dynamics, cost function and ter-
minal constraints are not explicit function of time, we are able to prove that the control strategy is stationary as long as the system states stay on the optimal control trajectory. By designing a prudent terminal constraint for the planning horizon (too short would increase computation burden while too long would contain more than one switching point), the optimal control strategy is stationary and we are safe, without missing the switching point, to let the system evolve for the action horizon time duration, using the optimal control calculated during the planning horizon. Receding horizon control can not only greatly simplify the optimal control problem, but can also make the control strategy adapt to the changing environment. Optimal control is an off-line calculation, while receding horizon control is an on-line one.

1.5 Contributions of The Dissertation

The contributions of the dissertation are to lay out a formal optimal control foundation for the 1 and 2-dimensional persistent monitoring problem. Chapter 2 formulates the optimal control framework for 1-dimensional persistent monitoring problem, proves properties concerning optimal speed for movement and shows that it boils down to a parameterized optimization problem. An efficient event-driven gradient descent algorithm that can achieve the optimal cost is presented. We also discuss the effect of adding upper bound constraints to some sampling points in the mission space and show that our parameterized optimization algorithm using gradient descent method can be well extended to this problem with upper bound constraints. The upper bound constraints can provide guarantee that sampling points in the mission space don’t have uncertainty values higher than the bound. It justifies applicability of the optimal control algorithm when some points in the mission space need significant attention.

Chapter 3 address the same persistent monitoring problem in a 2-dimensional mission space. Using an analysis similar to the 1-dimensional case, we find that we can no longer identify a parametric representation of optimal agent trajectories. The key contribution is that we formally prove that an elliptical agent trajectory outperforms a linear one in terms of
the uncertainty metric we are using. Motivated by this result, we formulate a 2-dimensional persistent monitoring problem as one of determining optimal elliptical trajectories for a given number of agents.

In Chapter 4, our contribution is to represent an agent trajectory in terms of general function families characterized by a set of parameters that we seek to optimize, given a persistent monitoring objective function for the 2-dimensional persistent monitoring problem. In particular, we study two families of functions: Lissajous functions and a Fourier series representation of a trajectory. Motivated by the simple oscillatory optimal trajectory structure in the 1-dimensional problem, we consider Lissajous functions because of their property to systematically describe complex harmonic motion in a 2-dimensional space. Trajectories based on a Fourier series representation, on the other hand, are used to approximate any arbitrary trajectory and are more suitable when the mission space is irregular (i.e., its shape is complex or the weights and distribution of sampling points in the mission space are inhomogeneous).

In Chapter 5, two centralized RH controllers are proposed for 1 and 2-dimensional persistent monitoring problem respectively, which are capable of obtaining near optimal cost on the fly.

1.6 Dissertation Organization

The rest of the proposed thesis is organized as follows. Chapter 2 address the 1-dimensional persistent monitoring problem by proposing an optimal control framework to drive agents so as to minimize a metric of uncertainty over the environment. Section 2.1 formulates the optimal control framework for 1-dimensional problem. Section 2.2 characterizes the solution of the problem in terms of two parameter vectors specifying switching points in the mission space and associated dwelling times at them. Section 2.3 and Section 2.4 introduce IPA and use it to obtain the gradient of the cost function with respect to the switching points and dwelling time. In conjunction with a gradient descent based algorithm, a complete parameter optimization solution is provided. Section 2.5 presents some numerical
experiment results for the multi-agent 1-dimensional unconstrained persistent monitoring problem. Section 2.6 formulates the same problem with performance constraints added, where the penalizing time intervals are activated during which they are violated. We include numerical examples illustrating the solution approach and providing some comparisons between unconstrained and constrained cases.

Chapter 3 address the persistent monitoring problem in 2-dimensional mission spaces where the objective is to control the trajectories of multiple cooperating agents to minimize an uncertainty metric through elliptical agent trajectories. Section 3.1 formulates the optimal control framework for the 2-dimensional mission space and Section 3.2 presents the solution approach using Hamiltonian analysis. In Section 3.3, we establish our key result that elliptical agent trajectories outperform linear ones in terms of minimizing an uncertainty metric per unit area. In Section 3.4, we formulate and solve the problem of determining optimal elliptical agent trajectories using an algorithm driven by gradients evaluated through IPA. In Section 3.5, we incorporate a stochastic comparison algorithm for obtaining globally optimal solutions and in Section 3.6 we provide numerical results to illustrate our approach and compare it to computationally intensive solutions based on a TBPV solver. Section 3.7 concludes the chapter.

Chapter 4 approach the same multi-agent 2-dimensional persistent monitoring problem by representing an agent trajectory in terms of general function families characterized by a set of parameters that we can optimize. In particular, we have applied this approach to the family of Lissajous functions as well as a Fourier series representation of an agent trajectory. We skip the problem formulation and the Hamiltonian analysis as they are shown in Section 3.1 and Section 3.2. Section 4.1 formulates and solve the problem of determining optimal trajectories based on general function representations using a gradient-based algorithm using IPA. In Section 4.2, we concentrate on two particular function families applying the general analysis. Section 4.3 provides numerical results and Section 4.4 concludes the chapter.

Chapter 5 propose centralized RHC algorithm for 1 and 2-dimensional multi-agent per-
sistent monitoring problem, which can achieve near optimal results, compared to the gradient descent algorithm using IPA for 1-dimensional case and TPBVP for 2-dimensional case. In Section 5.1, we prove the stationarity of the optimal switching point for a well defined free final time problem and then we show that the optimal final time equals the final time constraint. These two propositions motivate us to use a centralized RHC to solve multi-agent 1-dimensional persistent monitoring problem. In Section 5.2, a centralized RHC, where the planning horizon is defined as the smallest distance for all agents to the boundary of the mission space is presented to solve 2-dimensional persistent monitoring problem. Section 5.3 shows numerical experiment results for both cases respectively, where 1-dimensional RHC is compared to the gradient descent IPA algorithm, while 2-dimensional RHC results are compared to TPBVP results. We show that they both achieve near optimal cost.
Chapter 2
1-dimensional Persistent Monitoring Problem

Autonomous cooperating agents may be used to perform tasks such as coverage control, surveillance, and environmental sampling. Persistent monitoring arises in a large dynamically changing environment which cannot be fully covered by a stationary team of available agents. Thus, persistent monitoring differs from traditional coverage tasks due to the perpetual need to cover a changing environment, i.e., all areas of the mission space must be visited infinitely often. The main challenge in designing control strategies in this case is in balancing the presence of agents in the changing environment so that it is covered over time optimally (in some well-defined sense) while still satisfying sensing and motion constraints.

In this chapter, we addressed the persistent monitoring problem by proposing an optimal control framework in the 1-dimensional mission space to drive multiple agents so as to minimize a metric of uncertainty over the environment. This metric is a function of both space and time defined so that uncertainty at a point grows if it is not covered by any agent sensors. To model sensor coverage, we define a probability of detecting events at each point of the mission space by agent sensors. Thus, the uncertainty of the environment decreases with a rate proportional to the event detection probability, i.e., the higher the sensing effectiveness is, the faster the uncertainty is reduced. Our ultimate goal is to optimally control a team of cooperating agents in a two or three-dimensional environment.

We show that the optimal control problem can be reduced to a parametric optimization problem. In particular, the optimal trajectory of each agent is to move at full speed until it reaches some switching point, dwell on the switching point for some time (possibly zero), and then switch directions. In addition, we prove that all agents should never reach the end points of the mission space. Thus, each agents optimal trajectory is fully described
by a set of switching points \( \{\theta_1, \ldots, \theta_K\} \) and associated waiting times at these points, \( \{w_1, \ldots, w_K\} \). As a result, we show that the behavior of the agents operating under optimal control is described by a hybrid system. This allows us to make use of IPA to determine gradients of the objective function with respect to these parameters and subsequently obtain optimal switching locations and waiting times that fully characterize an optimal solution.

When adding uncertainty thresholds, we show that only the event time instants at which the uncertainty values exceed or go below these upper bound constraints influence the evaluation of the gradient of the objective function with respect to the switching locations and waiting times. The IPA approach also allows us to exploit its inherent robustness properties and readily extend this solution approach to a stochastic uncertainty model.

The remainder of the chapter is organized as follows. Section 2.1 formulates the optimal control framework for 1-dimensional problem. Section 2.2 characterizes the solution of the problem in terms of two parameter vectors specifying switching points in the mission space and associated dwelling times at them. Section 2.3 and Section 2.4 introduce IPA and use it to obtain the gradient of the cost function with respect to the switching points and dwelling time. In conjunction with a gradient descent based algorithm, a complete parameter optimization solution is provided. Section 2.5 presents some numerical experiment results for the multi-agent 1-dimensional unconstrained persistent monitoring problem. Section 2.6 formulates the same problem with performance constraints added, where the penalizing time intervals are activated during which they are violated. We include numerical examples illustrating the solution approach and providing some comparisons between unconstrained and constrained cases.

2.1 1D Persistent Monitoring Problem Formulation

We consider \( N \) mobile agents moving in a 1-dimensional mission space of length \( L \), for simplicity taken to be an interval \([0, L] \subset \mathbb{R}\). Let the position of the agents at time \( t \) be \( s_n(t) \in [0, L], n = 1, \ldots, N \), following the dynamics:

\[
\dot{s}_n(t) = u_n(t)
\]
i.e., we assume that the agent can control its direction and speed. Without loss of generality, after some rescaling with the size of the mission space $L$, we further assume that the speed is constrained by $|u_n(t)| \leq 1$, $n = 1, \ldots, N$. For the sake of generality, we include the additional constraint:

$$a \leq s(t) \leq b, \; a \geq 0, \; b \leq L$$  \hspace{1cm} (2.2)

over all $t$ to allow for mission spaces where the agents may not reach the end points of $[0, L]$, possibly due to the presence of obstacles. We also point out that the agent dynamics in (2.1) can be replaced by a more general model of the form $\dot{s}_n(t) = g_n(s_n) + b_n u_n(t)$ without affecting the main results of our analysis (see also Remark 1 in the next section.) Finally, an additional constraint may be imposed if we assume that the agents are initially located so that $s_n(0) < s_{n+1}(0)$, $n = 1, \ldots, N$, and we wish to prevent them from subsequently crossing each other over all $t$:

$$s_n(t) - s_{n+1}(t) \leq 0$$  \hspace{1cm} (2.3)

We associate with every point $x \in [0, L]$ a function $p_n(x, s_n)$ that measures the probability that an event at location $x$ is detected by agent $n$. We also assume that $p_n(x, s_n) = 1$ if $x = s_n$, and that $p_n(x, s_n)$ is monotonically nonincreasing in the distance $|x - s_n|$ between $x$ and $s_n$, thus capturing the reduced effectiveness of a sensor over its range which we consider to be finite and denoted by $r_n$ (this is the same as the concept of sensor footprint" found in the robotics literature.) Therefore, we set $p_n(x, s_n) = 0$ when $|x - s_n| > r_n$. Although our analysis is not affected by the precise sensing model $p_n(x, s_n)$, we will limit ourselves to a linear decay model as follows:

$$p_n(x, s_n) = \begin{cases} \frac{1}{r_n} \frac{|x - s_n|}{r_n}, & \text{if } |x - s_n| \leq r_n \\ 0, & \text{if } |x - s_n| > r_n \end{cases}$$  \hspace{1cm} (2.4)

Next, consider a set of points $\{\alpha_i\}, \; i = 1, \ldots, M$, $\alpha_i \in [0, L]$, and associate a time-varying measure of uncertainty with each point $\alpha_i$, which we denote by $R_i(t)$. Without loss of generality, we assume $0 \leq \alpha_1 \leq \cdots \leq \alpha_M \leq L$ and, to simplify notation, we set $p_{n,i}(s_n(t)) \equiv p_n(\alpha_i, s_n(t))$. This set may be selected to contain points of interest in the environment, or
sampled points from the mission space. Alternatively, we may consider a partition of $[0, L]$ into $M$ intervals whose center points are $\alpha_i = \frac{(2i-1)L}{2M}$, $i = 1, \ldots, M$. We can then set $p_n(x, s_n(t)) = p_{n,i}(s_n(t))$ for all $x \in [\alpha_i - L/2M, \alpha_i + L/2M]$. Therefore, the joint probability of detecting an event at location $x \in [\alpha_i - L/2M, \alpha_i + L/2M]$ by all the $N$ agents simultaneously (assuming detection independence) is:

$$P_i(s(t)) = 1 \prod_{n=1}^{N} [1 - p_{n,i}(s_n(t))]$$

where we set $s(t) = [s_1(t), \ldots, s_N(t)]^T$. We define uncertainty functions $R_i(t)$ associated with the intervals $[\alpha_i - L/2M, \alpha_i + L/2M]$, $i = 1, \ldots, M$, so that they have the following properties:

(i) $R_i(t)$ increases with a prespecified rate $A_i$ if $P_i(s(t)) = 0$,
(ii) $R_i(t)$ decreases with a fixed rate $B$ if $P_i(s(t)) = 1$ and
(iii) $R_i(t) \geq 0$ for all $t$. It is then natural to model uncertainty so that its decrease is proportional to the probability of detection. In particular, we model the dynamics of $R_i(t)$, $i = 1, \ldots, M$, as follows:

$$\dot{R}_i(t) = \begin{cases} 0 & \text{if } R_i(t) = 0, \ A_i \leq BP_i(s(t)) \\ A_i & \text{BP}_i(s(t)) \end{cases}$$

where we assume that initial conditions $R_i(0)$, $i = 1, \ldots, M$, are given and that $B > A_i > 0$ (thus, the uncertainty strictly decreases when there is perfect sensing $P_i(s(t)) = 1$.)

Viewing persistent monitoring as a polling system, each point $\alpha_i$ (equivalently, $i$th interval in $[0, L]$) is associated with a virtual queue where uncertainty accumulates with inflow rate $A_i$. The service rate of this queue is time-varying and given by $BP_i(s(t))$, controllable through the agent position at time $t$. Figure 2.1 illustrates this polling system when $N = 1$. This interpretation is convenient for characterizing the stability of such a system over a mission time $T$: For each queue, we may require that $\int_0^T A_i < \int_0^T BP_i(s(t))dt$. Alternatively, we may require that each queue becomes empty at least once over $[0, T]$. We may also impose conditions such as $R_i(T) \leq R_{\max}$ for each queue as additional constraints for our problem so as to provide bounded uncertainty guarantees. Note that this analogy readily extends to two or three-dimensional settings.

The goal of the optimal persistent monitoring problem we consider is to control the
Figure 2-1: A queueing system analog of the persistent monitoring problem.

movement of the $N$ agents through $u_n(t)$ in (2.1) so that the cumulative uncertainty over all sensing points $\{\alpha_i\}, i = 1, \ldots, M$ is minimized over a fixed time horizon $T$. Thus, setting $u(t) = [u_1(t), \ldots, u_N(t)]$ we aim to solve the following optimal control problem $P2.1$:

$$
\min_{u(t)} J = \frac{1}{T} \int_0^T \sum_{i=1}^M R_i(t) dt
$$

subject to the agent dynamics (2.1), uncertainty dynamics (2.6), control constraint $|u_n(t)| \leq 1$, $t \in [0, T]$, and state constraints (2.2), $t \in [0, T]$. Note that we require $a \leq r_n$ and $b \geq L - r_m$, for at least some $n, m = 1, \ldots, N$; this is to ensure that there are no points in $[0, L]$ which can never be sensed, i.e., any $i$ such that $\alpha_i < a$ or $\alpha_i > b + r_n$ would always lie outside any agent’s sensing range. We will omit the additional constraint (2.3) from our initial analysis, but we will show that, when it is included, the optimal solution never allows it to be active.

2.2 Optimal Control Solution

We first characterize the optimal control solution of problem $P2.1$ and show that it can be reduced to a parametric optimization problem. This allows us to utilize an Infinitesimal Perturbation Analysis (IPA) gradient estimation approach (Cassandras et al., 2010) to find a complete optimal solution through a gradient-based algorithm. We define the state vector
\[ x(t) = [s_1(t), \ldots, s_N(t), R_1(t), \ldots, R_M(t)]^T \] and the associated costate vector \( \lambda(t) = [\lambda_{s_1}(t), \ldots, \lambda_{s_N}(t), \lambda_1(t), \ldots, \lambda_M(t)]^T \). In view of the discontinuity in the dynamics of \( R_i(t) \) in (2.6), the optimal state trajectory may contain a boundary arc when \( R_i(t) = 0 \) for some \( i \); otherwise, the state evolves in an interior arc. We first analyze the system operating in such an interior arc and omit the constraint (2.2) as well. Using (2.1) and (2.6), the Hamiltonian is

\[
H(x, \lambda, u) = \sum_{i=1}^{M} R_i(t) + \sum_{n=1}^{N} \lambda_{s_n}(t) u_n(t) + \sum_{i=1}^{M} \lambda_i(t) \dot{R}_i(t) \tag{2.8}
\]

and the costate equations \( \dot{\lambda} = \frac{\partial H}{\partial x} \) are

\[
\dot{\lambda}_i(t) = \frac{\partial H}{\partial R_i(t)} = 1, \quad i = 1, \ldots, M \tag{2.9}
\]

\[
\dot{\lambda}_{s_n}(t) = \frac{\partial H}{\partial s_n(t)} = \frac{B}{r_n} \sum_{i \in F_n(t)} \lambda_i(t) \prod_{d \neq n} \left[1 - p_{d,i}(s_d(t))\right] + \frac{B}{r_n} \sum_{i \in F_n^+(t)} \lambda_i(t) \prod_{d \neq n} \left[1 - p_{d,i}(s_d(t))\right] \tag{2.10}
\]

where we have used (2.4), and the sets \( F_n(t) \) and \( F_n^+(t) \) are defined as

\[
F_n(t) = \{i : s_n(t) - r_n \leq \alpha_i \leq s_n(t)\}, \quad F_n^+(t) = \{i : s_n(t) < \alpha_i \leq s_n(t) + r_n\} \tag{2.11}
\]

for \( n = 1, \ldots, N \). Note that \( F_n(t) \), \( F_n^+(t) \) identify all points \( \alpha_i \) to the left and right of \( s_n(t) \) respectively that are within agent \( n \)'s sensing range. Since we impose no terminal state constraints, the boundary conditions are \( \lambda_i(T) = 0, \quad i = 1, \ldots, M \) and \( \lambda_{s_n}(T) = 0, \quad n = 1, \ldots, N \). Applying the Pontryagin minimum principle to (2.8) with \( u^*(t), \quad t \in [0,T] \), denoting an optimal control, we have

\[
H(x^*, \lambda^*, u^*) = \min_{u_n \in [1,1], n=1, \ldots, N} H(x, \lambda, u)
\]
and it is immediately obvious that it is necessary for an optimal control to satisfy:

\[
    u^*_n(t) = \begin{cases} 
        1 & \text{if } \lambda_{s_n}(t) < 0 \\
        1 & \text{if } \lambda_{s_n}(t) > 0 
    \end{cases}
\]  

(2.12)

This condition excludes the possibility that \( \lambda_{s_n}(t) = 0 \) over some finite singular intervals (Bryson and Ho, 1975). We will show that if \( s_n(t) = a > 0 \) or \( s_n(t) = b < L \), then \( \lambda_{s_n}(t) = 0 \) for some \( n \in \{1, \ldots, N\} \) may in fact be possible for some finite arc; otherwise \( \lambda_{s_n}(t) = 0 \) can arise only when \( u_n(t) = 0 \).

The implication of (2.9) with \( \lambda_i(T) = 0 \) is that \( \lambda_i(t) = T \) for all \( t \in [0,T] \) and all \( i = 1, \ldots, M \) and that \( \lambda_i(t) \) is monotonically decreasing starting with \( \lambda_i(0) = T \). However, this is only true if the entire optimal trajectory is an interior arc, i.e., all \( R_i(t) \geq 0 \) constraints for all \( i = 1, \ldots, M \) remain inactive. On the other hand, looking at (2.10), observe that when the two end points, 0 and \( L \), are not within the range of an agent, we have \( |F_n(t)| = |F_n^+(t)| \), since the number of indices \( i \) satisfying \( s_n(t) < \alpha_i \leq s_n(t) + r_n \) is the same as that satisfying \( s_n(t) \leq \alpha_i \leq s_n(t) + r_n \). Consequently, for the one-agent case \( N = 1 \), (2.10) becomes

\[
    \dot{\lambda}_{s_1}(t) = \frac{B}{r_1} \sum_{i \in F_1(t)} \lambda_i(t) + \frac{B}{r_1} \sum_{i \in F_1^+(t)} \lambda_i(t)
\]  

(2.13)

and \( \dot{\lambda}_{s_1}(t) = 0 \) since the two terms in (2.13) will cancel out, i.e., \( \lambda_{s_1}(t) \) remains constant as long as this condition is satisfied and, in addition, none of the state constraints \( R_i(t) \geq 0 \), \( i = 1, \ldots, M \), is active. Thus, for the one agent case, as long as the optimal trajectory is an interior arc and \( \lambda_{s_1}(t) < 0 \), the agent moves at maximal speed \( u^*_1(t) = 1 \) in the positive direction towards the point \( s_1 = b \). If \( \lambda_{s_1}(t) \) switches sign before any of the state constraints \( R_i(t) \geq 0 \), \( i = 1, \ldots, M \), becomes active or the agent reaches the end point \( s_1 = b \), then \( u^*_1(t) = 1 \) and the agent reverses its direction or, possibly, comes to rest.

In what follows, we examine the effect of the state constraints which significantly complicates the analysis, leading to a challenging two-point-boundary-value problem (TPBVP). However, we will establish the fact that the complete solution boils down to determining
a set of switching locations over \([a, b]\) and waiting times at these switching points, with the end points, 0 and \(L\), being always infeasible on an optimal trajectory. This is a much simpler problem that we are subsequently able to solve.

We begin by recalling that the dynamics in (2.6) indicate a discontinuity arising when the condition \(R_i(t) = 0\) is satisfied while \(\dot{R}_i(t) = A_i BP_i(s(t)) < 0\) for some \(i = 1, \ldots, M\). Thus, \(R_i = 0\) defines an interior boundary condition which is not an explicit function of time. Following standard optimal control analysis (Bryson and Ho, 1975), if this condition is satisfied at time \(t\) for some \(j \in \{1, \ldots, M\}\),

\[
H \ x(t), \lambda(t), u(t) = H \ x(t^+), \lambda(t^+), u(t^+) \]  

(2.14)

where we note that one can choose to set the Hamiltonian to be continuous at the entry point of a boundary arc or at the exit point. Using (2.8) and (2.6), (2.14) implies:

\[
\sum_{n=1}^{N} \lambda^*_n(t^+ u_n^*(t^+)) + \lambda_j^*(t^+) [A_j(t) \ BP_j(s(t))] = \sum_{n=1}^{N} \lambda^*_n(t^+ u_n^*(t^+)) \]  

(2.15)

In addition, \(\lambda^*_n(t^+) = \lambda^*_n(t^+\) for all \(n = 1, \ldots, N\) and \(\lambda_j^*(t^+) = \lambda_j^*(t^+\) for all \(i \neq j\), but \(\lambda_j^*(t^+)\) may experience a discontinuity so that:

\[
\lambda_j^*(t^+) = \lambda_j^*(t^+) \ \pi_j \]  

(2.16)

where \(\pi_j \geq 0\) is a multiplier associated with the constraint \(R_j(t) \leq 0\). Recalling (2.12), since \(\lambda^*_n(t)\) remains unaffected, so does the optimal control, i.e., \(u_n^*(t) = u_n^*(t^+)\). Moreover, since this is an entry point of a boundary arc, it follows from (2.6) that \(A_j BP_j(s(t)) < 0\). Therefore, (2.15) and (2.16) imply that \(\lambda_j^*(t^+) = 0\) and \(\lambda_j^*(t^+) = \pi_j \geq 0\). Thus, \(\lambda_i(t)\) always decreases with constant rate 1 until \(R_i(t) = 0\) is active, at which point \(\lambda_i(t)\) jumps to a non-negative value \(\pi_i\) and decreases with rate 1 again. The value of \(\pi_i\) is determined by how long it takes for the agents to reduce \(R_i(t)\) to 0 once again. Obviously,

\[
\lambda_i(t) \geq 0, \ i = 1, \ldots, M, \ t \in [0, T] \]  

(2.17)
with equality holding only if $t = T$, or $t = t_0$ with $R_i(t_0) = 0$, $R_i(t') > 0$, where $t' \in [t_0, \delta, t_0), \delta > 0$. The actual evaluation of the costate vector over the interval $[0, T]$ requires solving (2.10), which in turn involves the determination of all points where the state variables $R_i(t)$ reach their minimum feasible values $R_i(t) = 0, i = 1, \ldots, M$. This generally involves the solution of a two-point-boundary-value problem. However, our analysis thus far has already established the structure of the optimal control (2.12) which we have seen to remain unaffected by the presence of boundary arcs when $R_i(t) = 0$ for one or more $i = 1, \ldots, M$.

We will next prove some additional structural properties of an optimal trajectory, based on which we show that it is fully characterized by a set of non-negative scalar parameters. Determining the values of these parameters is a much simpler problem that does not require the solution of a two-point-boundary-value problem.

Let us turn our attention to the constraints $s_n(t) \geq a$ and $s_n(t) \leq b$ and consider first the case where $a = 0, b = L$, i.e., the agents can move over the entire $[0, L]$. We shall make use of the following technical condition:

**Assumption 1**: For any $n = 1, \ldots, N, i = 1, \ldots, M, t \in (0, T)$, and any $\epsilon > 0$, if $s_n(t) = 0$, $s_n(t - \epsilon) > 0$, then either $R_i(\tau) > 0$ for all $\tau \in [t - \epsilon, t]$ or $R_i(\tau) = 0$ for all $\tau \in [t - \epsilon, t]$; if $s_n(t) = L, s_n(t - \epsilon) < L$, then either $R_i(\tau) > 0$ for all $\tau \in [t - \epsilon, t]$ or $R_i(\tau) = 0$ for all $\tau \in [t - \epsilon, t]$.

This condition excludes the case where an agent reaches an endpoint of the mission space at the exact same time that any one of the uncertainty functions reaches its minimal value of zero. Then, the following proposition asserts that neither of the constraints $s_n(t) \geq 0$ and $s_n(t) \leq L$ can become active on an optimal trajectory.

**Proposition 2.1.** Under Assumption 1, if $a = 0, b = L$, then on an optimal trajectory:

$s^*_n(t) \neq 0$ and $s^*_n(t) \neq L$ for all $t \in (0, T), n \in \{1, \ldots, N\}$.

**Proof.** Suppose at $t = t_0 < T$ an agent reaches the left endpoint, i.e., $s^*_n(t_0) = 0$, $s^*_n(t_0) > 0$. We will then establish a contradiction. Thus, assuming $s^*_n(t_0) = 0$, we first show that $\lambda^*_n(t_0) = 0$ by a contradiction argument. Assume that $\lambda^*_n(t_0) \neq 0$, in which case, since the agent is moving toward $s_n = 0$, we have $u^*_n(t_0) = 1$ and $\lambda^*_n(t_0) > 0$ from
(2.12). Then, $\lambda^*_n(t)$ may experience a discontinuity so that

$$
\lambda^*_n(t_0) = \lambda^*_n(t_0^+) \pi_n
$$

(2.18)

where $\pi_n \geq 0$ is a scalar constant. It follows that $\lambda^*_n(t_0^+) = \lambda^*_n(t_0) + \pi_n > 0$. Since the constraint $s_n(t) = 0$ is not an explicit function of time, we have

$$
\lambda^*_n(t_0^+) u^*_n(t_0) = \lambda^*_n(t_0^+) u^*_n(t_0^+)
$$

(2.19)

On the other hand, $u^*_n(t_0^+) \geq 0$, since agent $n$ must either come to rest or reverse its motion at $s_n = 0$, hence $\lambda^*_n(t_0^+) u^*_n(t_0) \geq 0$. This violates (2.19), since $\lambda^*_n(t_0) u^*_n(t_0) < 0$. This contradiction implies that $\lambda^*_n(t_0) = 0$. Next, consider (2.10) and observe that in (2.11) we have $F_n(t_0) = \emptyset$, since $\alpha_i > s^*_n(t_0) = 0$ for all $i = 1, \ldots, M$. Therefore, recalling (2.17), it follows from (2.10) that

$$
\dot{\lambda^*_n}(t_0) = \frac{B}{r_n} \sum_{i \in F^+_n(t_0)} \lambda_i(t_0) \prod_{d \neq n} [1 - p_{d,i}(s_d(t_0))] \geq 0
$$

Under Assumption 1, there exists $\delta_1 > 0$ such that during the interval $(t_0 - \delta_1, t_0)$ no $R_i(t) \geq 0$ becomes active, hence no $\lambda_i(t)$ encounters a jump for $i = 1, \ldots, M$. It follows that $\lambda^*_n(t) > 0$ for $i \in F^+_n(t)$ and $\lambda^*_n(t)$ is continuous with $\dot{\lambda^*_n}(t) > 0$ for $t \in (t_0 - \delta_1, t_0)$. Again, since $s^*_n(t_0) = 0$, there exists some $\delta_2 \leq \delta_1$ such that for $t \in (t_0 - \delta_2, t_0)$, we have $u^*_n(t) < 0$ and $\lambda^*_n(t) \geq 0$. Thus, for $t \in (t_0 - \delta_2, t_0)$, we have $\lambda^*_n(t) \geq 0$ and $\dot{\lambda^*_n}(t) > 0$. This contradicts the established fact that $\lambda^*_n(t_0) = 0$ and we conclude that $s^*_n(t) \neq 0$ for all $t \in [0, T]$, $n = 1, \ldots, N$. Using a similar line of argument, we can also show that $s^*_n(t) \neq L$.

\textbf{Proposition 2.2.} If $a > 0$ and (or) $b < L$, then on an optimal trajectory there exist finite length intervals $[t_0, t_1]$ such that $s_n(t) = a$ and (or) $s_n(t) = b$, for some $n \in \{1, \ldots, N\}$, $t \in [t_0, t_1]$, $0 \leq t_0 < t_1 \leq T$.

\textbf{Proof.} Proceeding as in the proof of Proposition 2.1, when $s^*_n(t_0) = a$ we can establish (2.19) and the fact that $\lambda^*_n(t_0) = 0$. On the other hand, $u^*_n(t_0^+) \geq 0$, since the agent must either come to rest or reverse its motion at $s_n(t_0) = a$. In other words, when $s_n(t_0) = a$
on an optimal trajectory, (2.19) is satisfied either with the agent reversing its direction immediately (in which case $t_1 = t_0$ and $\lambda_{s_n}^* (t_0^+) = 0$) or staying on the boundary arc for a finite time interval (in which case $t_1 > t_0$ and $u_{n}^* (t) = 0$ for $t \in [t_0, t_1]$). The exact same argument can be applied to $s_n (t) = b$.

The next result establishes the fact that on an optimal trajectory, every agent either moves at full speed or is at rest.

**Proposition 2.3.** On an optimal trajectory, either $u_{n}^* (t) = \pm 1$ if $\lambda_{s_n}^* (t) \neq 0$, or $u_{n}^* (t) = 0$ if $\lambda_{s_n}^* (t) = 0$ for $t \in [0, T]$, $n = 1, \ldots, N$.

**Proof.** When $\lambda_{s_n}^* (t) \neq 0$, we have shown in (2.12) that $u_{n}^* (t) = \pm 1$, depending on the sign of $\lambda_{s_n}^* (t)$. Thus, it remains to consider the case $\lambda_{s_n}^* (t) = 0$ for some $t \in [t_1, t_2]$, where $0 \leq t_1 < t_2 \leq T$. Since the state is in a singular arc, $\lambda_{s_n}^* (t)$ does not provide information about $u_{n}^* (t)$. On the other hand, the Hamiltonian in (2.8) is not a explicit function of time, therefore, setting $H (x^*, \lambda^*, u^*) = H^*$, we have $\frac{dH^*}{dt} = 0$, which gives

$$\frac{dH^*}{dt} = \sum_{i=1}^{M} \dot{R}_i^* (t) + \sum_{n=1}^{N} \dot{\lambda}_{s_n}^* (t) u_{n}^* (t) + \sum_{n=1}^{N} \lambda_{s_n}^* (t) \ddot{u}_{n}^* (t) + \sum_{i=1}^{M} \dot{\lambda}_i^* (t) \dot{R}_i^* (t) + \sum_{i=1}^{M} \lambda_i^* (t) \ddot{R}_i^* (t) = 0 \quad (2.20)$$

Define $S (t) = \{ n| \lambda_{s_n}^* (t) = 0, n = 1, \ldots, N \}$ as the set of indices of agents that are in a singular arc and $\bar{S} (t) = \{ n| \lambda_{s_n}^* (t) \neq 0, n = 1, \ldots, N \}$ as the set of indices of all other agents. Thus, $\lambda_{s_n}^* (t) = 0$, $\dot{\lambda}_{s_n}^* (t) = 0$ for $t \in [t_1, t_2], n \in S (t)$. In addition, agents move with constant full speed, either 1 or $-1$, so that $\dot{u}_{n}^* (t) = 0$, $n \in \bar{S} (t)$. Then, (2.68) becomes

$$\frac{dH^*}{dt} = \sum_{i=1}^{M} [1 + \dot{\lambda}_i^* (t)] \dot{R}_i^* (t) + \sum_{n \in \bar{S} (t)} \dot{\lambda}_{s_n}^* (t) u_{n}^* (t) + \sum_{i=1}^{M} \lambda_i^* (t) \ddot{R}_i^* (t) = 0 \quad (2.21)$$

From (2.9), $\dot{\lambda}_i^* (t) = -1$, $i = 1, \ldots, M$, so $1 + \dot{\lambda}_i^* (t) = 0$, leaving only the last two terms above. Note that $\dot{\lambda}_{s_n}^* (t) = \frac{\partial H^*}{\partial s_n^* (t)}$ and writing $\ddot{R}_i^* (t) = \frac{d\dot{R}_i^* (t)}{dt}$ we get:

$$\sum_{n \in \bar{S} (t)} u_{n}^* (t) \frac{\partial H^*}{\partial s_n^* (t)} + \sum_{i=1, R_i \neq 0}^{M} \lambda_i^* (t) \frac{d\dot{R}_i^* (t)}{dt} = 0$$
Recall from (2.6) that when $R_i(t) \neq 0$ we have $\hat{R}_i(t) = A_i \prod_{n=1}^{N} \left[ 1 - p_i(s_n(t)) \right]$, so that

$$\frac{\partial H^*}{\partial s_n^*(t)} = B \sum_{i=1, R_i \neq 0}^{M} \lambda_i^*(t) \sum_{n \in S(t)} u_n^*(t) \frac{\partial p_i(s_n^*(t))}{\partial s_n^*(t)} \prod_{d \neq n}^{N} (1 - p_i(s_d^*(t)))$$

$$\frac{d\hat{R}_i^*(t)}{dt} = B \sum_{n=1}^{N} u_n^*(t) \frac{\partial p_i(s_n^*(t))}{\partial s_n^*(t)} \prod_{d \neq n}^{N} (1 - p_i(s_d^*(t)))$$

which results in

$$B \sum_{i=1, R_i \neq 0}^{M} \lambda_i^*(t) \sum_{n \in S(t)} u_n^*(t) \frac{\partial p_i(s_n^*(t))}{\partial s_n^*(t)} \prod_{d \neq n}^{N} (1 - p_i(s_d^*(t))) \tag{2.22}$$

$$B \sum_{i=1, R_i \neq 0}^{M} \lambda_i^*(t) \sum_{n=1}^{N} u_n^*(t) \frac{\partial p_i(s_n^*(t))}{\partial s_n^*(t)} \prod_{d \neq n}^{N} (1 - p_i(s_d^*(t))) = 0 \tag{2.23}$$

Note that $\frac{\partial p_i(s_n^*(t))}{\partial s_n^*(t)} = \pm \frac{1}{r_i}$ or 0, depending on the relative position of $s_i^*(t)$ with respect to $\alpha_i$. Moreover, (2.71) is invariant to $M$ or the precise way in which the mission space $[0, L]$ is partitioned, which implies that

$$\lambda_i^*(t) \sum_{n \in S(t)} u_n^*(t) \frac{\partial p_i(s_n^*(t))}{\partial s_n^*(t)} \prod_{d \neq n}^{N} (1 - p_i(s_d^*(t))) = 0$$

for all $i = 1, \ldots, M, t \in [t_1, t_2]$. Since $\lambda_i^*(t) = 1, i = 1, \ldots, M$, it is clear that to satisfy this equality we must have $u_n^*(t) = 0$ for all $t \in [t_1, t_2], n \in S(t)$. In conclusion, in a singular arc with $\lambda_n^*(t) = 0$ for some $n \in \{1, \ldots, N\}$, the optimal control is $u_n^*(t) = 0$. ■

Next, we consider the case where the additional state constraint (2.3) is included. We can then prove that this constraint is never active on an optimal trajectory, i.e., agents reverse their direction before making contact with any other agent.

**Proposition 2.4.** If the constraint (2.3) is included in problem $P1$, then on an optimal trajectory, $s_n^*(t) \neq s_{n+1}^*(t)$ for $t \in (0, T], n = 1, \ldots, N - 1$.

**Proof.** Suppose at $t = t_0 < T$ we have $s_n^*(t_0) = s_{n+1}^*(t_0)$, for some $n = 1, \ldots, N - 1$. 
We will then establish a contradiction. First assuming that both agents are moving (as opposed to one being at rest) toward each other, we have \( u_{n}^{*}(t_{0}) = 1 \) and \( u_{n+1}^{*}(t_{0}) = -1 \). From (2.12) and Prop 2.3, we know \( \lambda_{s_{n}}^{*}(t_{0}) < 0 \) and \( \lambda_{s_{n+1}}^{*}(t_{0}) > 0 \). When the constraint \( s_{n}(t) s_{n+1}(t) \leq 0 \) is active, \( \lambda_{s_{n}}^{*}(t) \) and \( \lambda_{s_{n+1}}^{*}(t) \) may experience a discontinuity so that

\[
\lambda_{s_{n}}^{*}(t_{0}) = \lambda_{s_{n}}^{*}(t_{0}^{+}) + \pi, \quad \lambda_{s_{n+1}}^{*}(t_{0}) = \lambda_{s_{n+1}}^{*}(t_{0}^{+}) - \pi
\]  

(2.24)

where \( \pi \geq 0 \) is a scalar constant. It follows that \( \lambda_{s_{n}}^{*}(t_{0}^{+}) = \lambda_{s_{n}}^{*}(t_{0}) + \pi < 0 \) and \( \lambda_{s_{n+1}}^{*}(t_{0}^{+}) = \lambda_{s_{n+1}}^{*}(t_{0}) + \pi > 0 \). Since the constraint \( s_{n}(t) s_{n+1}(t) \leq 0 \) is not an explicit function of time, we have

\[
\lambda_{s_{n}}^{*}(t_{0}) u_{n}^{*}(t_{0}) + \lambda_{s_{n+1}}^{*}(t_{0}) u_{n+1}^{*}(t_{0}) = \lambda_{s_{n}}^{*}(t_{0}^{+}) u_{n}^{*}(t_{0}^{+}) + \lambda_{s_{n+1}}^{*}(t_{0}^{+}) u_{n+1}^{*}(t_{0}^{+})
\]  

(2.25)

On the other hand, \( u_{n}^{*}(t_{0}^{+}) \leq 0 \) and \( u_{n+1}^{*}(t_{0}^{+}) \geq 0 \), since agents \( n \) and \( n + 1 \) must either come to rest or reverse their motion after making contact, hence

\[
\lambda_{s_{n}}^{*}(t_{0}^{+}) u_{n}^{*}(t_{0}^{+}) + \lambda_{s_{n+1}}^{*}(t_{0}^{+}) u_{n+1}^{*}(t_{0}^{+}) \geq 0. \]  

This violates (2.25), since

\[
\lambda_{s_{n}}^{*}(t_{0}) u_{n}^{*}(t_{0}) + \lambda_{s_{n+1}}^{*}(t_{0}) u_{n+1}^{*}(t_{0}) < 0.
\]

This contradiction implies that \( s_{n}(t) s_{n+1}(t) = 0 \) cannot be active and we conclude that \( s_{n}^{*}(t) \neq s_{n+1}^{*}(t) \) for \( t \in [0, T] \), \( n = 1, \ldots, N - 1 \). Moreover, if one of the two agents is at rest when \( s_{n}^{*}(t_{0}) = s_{n+1}^{*}(t_{0}) \), the same argument still holds since it is still true that

\[
\lambda_{s_{n}}^{*}(t_{0}) u_{n}^{*}(t_{0}) + \lambda_{s_{n+1}}^{*}(t_{0}) u_{n+1}^{*}(t_{0}) < 0.
\]  

Based on this analysis, the optimal control \( u_{n}^{*}(t) \) depends entirely on the sign of \( \lambda_{s_{n}}^{*}(t) \) and, in light of Propositions 2.1-2.3, the solution of the problem reduces to determining:

(i) **switching points** in \([0, L]\) where an agent switches from \( u_{n}^{*}(t) = \pm 1 \) to either \( \mp 1 \) or 0; or from \( u_{n}^{*}(t) = 0 \) to either \( \pm 1 \), and (ii) if an agent switches from \( u_{n}^{*}(t) = \pm 1 \) to 0, **waiting times** until the agent switches back to a speed \( u_{n}^{*}(t) = \pm 1 \). In other words, the full solution is characterized by two parameter vectors for each agent \( n \): \( \theta_{n} = [\theta_{n,1}, \ldots, \theta_{n, n}]^{T} \) and \( w_{n} = [w_{n,1}, \ldots, w_{n, n}]^{T} \), where \( \theta_{n, \xi} \in (0, L) \) denotes the \( \xi \)th location where agent \( n \) changes its speed from \( \pm 1 \) to 0 and \( w_{n, \xi} \geq 0 \) denotes the time (which is possibly null) that agent \( n \) dwell on \( \theta_{n, \xi} \). Note that \( n \) is generally not known a priori and depends on the time horizon \( T \). In addition, we always assume that agent \( n \) reverses its direction.
after leaving the switching point $\theta_{n,\xi}$ with respect to the one it had when reaching $\theta_{n,\xi}$. This seemingly excludes the possibility of an agent’s control following a sequence $1, 0, 1$ or $1, 0, 1$. However, these two motion behaviors can be captured as two adjacent switching points approaching each other: when $|\theta_{n,\xi} - \theta_{n,\xi+1}| \to 0$, the agent control follows the sequence $1, 0, 1$ or $1, 0, 1$, and the waiting time associated with $u^*_n(t) = 0$ is $w_{n,\xi} + w_{n,\xi+1}$.

For simplicity, we will assume that $s_n(0) = 0$, so that it follows from Proposition 2.1 that $u^*_n(0) = 1$, $n = 1, \ldots, N$. Therefore, $\theta_{n,1}$ corresponds to the optimal control switching from 1 to 0. Furthermore, $\theta_{n,\xi}$ with $\xi$ odd (even) always corresponds to $u^*_n(t)$ switching from 1 to 0 ( 1 to 0.) Thus, we have the following constraints on the switching locations for all $\xi = 2, \ldots, n$:

$$
\begin{cases}
\theta_{n,\xi} \leq \theta_{n,\xi-1}, & \text{if } \xi \text{ is even} \\
\theta_{n,\xi} \geq \theta_{n,\xi-1}, & \text{if } \xi \text{ is odd}.
\end{cases}
$$

(2.26)

It is now clear that the behavior of each agent under the optimal control policy is that of a hybrid system whose dynamics undergo switches when $u^*_n(t)$ changes from $\pm 1$ to 0 and from 0 to $\mp 1$ or when $R_i(t)$ reaches or leaves the boundary value $R_i = 0$. As a result, we are faced with a parametric optimization problem for a system with hybrid dynamics. This is a setting where one can apply the generalized theory of Infinitesimal Perturbation Analysis (IPA) in (Cassandras et al., 2010),(Wardi et al., 2010) to conveniently obtain the gradient of the objective function $J$ in (2.7) with respect to the vectors $\theta$ and $w$, and therefore, determine (generally, locally) optimal vectors $\theta^*$ and $w^*$ through a gradient-based optimization approach. Note that this is done on line, i.e., the gradient is evaluated by observing a trajectory with given $\theta$ and $w$ over $[0, T]$ based on which $\theta$ and $w$ are adjusted until convergence is attained using standard gradient-based algorithms.

**Remark 2.1.** If the agent dynamics in (2.1) are replaced by a model such as $s_n(t) = g_n(s_n) + b_n u_n(t)$, observe that (2.12) still holds. The difference lies in (2.10) which would involve a dependence on $\frac{dg_n(s_n)}{ds_n}$ and further complicate the associated two-point-boundary-value problem. However, since the optimal solution is also defined by parameter vectors $\theta_n = [\theta_{n,1}, \ldots, \theta_{n,n}]^T$ and $w_n = [w_{n,1}, \ldots, w_{n,n}]^T$ for each agent $n$, we can still apply the IPA approach presented in the next section.
2.3 Infinitesimal Perturbation Analysis (IPA)

Our analysis thus far has shown that, on an optimal trajectory, the agent moves at full speed, dwells on a switching point (possibly for zero time) and never reaches either boundary point, i.e., $0 < s^*_n(t) < L$. Thus, the $n$th agent’s movement can be parameterized through $	heta_n = [\theta_{n,1}, \ldots, \theta_{n,n}]^T$ and $w_n = [w_{n,1}, \ldots, w_{n,n}]^T$ where $\theta_{n,\xi}$ is the $\xi$th control switching point and $w_{n,\xi}$ is the waiting time for this agent at the $\xi$th switching point. Therefore, the solution of problem P1 reduces to the determination of optimal parameter vectors $\theta^*_n$ and $w^*_n$, $n = 1, \ldots, N$. As we pointed out, the agent’s optimal behavior defines a hybrid system, and the switching locations translate to switching times between particular modes of this system. This is similar to switching-time optimization problems, e.g., (Egerstedt et al., 2006),(Shaikh and Caines, 2007),(Xu and Antsaklis, 2004), except that we can only control a subset of mode switching times. We make use of IPA in part to exploit robustness properties that the resulting gradients possess (Yao and Cassandras, 2011); specifically, we will show that they do not depend on the uncertainty model parameters $A_i$, $i = 1, \ldots, M$, and may therefore be used without any detailed knowledge of how uncertainty affects the mission space.

2.3.1 Single-agent solution with $a = 0$ and $b = L$

To maintain some notational simplicity, we begin with a single agent who can move on the entire mission space $[0, L]$ and will then provide the natural extension to multiple agents and a mission space limited to $[a, b] \subset [0, L]$. We present the associated hybrid automaton model for this single-agent system operating on an optimal trajectory. Our goal is to determine $\nabla J(\theta, w)$, the gradient of the objective function $J$ in (2.7) with respect to $\theta$ and $w$, which can then be used in a gradient-based algorithm to obtain optimal parameter vectors $\theta^*_n$ and $w^*_n$, $n = 1, \ldots, N$. We will apply IPA, which provides a formal way to obtain state and event time derivatives with respect to parameters of hybrid systems, from which we can subsequently obtaining $\nabla J(\theta, w)$.

Hybrid automaton model. We use a standard definition of a hybrid automaton (e.g.,
see (Cassandras and Lygeros, 2007)) as the formalism to model the system described above. Thus, let $q \in Q$ (a countable set) denote the discrete state (or mode) and $x \in X \subseteq \mathbb{R}^n$ denote the continuous state. Let $v \in \mathcal{Y}$ (a countable set) denote a discrete control input and $u \in U \subseteq \mathbb{R}^m$ a continuous control input. Similarly, let $\delta \in \Delta$ (a countable set) denote a discrete disturbance input and $d \in D \subseteq \mathbb{R}^p$ a continuous disturbance input. The state evolution is determined by means of (i) a vector field $f : Q \times X \times U \times D \rightarrow X$, (ii) an invariant (or domain) set $Inv : Q \times \mathcal{Y} \times \Delta \rightarrow 2^X$, (iii) a guard set $Guard : Q \times X \times \mathcal{Y} \times \Delta \rightarrow 2^X$, and (iv) a reset function $r : Q \times X \times \mathcal{Y} \times \Delta \rightarrow X$. The system remains at a discrete state $q$ as long as the continuous (time-driven) state $x$ does not leave the set $Inv(q, v, \delta)$. If $x$ reaches a set $Guard(q, q', v, \delta)$ for some $q' \in Q$, a discrete transition can take place. If this transition does take place, the state instantaneously resets to $(q', x')$ where $x'$ is determined by the reset map $r(q, q', x, v, \delta)$. Changes in $v$ and $\delta$ are discrete events that either enable a transition from $q$ to $q'$ by making sure $x \in Guard(q, q', v, \delta)$ or force a transition out of $q$ by making sure $x \not\in Inv(q, v, \delta)$. We will classify all events that cause discrete state transitions in a manner that suits the purposes of IPA. Since our problem is set in a deterministic framework, $\delta$ and $d$ will not be used.

We show in Fig. 2.2 a partial hybrid automaton model of the single-agent system where $a = 0$ and $b = L$. Since there is only one agent, we set $s(t) = s_1(t)$, $u(t) = u_1(t)$ and $\theta = \theta_1$ for simplicity. Due to the size of the overall model, Fig. 2.2 is limited to the behavior of the agent with respect to a single $\alpha_i$, $i \in \{1, \ldots, M\}$ and ignores modes where the agent dwells on the switching points (these, however, are included in our extended analysis in Section 2.3.2.) The model consists of 14 discrete states (modes) and is symmetric in the sense that states 1–7 correspond to the agent operating with $u(t) = 1$, and states 8–14 correspond to the agent operating with $u(t) = 1$. States where $u(t) = 0$ are omitted since we do not include the waiting time parameter $w = w_1$ here. The events that cause state transitions can be placed in three categories: (i) The value of $R_i(t)$ becomes 0 and triggers a switch in the dynamics of (2.6). This can only happen when $R_i(t) > 0$ and $\dot{R}_i(t) = A_i \cdot B p_i(s(t)) < 0$ (e.g., in states 3 and 4), causing a transition to state 7 in which the invariant condition is
$R_i(t) = 0$. (ii) The agent reaches a switching location, indicated by the guard condition $s(t) = \theta_\xi$ for any $\xi = 1, \ldots, n$. In these cases, a transition results from a state $z$ to $z + 1$ if $z = 1, \ldots, 6$ and to $z + 7$ otherwise. (iii) The agent position reaches one of several critical values that affect the dynamics of $R_i(t)$ while $R_i(t) > 0$. Specifically, when $s(t) = \alpha_i + r$, the value of $p_i(s(t))$ becomes strictly positive and $\dot{R}_i(t) = A_i \beta p_i(s(t)) > 0$, as in the transition $5 \rightarrow 6$. Subsequently, when $s(t) = \alpha_i + (1/\beta) A_i/B$, as in the transition $6 \rightarrow 7$, the value of $p_i(s(t))$ becomes sufficiently large to cause $\dot{R}_i(t) = A_i \beta p_i(s(t)) < 0$ so that a transition due to $R_i(t) = 0$ becomes feasible at this state. Similar transitions occur when $s(t) = \alpha_i$, $s(t) = \alpha_i + r(1/\beta) A_i/B$, and $s(t) = \alpha_i + r$. The latter results in state 6 where $\dot{R}_i(t) = A_i > 0$ and the only feasible event is $s(t) = \theta_\xi$, $\xi$ odd, when a switch must occur and a transition to state 13 takes place (similarly for state 8).

IPA review. Before proceeding, we provide a brief review of the IPA framework for general stochastic hybrid systems as presented in (Cassandras et al., 2010). The purpose of IPA is to study the behavior of a hybrid system state as a function of a parameter.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure2.png}
\caption{Hybrid automaton for each $\alpha_i$. Red arrows represent events when the control switches between 1 and 1. Blue arrows represent events when $R_i$ becomes 0. Black arrows represent all other events.}
\end{figure}
vector \( \theta \in \Theta \) for a given compact, convex set \( \Theta \subset \mathbb{R}^l \). Let \( \{ \tau_k(\theta) \}, \ k = 1, \ldots, K, \) denote the occurrence times of all events in the state trajectory. For convenience, we set \( \tau_0 = 0 \) and \( \tau_{K+1} = T \). Over an interval \( [\tau_k(\theta), \tau_{k+1}(\theta)) \), the system is at some mode during which the time-driven state satisfies \( \dot{x} = f_k(x, \theta, t) \). An event at \( \tau_k \) is classified as (i) \textit{Exogenous} if it causes a discrete state transition independent of \( \theta \) and satisfies \( \frac{d\tau_k}{d\theta} = 0 \); (ii) \textit{Endogenous}, if there exists a continuously differentiable function \( g_k : \mathbb{R}^n \times \Theta \rightarrow \mathbb{R} \) such that \( \tau_k = \min\{t > \tau_{k-1} : g_k(x(\theta, t), \theta) = 0\} \); and (iii) \textit{Induced} if it is triggered by the occurrence of another event at time \( \tau_m \leq \tau_k \). IPA specifies how changes in \( \theta \) influence the state \( x(\theta, t) \) and the event times \( \tau_k(\theta) \) and, ultimately, how they influence interesting performance metrics which are generally expressed in terms of these variables.

Given \( \theta = [\theta_1, \ldots, \theta]^T \), we use the Jacobian matrix notation: \( x'(t) \equiv \frac{\partial x(\theta, t)}{\partial \theta} \), \( \tau_k' \equiv \frac{\partial \tau_k(\theta)}{\partial \theta} \), \( k = 1, \ldots, K \), for all state and event time derivatives. It is shown in (Cassandras et al., 2010) that \( x'(t) \) satisfies:

\[
\frac{d}{dt} x'(t) = \frac{\partial f_k(t)}{\partial x} x'(t) + \frac{\partial f_k(t)}{\partial \theta} \tag{2.27}
\]

for \( t \in [\tau_k, \tau_{k+1}) \) with boundary condition:

\[
x'(\tau_k^+) = x'(\tau_k^-) + \left[ f_k(1(\tau_k^-) \quad f_k(\tau_k^+)) \right] \tau_k' \tag{2.28}
\]

for \( k = 0, \ldots, K \). In addition, in (2.28), the gradient vector for each \( \tau_k \) is \( \tau_k' = 0 \) if the event at \( \tau_k \) is exogenous and

\[
\tau_k' = \left[ \frac{\partial g_k}{\partial x} f_k(\tau_k^-) \right]^{-1} \left( \frac{\partial g_k}{\partial \theta} + \frac{\partial g_k}{\partial x} x'(\tau_k^-) \right) \tag{2.29}
\]

if the event at \( \tau_k \) is endogenous (i.e., \( g_k(x(\theta, \tau_k^-), \theta) = 0 \)), defined as long as \( \frac{\partial g_k}{\partial x} f_k(\tau_k^-) \neq 0 \).

**IPA equations.** To clarify the presentation, we first note that \( i = 1, \ldots, M \) is used to index the points where uncertainty is measured; \( \xi = 1, \ldots, \) indexes the components of the parameter vector; and \( k = 1, \ldots, K \) indexes event times. In order to apply the three fundamental IPA equations (2.27)-(2.29) to our system, we use the state vector \( x(t) = [s(t), R_1(t), \ldots, R_M(t)]^T \) and parameter vector \( \theta = [\theta_1, \ldots, \theta]^T \). We then identify all events
that can occur in Fig. 2.2 and consider intervals \([\tau_k, \tau_{k+1})\) over which the system is in one of the 14 states shown for each \(i = 1, \ldots, M\). Applying (2.27) to \(s(t)\) with \(f_k(t) = 1\) for states \(i = 1, \ldots, M\) and \(f_k(t) = 0\) due to (2.1) and (2.12), the solution yields the gradient vector \(\nabla s(t) = \left[ \frac{\partial s_{\theta_1}}{\partial \theta_1}(t), \ldots, \frac{\partial s_{\theta_M}}{\partial \theta_M}(t) \right]^T\), where

\[
\frac{\partial s}{\partial \theta_{\xi}}(t) = \frac{\partial s}{\partial \theta_{\xi}}(\tau_k^+), \quad \text{for } t \in [\tau_k, \tau_{k+1})
\] (2.30)

for all \(k = 1, \ldots, K\), i.e., for all states \(z(t) \in \{1, \ldots, 14\}\).

Similarly, let \(\nabla R_i(t) = [\frac{\partial R_i}{\partial \theta_1}(t), \ldots, \frac{\partial R_i}{\partial \theta_M}(t)]^T\) for \(i = 1, \ldots, M\). We note from (2.6) that \(f_k(t) = 0\) for states \(z(t) \in Z_1 \equiv \{7, 14\}\); \(f_k(t) = A_i\) for states \(z(t) \in Z_2 \equiv \{1, 6, 8, 13\}\); and \(f_k(t) = A_i \cdot Bp_i(s(t))\) for all other states which we further classify into \(Z_3 \equiv \{2, 3, 11, 12\}\) and \(Z_4 \equiv \{4, 5, 9, 10\}\). Thus, solving (2.27) and using (2.30) gives:

\[
\nabla R_i(t) = \begin{cases} 0, & \text{if } z(t) \in Z_1 \cup Z_2 \\ B \left( \frac{\partial p_i(s)}{\partial s} \right) \nabla s \tau_k^+, & \text{otherwise} \end{cases}
\]

where \(\frac{\partial p_i(s)}{\partial s} = \pm \frac{1}{r}\) as evaluated from (2.4) depending on the sign of \(\alpha_i\) \(s(t)\) at each associated automaton state.

We now turn our attention to the determination of \(\nabla s \tau_k^+\) and \(\nabla R_i(\tau_k^+)\) which are needed to evaluate \(\nabla R_i(t)\) above. To do so, we use (2.28), which involves the event time gradient vectors \(\nabla \tau_k = [\frac{\partial \tau_k}{\partial \theta_1}, \ldots, \frac{\partial \tau_k}{\partial \theta_M}]^T\) for \(k = 1, \ldots, K\) (the value of \(K\) depends on \(T\)).

Looking at Fig. 2.2, there are three readily distinguishable cases regarding the events that cause discrete state transitions:

**Case 1:** An event at time \(\tau_k\) which is neither \(R_i = 0\) nor \(s = \theta_{\xi}\), for any \(\xi = 1, \ldots, r\).

In this case, it is easy to see that the dynamics of both \(s(t)\) and \(R_i(t)\) are continuous, so that \(f_k(\tau_k) = f_k(\tau_k^+)\) in (2.28) applied to \(s(t)\) and \(R_i(t), i = 1, \ldots, M\) gives:

\[
\begin{align*}
\nabla s(\tau_k^+) &= \nabla s(\tau_k) \\
\nabla R_i(\tau_k^+) &= \nabla R_i(\tau_k), \quad i = 1, \ldots, M
\end{align*}
\] (2.31)

**Case 2:** An event \(R_i = 0\) at time \(\tau_k\). This corresponds to transitions \(3 \to 7, 4 \to 7, 10 \to 14\) and \(11 \to 14\) in Fig. 2.2 where the dynamics of \(s(t)\) are still continuous, but
the dynamics of \( R_i(t) \) switch from \( f_k(T_k) = A_i Bp_i(s(T_k)) \) to \( f_k(T_k^+) = 0 \). Thus, \( \nabla s(T_k) = \nabla s(T_k^+) \), but we need to evaluate \( \tau_k^\xi \) to determine \( \nabla R_i(T_k^+) \). Observing that this event is endogenous, (2.29) applies with \( g_k = R_i = 0 \) and we get

\[
\frac{\partial \tau_k}{\partial \theta_k} = \frac{\partial R_k}{\partial \theta_k} \frac{\tau_k}{\nabla s(T_k)}, \quad \xi = 1, \ldots, k = 1, \ldots, K
\]

It follows from (2.28) that

\[
\frac{\partial R_i}{\partial \theta_k} \frac{\tau_k^+}{\nabla s(T_k)} = \frac{\partial R_i}{\partial \theta_k} \frac{\tau_k}{\nabla s(T_k)} \frac{[A_i Bp_i(s(T_k))] \frac{\partial R_i}{\partial \theta_k} \tau_k}{A_i Bp_i(s(T_k))} = 0
\]

Thus, whenever an event occurs at \( \tau_k \) such that \( R_i(\tau_k) \) becomes zero, \( \frac{\partial R_i}{\partial \theta_k} \frac{\tau_k^+}{\nabla s(T_k)} \) is always reset to 0 regardless of \( \frac{\partial R_i}{\partial \theta_k} \frac{\tau_k}{\nabla s(T_k)} \).

Case 3: An event at time \( \tau_k \) due to a control sign change at \( s = s_k, \xi = 1, \ldots, \). This corresponds to any transition between the upper and lower part of the hybrid automaton in Fig. 2.2. In this case, the dynamics of \( R_i(t) \) are continuous and we have \( \frac{\partial R_i}{\partial \theta_k} \frac{\tau_k^+}{\nabla s(T_k)} = \frac{\partial R_i}{\partial \theta_k} \frac{\tau_k}{\nabla s(T_k)} \) for all \( i, \xi, k \). On the other hand, we have \( s(\tau_k^+) = u(\tau_k^+) = u(\tau_k) = \pm 1 \). Observing that any such event is endogenous, (2.29) applies with \( g_k = s \theta_k = 0 \) for some \( \xi = 1, \ldots, \) and we get

\[
\frac{\partial \tau_k}{\partial \theta_k} = \frac{1}{u(\tau_k)} \frac{\partial s}{\partial \theta_k}(\tau_k)
\]

Combining (2.32) with (2.28) and recalling that \( u(\tau_k) = u(\tau_k) \), we have

\[
\frac{\partial s}{\partial \theta_k}(\tau_k^+) = \frac{\partial s}{\partial \theta_k}(\tau_k) + [u(\tau_k^+) \frac{1}{u(\tau_k)} \frac{\partial s}{\partial \theta_k}(\tau_k)] = 2
\]

where \( \frac{\partial s}{\partial \theta_k}(\tau_k) = 0 \) because \( \frac{\partial s}{\partial \theta_k}(0) = 0 = \frac{\partial s}{\partial \theta_k}(t) \) for all \( t \in [0, \tau_k] \), since the position of the agent cannot be affected by \( \theta_k \) prior to this event.

In this case, we also need to consider the effect of perturbations to \( \theta_j \) for \( j < \xi \), i.e., prior to the current event time \( \tau_k \) (clearly, for \( j > \xi \), \( \frac{\partial s}{\partial \theta_j}(\tau_k^+) = 0 \) since the current position of the agent cannot be affected by future events.) Observe that since \( g_k = s \theta_k = 0 \), we have \( \frac{\partial g_k}{\partial \theta_j} = 0 \) for \( j \neq \xi \) and (2.29) gives \( \frac{\partial r_k}{\partial \theta_j} = \frac{1}{u(\tau_k)} \frac{\partial s}{\partial \theta_j}(\tau_k) \), so that using this in (2.28) we
Combining the above results, the components of $\nabla s(\tau_k^+)$ where $\tau_k$ is the event time when $s(\tau_k) = \theta_\xi$ for some $\xi$, are given by

$$\frac{\partial s}{\partial \theta_j}(\tau_k^+) = \frac{\partial s}{\partial \theta_j}(\tau_k) \left[ \frac{u(\tau_k)}{\tau_k} \right] \frac{\partial u}{\partial \theta_j}(\tau_k) = \frac{\partial s}{\partial \theta_j}(\tau_k)$$

It follows from (2.30) and the analysis of all three cases above that $\frac{\partial s}{\partial \theta_\xi}(t)$ for all $\xi$ is constant throughout an optimal trajectory except at transitions caused by control switching locations (Case 3). In particular, for the $k$th event corresponding to $s(\tau_k) = \theta_\xi$, $t \in [\tau_k, T]$, if $u(t) = 1$, then $\frac{\partial s}{\partial \theta_\xi}(t) = 2$ if $\xi$ is odd, and $\frac{\partial s}{\partial \theta_\xi}(t) = 2$ if $\xi$ is even; similarly, if $u(t) = 1$, then $\frac{\partial s}{\partial \theta_\xi}(t) = 2$ if $\xi$ is odd and $\frac{\partial s}{\partial \theta_\xi}(t) = 2$ if $\xi$ is even. In summary, we can write:

$$\frac{\partial s}{\partial \theta_\xi}(t) = \begin{cases} (1)^\xi \cdot 2u(t) & t \geq \tau_k, \xi = 1, \ldots, \\ 0 & t < \tau_k \end{cases}$$

Finally, we can combine (2.34) with our results for $\frac{\partial R_i}{\partial \theta_\xi}(t)$ in all three cases above. Letting $s(\tau_i) = \theta_\xi$, we obtain the following expression for $\frac{\partial R_i}{\partial \theta_\xi}(t)$ for all $k \geq l$, $t \in [\tau_k, \tau_{k+1})$:

$$\frac{\partial R_i}{\partial \theta_\xi}(t) = \frac{\partial R_i}{\partial \theta_\xi}(\tau_k^+) + \begin{cases} 0 & \text{if } z(t) \in Z_1 \cup Z_2 \\ (1)^{\xi+1} \cdot \frac{2R_i}{r} u(\tau_k^+) \cdot (t - \tau_k) & \text{if } z(t) \in Z_3 \\ (1)^{\xi+1} \cdot \frac{2R_i}{r} u(\tau_k^+) \cdot (t - \tau_k) & \text{if } z(t) \in Z_4 \\ \end{cases}$$

with boundary condition

$$\frac{\partial R_i}{\partial \theta_\xi}(\tau_k^+) = \begin{cases} 0 & \text{if } z(\tau_k^+) \in Z_1 \\ \frac{\partial R_i}{\partial \theta_\xi}(\tau_k) & \text{otherwise} \end{cases}$$

**Objective Function Gradient Evaluation.** Based on our analysis, the objective function (2.7) in problem $\textbf{P1}$ can now be written as $J(\theta)$, a function of $\theta$ instead of $u(t)$ and we can rewrite it as

$$J(\theta) = \frac{1}{T} \sum_{i=1}^M \sum_{k=0}^K \frac{\tau_{k+1}^{(\theta)}}{\tau_k^{(\theta)}} \int_{\tau_k^{(\theta)}}^{\tau_{k+1}^{(\theta)}} R_i(t, \theta) \, dt$$
where we have explicitly indicated the dependence on $\theta$. We then obtain:

$$
\nabla J(\theta) = \frac{1}{T} \sum_{i=1}^{M} \sum_{k=0}^{K} \left( \int_{\tau_k}^{\tau_{k+1}} \nabla R_i(t) \, dt + R_i(\tau_{k+1}) \nabla \tau_{k+1} + R_i(\tau_k) \nabla \tau_k \right)
$$

Observing the cancelation of all terms of the form $R_i(\tau_k) \nabla \tau_k$ for all $k$ (with $\tau_0 = 0$, $\tau_{K+1} = T$ fixed), we finally get

$$
\nabla J(\theta) = \frac{1}{T} \sum_{i=1}^{M} \sum_{k=0}^{K} \int_{\tau_k(\theta)}^{\tau_{k+1}(\theta)} \nabla R_i(t) \, dt.
$$

The evaluation of $\nabla J(\theta)$ therefore depends entirely on $\nabla R_i(t)$, which is obtained from (2.35)-(2.36) and the observable event times $\tau_k$, $k = 1, \ldots, K$, given initial conditions $s(0) = 0$, $R_i(0)$ for $i = 1, \ldots, M$ and $\nabla R_i(0) = 0$. Since $\nabla R_i(t)$ itself depends only on the event times $\tau_k$, $k = 1, \ldots, K$, the gradient $\nabla J(\theta)$ is obtained by observing the switching times in a trajectory over $[0, T]$ characterized by the vector $\theta$.

### 2.3.2 Multi-agent solution where $a \geq 0$ and $b \leq L$

Next, we extend the results obtained in the previous section to the general multi-agent problem where we also allow $a \geq 0$ and $b \leq L$. Recall that we require $0 \leq a \leq r_n$ and $L \ r_m \leq b \leq L$, for at least some $n, m = 1, \ldots, N$ since, otherwise, controlling agent movement cannot affect $R_i(t)$ for all $\alpha_i$ located outside the sensing range of agents. We now include both parameter vectors $\theta_n = [\theta_{n,1}, \ldots, \theta_{n,n}]^T$ and $w_n = [w_{n,1}, \ldots, w_{n,n}]^T$ for each agent $n$ and, for notational simplicity, concatenate them to construct $\theta = [\theta_1, \ldots, \theta_N]^T$ and $w = [w_1, \ldots, w_N]^T$. The solution of problem $P1$ reduces to the determination of optimal parameter vectors $\theta^*$ and $w^*$ and we will use IPA to evaluate $\nabla J(\theta, w) = \left[ \frac{dJ(\theta, w)}{d\theta} \frac{dJ(\theta, w)}{dw} \right]^T$. Similar to (2.72), it is clear that this depends on $\nabla R_i(t) = \left[ \frac{\partial R_i(t)}{\partial \theta} \frac{\partial R_i(t)}{\partial w} \right]^T$ and the event times $\tau_k$, $k = 1, \ldots, K$, observed on a trajectory over $[0, T]$ with given $\theta$ and $w$.

**IPA equations.** We begin by recalling the dynamics of $R_i(t)$ in (2.6) which depend on the relative positions of all agents with respect to $\alpha_i$ and change at time instants $\tau_k$ such that either $R_i(\tau_k) = 0$ with $R_i(\tau_k) > 0$ or $A_i > BP_i(s(\tau_k))$ with $R_i(\tau_k) = 0$. Moreover,
using (2.1) and our earlier Hamiltonian analysis, the dynamics of $s_n(t)$, $n = 1, \ldots, N$, in an optimal trajectory can be expressed as follows. Define $\Theta_{n, \xi} = (\theta_{n, \xi}, 1, \theta_{n, \xi})$ if $\xi$ is odd and $\Theta_{n, \xi} = (\theta_{n, \xi}, \theta_{n, \xi}, 1)$ if $\xi$ is even to be the $\xi$th interval between successive switching points for any $n = 1, \ldots, N$, where $\theta_{n, 0} = s_n(0)$. Then, for $\xi = 1, 2, \ldots,$

$$\dot{s}_n(t) = \begin{cases} 
1 & s_n(t) \in \Theta_{n, \xi}, \; \xi \text{ odd} \\
1 & s_n(t) \in \Theta_{n, \xi}, \; \xi \text{ even} \\
0 & \text{otherwise}
\end{cases} \tag{2.38}$$

where transitions for $s_n(t)$ from $\pm 1$ to $\mp 1$ are incorporated by treating them as cases where $w_{n, \xi} = 0$, i.e., no dwelling at a switching point $\theta_{n, \xi}$ (in which case $\dot{s}_n(t) = 0$.) We can now concentrate on all events causing switches either in the dynamics of any $R_i(t)$, $i = 1, \ldots, M$, or the dynamics of any $s_n(t)$, $n = 1, \ldots, N$. From (2.28), any other event at some time $\tau_k$ in this hybrid system cannot modify the values of $\nabla R_i(t) = \left[ \frac{\partial R_i(t)}{\partial \theta} \frac{\partial R_i(t)}{\partial w} \right]^T$ or $\nabla s_n(t) = \left[ \frac{\partial s_n(t)}{\partial \theta_n} \frac{\partial s_n(t)}{\partial w_n} \right]^T$ at $t = \tau_k$.

First, applying (2.27) to $s_n(t)$ with $f_k(t) = 1$, 1 or 0 due to (2.38), the solution yields

$$\nabla s_n(t) = \nabla s_n(\tau_k^+), \; \text{for} \; t \in [\tau_k, \tau_{k+1}) \tag{2.39}$$

for all $k = 1, \ldots, K$, $n = 1, \ldots, N$. Similarly, applying (2.27) to $R_i(t)$ and using (2.6) gives:

$$\frac{\partial R_i}{\partial \theta_{n, \xi}}(t) = \frac{\partial R_i}{\partial \theta_{n, \xi}}(\tau_k^+)$$

$$\begin{cases} 
0 & \text{if } R_i(t) = 0, \quad A_i < BP_i(s(t)) \\
B \prod_{d \neq n} \left( 1 - p_i(s_d(t)) \right) \left( \frac{\partial p_i(s_n)}{\partial s_n} \right) \frac{\partial s_n(\tau_k^+)}{\partial \theta_{n, \xi}} \cdot (t, \tau_k) & \text{otherwise}
\end{cases} \tag{2.40}$$
and

$$\frac{\partial R_i}{\partial w_{n,\xi}}(t) = \frac{\partial R_i}{\partial w_{n,\xi}} \tau_k^+ \right\} \begin{cases} 0, & \text{if } R_i(t) = 0, \\ \frac{\prod_{d \neq n} (1 - p_1(s_d(t))) \frac{\partial s_n(\tau_k^+)}{\partial w_{n,\xi}} \cdot (t, \tau_k)}{A_i < BP_i(s(t))}, & \text{otherwise} \end{cases}$$

Thus, it remains to determine the components of \(\nabla s_n(\tau_k^+)\) and \(\nabla R_i(\tau_k^+)\) in (2.39)-(2.41) using (2.28). This involves the event time gradient vectors \(\nabla \tau_k = \left[\frac{\partial \tau_k}{\partial \theta} \frac{\partial \tau_k}{\partial \omega}\right]^T\) for \(k = 1, \ldots, K\), which will be determined through (2.29). There are three possible cases regarding the events that cause switch in the dynamics of \(R_i(t)\) or \(s_n(t)\) as mentioned above:

**Case 1:** An event at time \(\tau_k\) such that \(\dot{R}_i(t)\) switches from \(\dot{R}_i(t) = 0\) to \(\dot{R}_i(t) = A_i \cdot BP_i(s(t))\). In this case, it is easy to see that the dynamics of both \(s_n(t)\) and \(R_i(t)\) are continuous, so that \(f_{k_1}(\tau_k) = f_{k_2}(\tau_k^+)\) in (2.28) applied to \(s_n(t)\) and \(R_i(t)\), \(i = 1, \ldots, M\), \(n = 1, \ldots, N\), and we get

\[
\nabla s_n(\tau_k^+) = \nabla s_n(\tau_k) , \ n = 1, \ldots, N \\
\nabla R_i(\tau_k^+) = \nabla R_i(\tau_k) , \ i = 1, \ldots, M
\]

**Case 2:** An event at time \(\tau_k\) such that \(\dot{R}_i(t)\) switches from \(\dot{R}_i(t) = A_i \cdot BP_i(s(t))\) to \(\dot{R}_i(t) = 0\), i.e., \(R_i(\tau_k)\) becomes zero. In this case, we need to first evaluate \(\nabla \tau_k\) from (2.29) in order to determine \(\nabla R_i(\tau_k^+)\) through (2.28). Observing that this event is endogenous, (2.29) applies with \(g_k = R_i = 0\) and we get

\[
\nabla \tau_k = \frac{\nabla R_i(\tau_k)}{A_i \tau_k} \cdot \frac{BP_i(s(\tau_k))}{BP_i(\tau_k)}
\]

It follows from (2.28) that

\[
\nabla R_i(\tau_k^+) = \nabla R_i(\tau_k) \cdot \frac{[A_i \tau_k]}{A_i \tau_k} \cdot \frac{BP_i(s(t))) \nabla R_i(\tau_k)}{BP_i(\tau_k)} = 0
\]
Thus, $\nabla R_i(\tau_k^\dagger)$ is always reset to 0 regardless of $\nabla R_i(\tau_k)$. In addition, (2.42) holds, since the the dynamics of $s_n(t)$ are continuous at time $\tau_k$.

Case 3: An event at time $\tau_k$ such that the dynamics of $s_n(t)$ switch from $\pm 1$ to 0, or from 0 to $\pm 1$. Clearly, (4.9) holds since the the dynamics of $R_i(t)$ are continuous at this time. However, determining $\nabla s_n(\tau_k^\dagger)$ is more elaborate and requires us to consider its components separately, first $\frac{\partial s_n(\tau_k^\dagger)}{\partial \theta_n}$ and then $\frac{\partial s_n(\tau_k^\dagger)}{\partial \omega_n}$.

Case 3.1: Evaluation of $\frac{\partial s_n(\tau_k^\dagger)}{\partial \theta_n}$.

Case 3.1.1: An event at time $\tau_k$ such that the dynamics of $s_n(t)$ in (2.38) switch from $\pm 1$ to 0. This is an endogenous event and (2.29) applies with $g_k = s_n$ and $\theta_n, \xi = 0$ for some $\xi = 1, \ldots, n$ and we have:

$$\frac{\partial \tau_k}{\partial \theta_{n,\xi}} = \frac{1}{u_n(\tau_k)} \frac{\partial s_n(\tau_k)}{\partial \theta_{n,\xi}}$$

(2.46)

and (2.28) yields

$$\frac{\partial s_n}{\partial \theta_{n,\xi}}(\tau_k^\dagger) = \frac{\partial s_n}{\partial \theta_{n,\xi}}(\tau_k) + \left[ u_n(\tau_k) \right] \left[ 0 \right] \frac{1}{u_n(\tau_k)} \frac{\partial s_n}{\partial \theta_{n,\xi}}(\tau_k) = 1$$

(2.47)

As in Case 3 of Section 2.3.1, we also need to consider the effect of perturbations to $\theta_j$ for $j < \xi$, i.e., prior to the current event time $\tau_k$ (clearly, for $j > \xi$, $\frac{\partial s_n}{\partial \theta_j}(\tau_k^\dagger) = 0$ since the current position of the agent cannot be affected by future events.) Observe that $\frac{\partial g_k}{\partial \theta_j} = 0$, therefore, (2.29) becomes

$$\frac{\partial \tau_k}{\partial \theta_{n,j}} = \frac{\partial \tau_k}{\partial \theta_{n,j}}$$

(2.48)

and using this in (2.28) gives:

$$\frac{\partial s_n}{\partial \theta_{n,j}}(\tau_k^\dagger) = \frac{\partial s_n}{\partial \theta_{n,j}}(\tau_k) + \left[ u_n(\tau_k) \right] \left[ 0 \right] \frac{\partial s_n}{\partial \theta_{n,j}}(\tau_k) = 0$$

(2.49)

Thus, combining the above results, when $s_n(\tau_k) = \theta_{q,\xi}$ for some $\xi$ and the agent switches from $\pm 1$ to 0, we have

$$\frac{\partial s_n}{\partial \theta_{n,j}}(\tau_k^\dagger) = \begin{cases} 0, & \text{if } j \neq \xi \\ 1, & \text{if } j = \xi \end{cases}$$

(2.50)

Case 3.1.2: An event at time $\tau_k$ such that the dynamics of $s_n(t)$ in (2.38) switch from
0 to ±1. This is an induced event since it is triggered by the occurrence of some other endogenous event when the agent switches from ±1 to 0 (see Case 3.1.1 above.) Suppose the agent starts from an initial position \( s_n(0) = a \) with \( u_n(0) = 1 \) and \( \tau_k \) is the time the agent switches from the 0 to ±1 at the switching point \( \theta_{n,\xi} \). If \( \theta_{n,\xi} \) is such that \( u_n(\tau_k^+) = 1 \), then \( \xi \) is even and \( \tau_k \) can be calculated as follows:

\[
\tau_k = (\theta_{n,1} a) + w_{n,1} + (\theta_{n,1} \theta_{n,2}) + w_{n,2} + \ldots + (\theta_{n,\xi} \theta_{n,\xi}) + w_{n,\xi} \tag{2.51}
\]

\[
= 2 \left( \sum_{v=1, v \text{ odd}}^{\xi} \theta_{n,v} \sum_{v=2, v \text{ even}}^{\xi} \theta_{n,v} + \sum_{v=1}^{\xi} w_{n,v} \theta_{n,\xi} \right)
\]

Similarly, if \( \theta_{n,\xi} \) is the switching point such that \( u_n(\tau_k^+) = 1 \), then \( \xi \) is odd and we get:

\[
\tau_k = 2 \left( \sum_{v=1, v \text{ odd}}^{\xi} \theta_{n,v} \sum_{v=2, v \text{ even}}^{\xi} \theta_{n,v} + \sum_{v=1}^{\xi} w_{n,v} \theta_{n,\xi} \right) \tag{2.52}
\]

We can then directly obtain \( \frac{\partial \tau_k}{\partial \theta_{n,\xi}} \) as

\[
\frac{\partial \tau_k}{\partial \theta_{n,\xi}} = \text{sgn}(u(\tau_k^+)) \tag{2.53}
\]

Using (2.53) in (2.28) gives:

\[
\frac{\partial s_n}{\partial \theta_{n,\xi}}(\tau_k^+) = \frac{\partial s_n}{\partial \theta_{n,\xi}}(\theta_{n,\xi}) + \left[ \begin{array}{cc} 0 & u(\tau_k^+) \end{array} \right] \cdot \left[ \begin{array}{c} \text{sgn}(u(\tau_k^+)) \end{array} \right] = \frac{\partial s_n}{\partial \theta_{n,\xi}}(\theta_{n,\xi}) + 1 \tag{2.54}
\]

Once again, we need to consider the effect of perturbations to \( \theta_j \) for \( j < \xi \), i.e., prior to the current event time \( \tau_k \) (clearly, for \( j > \xi \), \( \frac{\partial s_n}{\partial \theta_j}(\tau_k^+) = 0 \).) In this case, from (2.51)-(2.52), we have

\[
\left\{ \begin{array}{l}
\frac{\partial \tau_k}{\partial \theta_{n,j}} = 2, \quad \text{if } j \text{ odd} \\
\frac{\partial \tau_k}{\partial \theta_{n,j}} = \frac{\partial \tau_k}{\partial \theta_{n,j}} = \frac{\partial \tau_k}{\partial \theta_{n,j}} = 2, \quad \text{if } j \text{ even}
\end{array} \right. \tag{2.55}
\]

and it follows from (2.28) that for \( j < \xi \):

\[
\frac{\partial s_n}{\partial \theta_{n,j}}(\tau_k^+) = \left\{ \begin{array}{ll}
\frac{\partial s_n}{\partial \theta_{n,j}}(\tau_k) + 2, & \text{if } u_n(\tau_k^+) = 1 \text{, } j \text{ even}, \text{ or } u_n(\tau_k^+) = 1, \text{ } j \text{ odd} \\
\frac{\partial s_n}{\partial \theta_{n,j}}(\tau_k) & \text{if } u_n(\tau_k^+) = 1, \text{ } j \text{ odd, or } u_n(\tau_k^+) = 1, \text{ } j \text{ even}
\end{array} \right. \tag{2.56}
\]

Case 3.2: Evaluation of \( \frac{\partial s_n(\tau_k^+)}{\partial u_n} \).
**Case 3.2.1:** An event at time \( \tau_k \) such that the dynamics of \( s_n(t) \) in (2.38) switch from \( \pm 1 \) to 0. This is an endogenous event and (2.29) applies with \( g_k = s_n \), \( \theta_{n,\xi} = 0 \) for some \( \xi = 1, \ldots, n \). Then, for any \( j \leq \xi \), we have:

\[
\frac{\partial \tau_k}{\partial w_{n,j}} = \frac{\partial s_n}{\partial w_{n,j}}(\tau_k) - \frac{u_n(\tau_k)}{u_n(\tau_k)} \quad (2.57)
\]

Combining (2.57) with (2.28) and since \( u_n(\tau_k) = \pm 1 \), we have

\[
\frac{\partial s_n}{\partial w_{n,j}}(\tau_k) = \frac{\partial s_n}{\partial w_{n,j}}(\tau_k) + [u_n(\tau_k) 0] - \frac{\partial s_n}{\partial w_{n,j}}(\tau_k) = 0 \quad (2.58)
\]

**Case 3.2.2:** An event at time \( \tau_k \) such that the dynamics of \( s_n(t) \) in (2.38) switch from 0 to \( \pm 1 \). As in Case 3.1.2, \( \tau_k \) is given by (2.51) or (2.52), depending on the sign of \( u_q(\tau_k^+) \). Thus, we have \( \frac{\partial \tau_k}{\partial w_{n,j}} = 1 \), for \( j \leq \xi \). Using this result in (2.28) and observing that \( \frac{\partial s_n}{\partial w_{n,j}}(\tau_k) = 0 \) from (2.58), we have

\[
\frac{\partial s_n}{\partial w_{n,j}}(\tau_k) = \frac{\partial s_n}{\partial w_{n,j}}(\tau_k) + [0 (\pm 1)] \cdot 1 = u_n(\tau_k^+) \quad (2.58)
\]

Combining the above results, we have for Case 3.2:

\[
\frac{\partial s_n}{\partial w_{n,j}}(\tau_k) = \begin{cases} 0, & \text{if } u_n(\tau_k) = \pm 1, \quad u_n(\tau_k^+) = 0 \\ \pm 1, & \text{if } u_n(\tau_k) = 0, \quad u_n(\tau_k^+) = \pm 1 \end{cases} \quad (2.60)
\]

Finally, note that \( \frac{\partial s_n}{\partial w_{n,\xi}}(t) = 0 \) for \( t \in [0, \tau_k) \), since the position of the agent \( n \) cannot be affected by \( w_{n,\xi} \) prior to such an event.

**Objective Function Gradient Evaluation.** Proceeding as in the evaluation of \( \nabla J(\theta) \) in Section 2.3.1, we are now interested in minimizing the objective function \( J(\theta, w) \) in (2.7) with respect to \( \theta \) and \( w \) and we can obtain \( \nabla J(\theta, w) = [\frac{dJ(\theta, w)}{d\theta} \frac{dJ(\theta, w)}{dw}]^T \) as

\[
\nabla J(\theta, w) = \frac{1}{T} \sum_{i=1}^{M} \sum_{k=0}^{K} \int_{\tau_k(\theta, w)}^{\tau_{k+1}(\theta, w)} \nabla R_i(t) \, dt
\]

This depends entirely on \( \nabla R_i(t) \), which is obtained from (2.40) and (2.41) and the event
times \( \tau_k, \ k = 1, \ldots, K \), given initial conditions \( s_n(0) = a \) for \( n = 1, \ldots, N \), and \( R_i(0) \) for \( i = 1, \ldots, M \). In (2.40), \( \frac{\partial R_i}{\partial \eta_{n, \xi}} \tau^+_k \) is obtained through (4.9) and (4.10), whereas \( \frac{\partial s_n(\tau^+_k)}{\partial \eta_{n, \xi}} \) is obtained through (2.39), (2.42), (2.50), and (2.56). In (2.41), \( \frac{\partial R_i}{\partial w_{n, \xi}} \tau^+_k \) is again obtained through (4.9) and (4.10), whereas \( \frac{\partial s_n(\tau^+_k)}{\partial w_{n, \xi}} \) is obtained through (2.42), and (2.60).

**Remark 2.2.** Observe that the evaluation of \( \nabla R_i(t) \), hence \( \nabla J(\theta, w) \), is independent of \( A_i, \ i = 1, \ldots, M \), i.e., the values in our uncertainty model. In fact, the dependence of \( \nabla R_i(t) \) on \( A_i, \ i = 1, \ldots, M \), manifests itself through the event times \( \tau_k, \ k = 1, \ldots, K \), that do affect this evaluation, but they, unlike \( A_i \) which may be unknown, are directly observable during the gradient evaluation process. Thus, the IPA approach possesses an inherent robustness property: there is no need to explicitly model how uncertainty affects \( R_i(t) \) in (2.6). Consequently, we may treat \( A_i \) as unknown without affecting the solution approach (the values of \( \nabla R_i(t) \) are obviously affected). We may also allow this uncertainty to be modeled through random processes \( \{A_i(t)\}, \ i = 1, \ldots, M \); in this case, however, the result of Proposition 2.3 no longer applies without some conditions on the statistical characteristics of \( \{A_i(t)\} \) and the resulting \( \tilde{\nabla} J(\theta, w) \) is an estimate of a stochastic gradient.

### 2.4 Objective Function Optimization

We now seek to obtain \( \theta^* \) and \( w^* \) minimizing \( J(\theta, w) \) through a standard gradient-based optimization scheme of the form

\[
[\theta^{l+1}, w^{l+1}]^T = [\theta^l, w^l]^T + \nabla J(\theta^l, w^l)
\]

(2.61)

where \( \{\eta^l_{\theta}\}, \{\eta^l_w\} \) are appropriate step size sequences and \( \nabla J(\theta^l, w^l) \) is the projection of the gradient \( \nabla J(\theta^l, w^l) \) onto the feasible set (the set of \( \theta^{l+1} \) satisfying the constraint (2.26), \( a \leq \theta^{l+1} \leq b \), and \( w^{l+1} \geq 0 \)). The optimization scheme terminates when \( |\nabla J(\theta, w)| < \varepsilon \) (for a fixed threshold \( \varepsilon \)) for some \( \theta \) and \( w \). Our IPA-based algorithm to obtain \( \theta^* \) and \( w^* \) minimizing \( J(\theta, w) \) is summarized in Algorithm 1 where we have adopted the Armijo method in step-size selection (see (Polak, 1997)) for \( \{\eta^l_{\theta}, \eta^l_w\} \).

One of the unusual features in (4.12) is the fact that the dimension \( n \) of \( \theta^*_n \) and \( w^*_n \) is
a priori unknown (it depends on $T$). Thus, the algorithm must implicitly determine this value along with $\theta_n^*$ and $w_n^*$. One can search over feasible values of $n \in \{1, 2, \ldots\}$ by starting either with a lower bound $n = 1$ or an upper bound to be found. The latter approach results in much faster execution and is followed in Algorithm 1. An upper bound is determined by observing that $\theta_{n,\xi}$ is the switching point where agent $n$ changes speed from 1 to 0 for $\xi$ odd and from 1 to 0 for $\xi$ even. By setting these two groups of switching points so that their distance is sufficiently small and waiting times $w_n = 0$ for each agent, we determine an approximate upper bound for $n$ as follows. First, we divide the feasible space $[a, b]$ evenly into $N$ intervals: $[a + \frac{n-1}{N} (b - a), a + \frac{n}{N} (b - a)]$, $n = 1, \ldots, N$. Define $D_n = a + \frac{2n-1}{2N} (b - a)$ to be the geometric center of each interval and set $\theta_{n,\xi} = D_n - \sigma$ if $\xi$ is even and $\theta_{n,\xi} = D_n + \sigma$ if $\xi$ is odd, so that the distance between switching points $\theta_{n,\xi}$ for $\xi$ odd and even is $2\sigma$, where $\sigma > 0$ is an arbitrarily small number, $n = 1, \ldots, N$. In addition, set $w_n = 0$. Then, $T$ must satisfy

$$\theta_{n,1} + s_n(0) + 2\sigma (n - 1) \leq T \leq \theta_{n,1} + s_n(0) + 2\sigma \quad (2.62)$$

$n = 1, \ldots, N$, where $n$ is the number of switching points agent $n$ can reach during $[0, T]$, given $\theta_{n,\xi}$ as defined above. From (2.62) and noting that $n$ is an integer, we have

$$n = \left\lceil \frac{1}{2\sigma} \left( T - \theta_{n,1} + s_n(0) \right) \right\rceil \quad (2.63)$$

where $\lceil \cdot \rceil$ is the ceiling function. Clearly, reducing $\sigma$ increases the initial number of switching points $n$ assigned to agent $n$ and $n \to \infty$ as $\sigma \to 0$. Therefore, $\sigma$ is selected sufficiently small while ensuring that the algorithm can be executed sufficiently fast.

As Algorithm 1 repeats steps 3-6, $w_{n,\xi} \geq 0$ and distances between $\theta_{n,\xi}$ for $\xi$ odd and even generally increase, so that the number of switching points agent $n$ can actually reach within $T$ decreases. In other words, as long as $\sigma$ is sufficiently small (hence, $n$ is sufficiently large), when the algorithm converges to a local minimum and stops, there exists $\zeta_n < n$, such that $\theta_{n,\xi}$ is the last switching point agent $n$ can reach within $[0, T]$, $n = 1, \ldots, N$. Observe that there generally exist $\xi$ such that $\zeta_n < \xi \leq n$ which correspond to points $\theta_{n,\xi}$
that agent \( n \) cannot reach within \((0, T]\); the associated derivatives of the cost with respect to such \( \theta_{n,\xi} \) are obviously 0, since perturbations to these \( \theta_{n,\xi} \) will not affect \( s_n(t), t \in (0, T]\) and thus the cost \( J(\theta, w) \). When \( |\nabla J(\theta, w)| < \epsilon \), we achieve a local minimum and stop, at which point the dimension of \( \theta^*_n \) and \( w^*_n \) is \( \zeta_n \).

**Algorithm 1:** IPA-based optimization algorithm to find \( \theta^* \) and \( w^* \)

1. Pick \( \sigma > 0 \) and \( \epsilon > 0 \).
2. Define \( D_n = a + \frac{2n - 1}{2N} (b - a) \), \( n = 1, \ldots, N \), and set \( \theta_{n,\xi} = D_n - \sigma \) if \( \xi \) even \( \theta_{n,\xi} = D_n + \sigma \) if \( \xi \) odd.
   
   Set \( w = [w_1, \ldots, w_N] = 0 \), where \( w_n = [w_{n,1}, \ldots, w_{n,\zeta_n}] \) and \( n = \left\lfloor \frac{1}{2\sigma} [T - \theta_{n,1} + s_n(0)] \right\rfloor \).
3. **repeat**
   4. Compute \( s_n(t), t \in [0, T] \) using \( s_n(0), (2.12), \theta \) and \( w \) for \( n = 1, \ldots, N \).
   5. Compute \( \nabla J(\theta, w) \) and update \( \theta, w \) through (4.12).
   6. **until** \( |\nabla J(\theta, w)| < \epsilon \)
   7. Set \( \theta^*_n = [\theta^*_{n,1}, \ldots, \theta^*_{n,\zeta_n}] \) and \( w^*_n = [w^*_{n,1}, \ldots, w^*_{n,\zeta_n}] \), where \( \zeta_n \) is the index of \( \theta_{n,\zeta_n} \), which is the last switching point agent \( n \) can reach within \((0, T], n = 1, \ldots, N \).

### 2.5 Numerical Experiments

Here we present some examples of persistent monitoring problems in which agent trajectories are determined using Algorithm 1. The first four are single-agent examples with \( L = 20, M = 21, \alpha_1 = 0, \alpha_M = 20 \), and the remaining sampling points are evenly spaced over \([0, 20]\). The sensing range in (2.4) is set to \( r = 4 \), the initial values of the uncertainty functions in (2.6) are \( R_i(0) = 4 \) for all \( i \), and the time horizon is \( T = 400 \). In Fig. 2·3(a) we show results where the agent is allowed to move over the entire space \([0, 20]\) and the uncertainty model is selected so that \( B = 3 \) and \( A_i = 0.1 \) for all \( i = 0, \ldots, 20 \), whereas in Fig. 2·3(b) the feasible space is limited to \([a, b]\) with \( a = r = 4 \) and \( b = L - r = 16 \). The top plot in each example shows the optimal trajectory \( s^*(t) \) obtained, while the bottom shows the cost \( J(\theta^l, w^l) \) as a function of iteration number. In Fig. 2·4, the trajectories in Fig. 2·3(a), (b) are magnified for the interval \( t \in [0, 75] \) to emphasize the presence of strictly positive waiting times at the switching points.

In Fig. 2·3(c) we show results for a case similar to Fig. 2·3(a) except that the values of \( A_i \) are selected so that \( A_0 = A_{20} = 0.5 \), while \( A_i = 0.1, i = 1, \ldots, 19 \). Note that the waiting
times at the switching points are now longer and even though it seems that the switching points are at the two end points, they are actually very close but not equal to these end points, consistent with Proposition 2.1. In Fig. 2.3(d), on the other hand, the values of $A_i$ are allowed to be random, thus dealing with a persistent monitoring problem in a stochastic mission space, where we can test the robustness of the proposed approach. In particular, each $A_i$ is treated as a piecewise constant random process $\{A_i(t)\}$ such that $A_i(t)$ takes on a fixed value sampled from an uniform distribution over $(0.075, 0.125)$ for an exponentially distributed time interval with mean 10 before switching to a new value. Note that the behavior of the system in this case is very similar to Fig. 2.3(a) where $A_i = 0.1$ for all $i = 0, \ldots, 20$ without any change in the way in which $\nabla J(\theta^l, w^l)$ is evaluated in executing (4.12). As already pointed out, this exploits a robustness property of IPA which makes the evaluation of $\nabla J(\theta^l, w^l)$ independent of the values of $A_i$. In general, however, when $A_i(t)$ is time-varying, Proposition 2.3 may no longer apply, since an extra term $\sum_i \dot{A}_i(t)$ would be present in (2.71). In such a case, $u^*_n(t)$ may be nonzero when $\lambda^*_n(t) = 0$ and the determination of an optimal trajectory through switching points and waiting times alone may no longer be possible. In the case of 2-3(d), $A_i(t)$ changes sufficiently slowly to maintain the validity of Proposition 2.3 over relatively long time intervals, under the assumption that w.p. 1 no event time coincides with the jump times in any $\{A_i(t)\}$.

In all cases, we initialize the algorithm with $\sigma = 5$ and $\varepsilon = 2 \times 10^{-10}$. The running times of Algorithm 1 are approximately 10 sec using Armijo step-sizes. Note that although the number of iterations for the examples shown may vary substantially, the actual algorithm running times do not. This is simply because the Armijo step-size method may require several trials per iteration to adjust the step-size in order to achieve an adequate decrease in cost. In Fig. 2-3(a),(d), the red line shows the cost as a function of iteration number using a constant step size and the two lines converge to the same approximate optimal value. Non-smoothness in Fig. 2-3(d) comes from the fact that it is a stochastic process. Note that in all cases the initial cost is significantly reduced indicating the importance of optimally selecting the values of the switching points and associated waiting times (if any).
Figure 2.3: One agent example. $L = 20, T = 400$. For each example, top plot: optimal trajectory; bottom plot: $J$ versus iterations.

Figure 2.5 shows two two-agent examples with $L = 40, M = 41$ and evenly spaced sampling points over $[0, L]$, $A_i = 0.01, B = 3, r = 4, R_i(0) = 4$ for all $i$ and $T = 400$. In Fig. 2.5(a) the agents are allowed to move over the whole mission space $[0, L]$, while in Fig. 2.5(b) they are only allowed to move over $[a, b]$ where $a = r$ and $b = L - r$. We initialize the algorithm with the same $\sigma$ and $\varepsilon$ as before. The algorithm running time is approximately 15 sec using Armijo step-sizes, and we observe once again significant reductions in cost.
Figure 2.4: Magnified trajectory for sub-figure (a) and (b) in Fig. 2.3, $t \in [0, 75]$.

(a) $a = 0, b = 20$.  
(b) $a = 4, b = 16$.

Figure 2.5: Two agent example. $L = 40, T = 400$. Top plot: optimal trajectory. Bottom plot: $J$ versus iterations.

(a) $a = 0, b = 20$. $J^* = 17.77$.  

(a) $a = 0, b = 20$.  
(b) $a = 4, b = 16$. 

2.6 Soft constraint case with $R_i \leq R_i^{\text{max}}$

Now we would like to impose hard constraints $R_i(t) \leq R_i^{\text{max}}$, for some $i = 1, \ldots, M$, so as to provide upper bound uncertainty guarantees. Instead of such hard constraints, however, we proceed by imposing soft constraints as an additional cost proportional to the current uncertainty value at $i$, $R_i(t)$, whenever it exceeds the predefined threshold $R_i^{\text{max}}$. Thus, setting $u(t) = [u_1(t), \ldots, u_N(t)]^T$, we aim to solve the following optimal control problem P2.2:

$$\min_{u(t)} J = \frac{1}{T} \int_0^T \sum_{i=1}^M [R_i(t) + \beta_i R_i(t) 1[R_i(t) \geq R_i^{\text{max}} ]] dt$$

where $\beta_i > 0$ for at least some $i \in 1, \ldots, M$.

Following the Hamiltonian analysis presented earlier, the optimal control $u^*_n(t)$ still only depends on the sign of $\lambda_{sn}^*(t)$. It is easy to verify that the proofs of Prop. 2.1, 2.2 and 2.4 still hold in this case, so that, under optimal control, agents never reach end points of the mission space and if $a > 0$ and (or) $b < L$, they may remain at points $a$ and $b$ for finite time interval. However, the proof of Proposition 2.3 needs to be modified so as to accommodate the penalty term added to the cost function; this changes the Hamiltonian expression, as well as the dynamics of $\lambda_j(t)$ in (2.9). We state the modified proposition below and provide a proof.

**Proposition 2.5.** On an optimal trajectory with $\beta_i > 0$ in (2.64) for at least some $i \in 1, \ldots, M$, either $u^*_n(t) = \pm 1$ if $\lambda_{sn}^*(t) \neq 0$, or $u^*_n(t) = 0$ if $\lambda_{sn}^*(t) = 0$ for $t \in [0, T]$, $n = 1, \ldots, N$.

**Proof.** When $\beta_i > 0$ for at least some $i \in \{1, \ldots, M\}$ in (2.64), if we analyze the system operating in an interior arc and omit the constraint (2.2), the Hamiltonian in (2.8) becomes

$$H(x, \lambda, u) = \sum_{i=1}^M R_i(t) [1 + \beta_i 1[R_i(t) \geq R_i^{\text{max}} ]] + \sum_{n=1}^N \lambda_{sn}(t) u_n(t) + \sum_{i=1}^M \lambda_i(t) \dot{R}_i(t)$$

(2.65)
For simplicity, we set \( I_i(R_i(t)) = R_i(t) [1 + \beta_i 1 [R_i(t) > R_i^{\text{max}}]] \) and (2.65) becomes

\[
H(x, \lambda, u) = \sum_{i=1}^{M} I_i(R_i(t)) + \sum_{n=1}^{N} \lambda_{sn}(t) u_n(t) + \sum_{i=1}^{M} \lambda_i(t) \dot{R}_i(t) \tag{2.66}
\]

The costate equations \( \dot{\lambda} = \frac{\partial H}{\partial x} \) give the dynamics of \( \lambda_i(t) \) as

\[
\dot{\lambda}_i(t) = \frac{\partial H}{\partial R_i(t)} = \frac{\partial H}{\partial I_i(t)} \frac{\partial I_i(t)}{\partial R_i(t)} = \frac{\partial I_i(t)}{\partial R_i(t)}, \quad i = 1, \ldots, M \tag{2.67}
\]

When \( \lambda_{sn}^*(t) \neq 0 \), we have shown in (2.12) that \( u_n^*(t) = \pm 1 \), depending on the sign of \( \lambda_{sn}^*(t) \). Thus, it remains to consider the case \( \lambda_{sn}^*(t) = 0 \) for some \( t \in [t_1, t_2] \), where \( 0 \leq t_1 < t_2 \leq T \). Since the state is in a singular arc, \( \lambda_{sn}^*(t) \) does not provide information about \( u_n^*(t) \). On the other hand, the Hamiltonian in (2.66) is not an explicit function of time, therefore, setting \( H(x^*, \lambda^*, u^*) = H^* \), we have \( \frac{dH^*}{dt} = 0 \), which gives

\[
\frac{dH^*}{dt} = \sum_{i=1}^{M} \frac{\partial I_i(t)}{\partial R_i(t)} \dot{R}_i^*(t) + \sum_{n=1}^{N} \dot{\lambda}_{sn}^* (t) u_n^*(t) + \sum_{n=1}^{N} \lambda_{sn}^* (t) \dot{u}_n^*(t) + \sum_{i=1}^{M} \dot{\lambda}_i^* (t) \dot{R}_i^*(t) + \sum_{i=1}^{M} \lambda_i^* (t) \dot{R}_i^*(t) = 0
\]

Define \( S(t) = \{ n | \lambda_{sn}(t) = 0, n = 1, \ldots, N \} \) as the set of indices of agents that are in a singular arc and \( \bar{S}(t) = \{ n | \lambda_{sn}(t) \neq 0, n = 1, \ldots, N \} \) as the set of indices of all other agents. Thus, \( \lambda_{sn}^*(t) = 0, \dot{\lambda}_{sn}^* (t) = 0 \) for \( t \in [t_1, t_2], n \in S(t) \). In addition, agents move with constant full speed, either 1 or 1, so that \( \dot{u}_n^*(t) = 0, n \in \bar{S}(t) \). Then, (2.68) becomes

\[
\frac{dH^*}{dt} = \sum_{i=1}^{M} \frac{\partial I_i(t)}{\partial R_i(t)} \dot{R}_i^*(t) + \sum_{n \in \bar{S}(t)} \dot{\lambda}_{sn}^* (t) u_n^*(t) + \sum_{i=1}^{M} \dot{\lambda}_i^* (t) \dot{R}_i^*(t) = 0
\]

From (2.67), \( \dot{\lambda}_i^* (t) = \frac{\partial H^*}{\partial R_i^*(t)} \), \( i = 1, \ldots, M \), so \( \frac{\partial I_i(t)}{\partial R_i(t)} + \dot{\lambda}_i^* (t) = 0 \), leaving only the last two terms above. Note that \( \dot{\lambda}_{sn}^* (t) = \frac{\partial H^*}{\partial s_{sn}^*(t)} \) and writing \( \dot{R}_i^*(t) = \frac{dR_i^*(t)}{dt} \) we get:

\[
\sum_{n \in \bar{S}(t)} u_n^*(t) \frac{\partial H^*}{\partial s_{sn}^*(t)} + \sum_{i=1, R_i \neq 0}^{M} \lambda_i^* (t) \frac{dR_i^*(t)}{dt} = 0
\]

The remainder of the proof is the same as given in Proposition 2.3 and is included here.
for the sake of completeness. Recall from (2.6) that when $R_i(t) \neq 0$, we have $\dot{R}_i(t) = A_i B \prod_{n=1}^{N} [1 - p_i(s_n(t))]$, so that

$$\frac{\partial H^*}{\partial s_n^*(t)} = B \sum_{i=1}^{M} \lambda^*_i(t) \frac{\partial p_i(s^*_n(t))}{\partial s^*_n(t)} \prod_{d \neq n}^{N} (1 - p_i(s^*_d(t)))$$

(2.69)

$$\frac{d\dot{R}_i^*(t)}{dt} = B \sum_{n=1}^{N} u^*_n(t) \frac{\partial p_i(s^*_n(t))}{\partial s^*_n(t)} \prod_{d \neq n}^{N} (1 - p_i(s^*_d(t)))$$

(2.70)

which results in

$$B \sum_{i=1, R_i \neq 0}^{M} \lambda^*_i(t) \sum_{n \in S(t)} u^*_n(t) \frac{\partial p_i(s^*_n(t))}{\partial s^*_n(t)} \prod_{d \neq n}^{N} (1 - p_i(s^*_d(t)))$$

$$= B \sum_{i=1, R_i \neq 0}^{M} \lambda^*_i(t) \sum_{n \in S(t)} u^*_n(t) \frac{\partial p_i(s^*_n(t))}{\partial s^*_n(t)} \prod_{d \neq n}^{N} (1 - p_i(s^*_d(t))) = 0$$

(2.71)

Note that $\frac{\partial p_i(s^*_n(t))}{\partial s^*_n(t)} = \pm \frac{1}{r_i}$ or 0, depending on the relative position of $s^*_n(t)$ with respect to $\alpha_i$. Moreover, (2.71) is invariant to $M$ or the precise way in which the mission space $[0, L]$ is partitioned, which implies that

$$\lambda^*_i(t) \sum_{n \in S(t)} u^*_n(t) \frac{\partial p_i(s^*_n(t))}{\partial s^*_n(t)} \prod_{d \neq n}^{N} (1 - p_i(s^*_d(t))) = 0$$

for all $i = 1, \ldots, M$, $t \in [t_1, t_2]$. Since $\dot{\lambda}^*_i(t) = 1$, $i = 1, \ldots, M$, it is clear that to satisfy this equality we must have $u^*_n(t) = 0$ for all $t \in [t_1, t_2], n \in S(t)$. In conclusion, in a singular arc with $\lambda^*_n(t) = 0$ for some $n \in \{1, \ldots, N\}$, the optimal control is $u^*_n(t) = 0$.

This proof establishes the fact that on an optimal control trajectory, every agent either moves at full speed or is at rest. Since Propositions 2.1, 2.2, 2.4 and 2.5 hold, the solution of the problem with $\beta_i > 0$ for at most some $i \in \{1, \ldots, M\}$ again reduces to determining the switching points $\theta = [\theta_1, \ldots, \theta_N]^T$ and the waiting time on those switching points.
$w = [w_1, \ldots, w_N]^T$. We obtain $\theta^*$ and $w^*$ minimizing $J(\theta, w)$ through a standard gradient-based optimization method using IPA to evaluate $\nabla J(\theta, w) = \left[ \frac{dJ(\theta, w)}{d\theta} \frac{dJ(\theta, w)}{dw} \right]^T$ in (2.72). Note that this gradient only depends on $\nabla R_i(t) = \left[ \frac{\partial R_i(t)}{\partial \theta} \frac{\partial R_i(t)}{\partial w} \right]^T$ and the event times $\tau_k$, $k = 1, \ldots, K$, observed on a trajectory over $[0, T]$. However, we now have additional events which correspond to (i) the uncertainty value $R_i(t)$ exceeding the threshold value $R_i^\text{max}$ for some $i \in \{1, \ldots, M\}$, and (ii) the uncertainty value $R_i(t)$ dropping below the threshold value $R_i^\text{max}$ for the same $i \in \{1, \ldots, M\}$. Next we show how these events affect the objective function gradient evaluation.

### 2.6.1 Objective Function Gradient Evaluation

Based on our analysis, we are still interested in minimizing the objective function $J(\theta, w)$ in (2.64) with respect to $\theta$ and $w$ and can obtain $\nabla J(\theta, w) = \left[ \frac{dJ(\theta, w)}{d\theta} \frac{dJ(\theta, w)}{dw} \right]^T$ as the gradient of

$$J(\theta, w) = \frac{1}{T} \sum_{i=1}^{M} \sum_{k=0}^{K} \int_{\tau_k(\theta, w)}^{\tau_{k+1}(\theta, w)} \left[ R_i(t) + \beta_i R_i(t) 1[R_i(t) > R_i^\text{max}] \right] dt$$

As illustrated in Fig. 2.6, suppose $\tau_e$ is the time instant when some $R_i(t)$ enters a region violating the constraint $R_i(t) \leq R_i^\text{max}$ and $\tau_l$ is the first time instant the same $R_i(t)$ leaves this region after entering it at time $\tau_e$. To generalize, let $[\tau_{e_j}, \tau_{l_j}]$ be the $j$th such interval,
i.e., \( R_i(\tau_{e_i^j}) \leq R_i^{\text{max}} \), \( R_i(t) > R_i^{\text{max}} \) for all \( t \in (\tau_{e_i^j}, \tau_{e_i^j}^+) \), and \( R_i(\tau_{e_i^j}^+) \leq R_i^{\text{max}} \). Then, define

\[
\Phi_i = \{(e_i^j, t_i^j), \ j = 1, \ldots, \phi_i\}
\]

to be the set of event index pairs with the starting and ending points of such intervals, where \( \phi_i \) is the number of such intervals where \( R_i(t) > R_i^{\text{max}} \) (noting that \( \phi_i \) may be zero), and

\[
\mathcal{E}_i = \{e : e = e_i^j \text{ for some } j = 1, \ldots, \phi_i\}
\]

\[
\mathcal{L}_i = \{l : l = l_i^j \text{ for some } j = 1, \ldots, \phi_i\}
\]

Then, we can rewrite \( J(\theta, w) \) above as

\[
J(\theta, w) = \frac{1}{T} \sum_{i=1}^{M} \left[ \sum_{k=0}^{K} \int_{\tau_k}^{\tau_{k+1}} R_i(t) dt + \sum_{j=1}^{\phi_i} \int_{\tau_{e_i^j}}^{\tau_{e_i^j}^+} \beta_i R_i(t) dt \right]
\]

Taking derivatives with respect to \( \theta \) and \( w \) and observing the cancelation of terms of the form \( R_i(\tau_k) \nabla \tau_k \), we finally get

\[
\nabla J(\theta, w) = 1 \sum_{i=1}^{M} \left[ \sum_{k=0}^{K} \int_{\tau_k}^{\tau_{k+1}} \nabla R_i(t) \left( 1 + \beta_i [R_i(t) > R_i^{\text{max}}] \right) dt + \beta_i R_i^{\text{max}} \left( \sum_{l \in \mathcal{L}_i} \nabla \tau_l \sum_{e \in \mathcal{E}_i} \nabla \tau_e \right) \right]
\]

(2.72)

The evaluation of \( \nabla J(\theta, w) \) therefore depends on \( \nabla R_i(t) \), \( \nabla \tau_e \), \( e \in \mathcal{E}_i \), and \( \nabla \tau_l \), \( l \in \mathcal{L}_i \), for all \( i = 1, \ldots, M \). We have already provided the details for calculating \( \nabla R_i(t) \) in Section 2.3 and note that this gradient is not affected by events at \( \tau_e \) and \( \tau_l \) since all agent dynamics and uncertainty dynamics remain unaffected in applying (2.28). Therefore, we are only left with the additional evaluation of \( \nabla \tau_e \) and \( \nabla \tau_l \). They are both determined by the movement of agents, hence the controllable parameters \( \theta \) and \( w \), so entering and leaving the region defined by \( R_i(t) > R_i^{\text{max}} \) are both endogenous events. The associated guard conditions
involved in (2.29) are both $R_i \max = 0$. Thus, from (2.29) we have,

$$\nabla \tau_k = \frac{\nabla R_i \tau_k}{A_i BP_i s \tau_k }, \text{ for } k = e \text{ or } l \tag{2.73}$$

Compared to the case without any constraints, the evaluation of the gradient of the cost function with respect to $\theta$ and $w$ has one more term: $\beta_i R_i \max (\sum_{i} \nabla \tau_l \sum_{e} \nabla \tau_e)$, which only requires the calculation of $\nabla \tau_k$ using (2.73) for the time instant when some $R_i(t)$ enters or leaves the region defined by $R_i(t) > R_i \max$.

Note that when $\beta \rightarrow \infty$, $\sum_{i=1}^{M} R_i(t)$ becomes negligible and the cost function becomes

$$J = \frac{1}{T} \int_{0}^{T} \sum_{i=1}^{M} \beta R_i(t) 1[R_i(t) \geq R_i \max] dt \tag{2.74}$$

Thus, all $R_i(t)$ become invisible to the cost function unless the constraints $R_i(t) \leq R_i \max$ are violated for some $i$. If the system is stable, i.e., agents have the capacity to keep all $R_i(t)$ low, then all $R_i(t)$ would be kept right below the $R_i \max$ constraints.

### 2.6.2 Numerical Experiments

We present two examples of persistent monitoring problems with and without the $R_i \max$ constraint respectively so as to compare them and evaluate the effect of the constraints. In both cases, we use two agents and parameters $L = 40$, $M = 41$, $a_1 = 0$, $a_M = 40$. Also, the remaining sampling points are evenly spaced over $[0, 40]$. The sensing range is set to $r = 4$, the initial values of the uncertainty are $R_i(0) = 2$ for all $i$, and the time horizon is $T = 200$. We select $B = 6$ and $A_i = 0.2$ for all $i$ except $A_20 = 0.4$, which means that the middle sampling point has uncertainty increasing twice as fast as all other points. We set the threshold constraint $R_i \max = 4$ for $i = 0, 1, ..., 40$. Figure 2·7 is an example with $\beta_i = 0, i = 0, \ldots, 40$, and it is equivalent to the problem without any constraint. Figure 2·8 shows an example with $\beta_i = 10, i = 0, \ldots, 40$.

On the left sides of Fig. 2·7 and Fig. 2·8, the top plot shows the cost $J(\theta^l, w^l)$ as a function of iteration number $l = 1, 2, \ldots$, while the bottom plot shows the optimal agent
trajectories. On the right sides of Fig. 2.7 and Fig. 2.8, nine sampling points are evenly selected and their corresponding $R_i(t)$ is shown. The horizontal red line in each such plot shows $R_i^{\text{max}}$. If we compare the 9 selected $R_i(t)$ between these two examples, the effect of increasing $\beta_i$ from $\beta_i = 0$ to $\beta_i = 10$, $i = 0, \ldots, 40$, is obvious in the sense that $R_{20}$ in Fig. 2.7 increases to unacceptably high values beyond $R_{i}^{\text{max}}$, whereas it is maintained to acceptable levels around $R_{i}^{\text{max}}$ in Fig. 2.8. Comparing the optimal agent trajectories shown in Fig. 2.7 and Fig. 2.8, we can see that, when $\beta_i = 10$ for all $i$, the agents sacrifice coverage at the end points in order to maintain the middle point uncertainty $R_{20}$ low.

In addition, if we apply the optimal agent trajectory with $\beta_i = 0$ to the cost function with $\beta_i = 10$, we obtain the cost $J_{\beta=10}(\theta_{\beta=0}^{*}, w_{\beta=0}^{*}) = 1861$, which is shown as the horizontal red line in the left plot of Fig. 2.8, while the actual optimal cost with $\beta_i = 10$, $i = 0, \ldots, 40$, is $J_{\beta=10}(\theta_{\beta=10}^{*}, w_{\beta=10}^{*}) = 530.1$. This is a significant cost reduction resulting from our algorithm which accounts for the desired performance constraints.

2.7 Summary

We have formulated an optimal persistent monitoring problem with the objective of controlling the movement of multiple cooperating agents to minimize an uncertainty metric in
Figure 2.8: Two agent example for $\beta_i = 10$, $i = 0, 1, \ldots, 40$. Left top plot: cost $J(\theta^i, w^i)$ as a function of iteration number with $J_{\beta=10}(\theta^i_{\beta=10}, w^i_{\beta=10}) = 530.1$, red line $J_{\beta=10}(\theta^i_{\beta=0}, w^i_{\beta=0}) = 1861$ represents the cost using the agent trajectories with $\beta_i = 0$ and cost function with $\beta_i = 10$, $i = 0, \ldots, 40$; left bottom plot shows the optimal agent trajectories. Right plot: Selected $R_i(t)$ vs. time. Red lines are the $R_{\text{max}}^i$ constraint.

a given mission space with and without some upper bound constraints for the uncertainty values in the mission space. In a 1-dimensional mission space, we have shown that the optimal solution is reduced to the determination of two parameter vectors for each agent: a sequence of switching locations and associated waiting times at these switching points. We have used Infinitesimal Perturbation Analysis (IPA) to evaluate sensitivities of the objective function with respect to all parameters and, therefore, obtain a complete on-line (locally optimal) solution through a gradient-based algorithm. Our work has also established the basis for extending this optimal control approach to a 2-dimensional mission space. In this case, similar simple solutions can no longer be derived. An alternative is to optimally assign each agent a linear trajectory, motivated by the 1-dimensional analysis. We prove, however, that elliptical trajectories outperform linear ones. Therefore, by perimetrically describing an elliptical trajectory, we formulate and solve instead a parametric optimization problem in which we seek to determine such trajectories optimizing an uncertainty metric over a 2-dimensional mission space.
Chapter 3

Elliptical Trajectories for 2-dimensional Persistent Monitoring Problem

In the Chapter 2, we addressed the persistent monitoring problem by proposing an optimal control framework to drive multiple cooperating agents so as to minimize a metric of uncertainty over the 1-dimensional environment. This metric is a function of both space and time such that uncertainty at a point grows if it is not covered by any agent sensors. To model sensor coverage, we define a probability of detecting events at each point of the mission space by agent sensors. Thus, the uncertainty of the environment decreases with a rate proportional to the event detection probability, i.e., the higher the sensing effectiveness is, the faster the uncertainty is reduced. It is shown in that the optimal control problem can be reduced to a parametric optimization problem. In particular, the optimal trajectory of each agent is to move at full speed until it reaches some switching point, dwell on the switching point for some time (possibly zero), and then switch directions. Thus, each agent's optimal trajectory is fully described by a set of switching points and associated waiting times at these points. This allows us to make use of IPA to determine gradients of the objective function with respect to these parameters and subsequently obtain optimal switching locations and waiting times that fully characterize an optimal solution. It also allows us to exploit robustness properties of IPA to readily extend this solution approach to a stochastic uncertainty model.

In this chapter, we address the same persistent monitoring problem in a 2-dimensional (2D) mission space. Using an analysis similar to the 1-dimensional (1D) case, we find that we can no longer identify a parametric representation of optimal agent trajectories. A complete solution requires a computationally intensive process for solving a Two Point
Boundary Value Problem (TPBVP) making any on-line solution to the problem infeasible. Motivated by the simple structure of the 1D problem, it has been suggested to assign each agent a linear trajectory for which the explicit 1D solution can be used. One could then reduce the problem to optimally carrying out this assignment. However, in a 2D space it is not obvious that a linear trajectory is a desirable choice. Indeed, a key contribution of this paper is to formally prove that an elliptical agent trajectory outperforms a linear one in terms of the uncertainty metric we are using. Motivated by this result, we formulate a 2D persistent monitoring problem as one of determining optimal elliptical trajectories for a given number of agents, noting that this includes the possibility that one or more agents share the same trajectory. We show that this problem can be explicitly solved using similar IPA techniques as in our 1D analysis. In particular, we use IPA to determine on line the gradient of the objective function with respect to the parameters that fully define each elliptical trajectory (center, orientation and length of the minor and major axes). This approach is scalable in the number of observed events, not states, of the underlying hybrid system characterizing the persistent monitoring process, so that it is suitable for on­line implementation. However, the standard gradient-based optimization process we use is generally limited to local, rather than global optimal solutions. Thus, we adopt a stochastic comparison algorithm from the literature (Bao and Cassandras, 1996) to overcome this problem.

The remainder of the chapter is organized as follows. Section 3.1 formulates the optimal control framework for the 2-dimensional mission space and Section 3.2 presents the solution approach using Hamiltonian analysis. In Section 3.3 we establish our key result that elliptical agent trajectories outperform linear ones in terms of minimizing an uncertainty metric per unit area. In Section 3.4 we formulate and solve the problem of determining optimal elliptical agent trajectories using an algorithm driven by gradients evaluated through IPA. In Section 3.5, we incorporate a stochastic comparison algorithm for obtaining globally optimal solutions and in Section 3.6 we provide numerical results to illustrate our approach and compare it to computationally intensive solutions based on a TPBVP solver. Section
3.1 2D Persistent Monitoring Problem Formulation

We consider $N$ mobile agents in a 2D rectangular mission space $\Omega \equiv [0, L_1] \times [0, L_2] \subset \mathbb{R}^2$. Let the position of the agents at time $t$ be $s_n(t) = [s^x_n(t), s^y_n(t)]$ with $s^x_n(t) \in [0, L_1]$ and $s^y_n(t) \in [0, L_2]$, $n = 1, \ldots, N$, following the dynamics:

\[ \begin{align*}
    \dot{s}^x_n(t) &= u_n(t) \cos \theta_n(t), \\
    \dot{s}^y_n(t) &= u_n(t) \sin \theta_n(t)
\end{align*} \tag{3.1} \]

where $u_n(t)$ is the scalar speed of the $n$th agent and $\theta_n(t)$ is the angle relative to the positive direction that satisfies $0 \leq \theta_n(t) < 2\pi$. Thus, we assume that each agent controls its orientation and speed. Without loss of generality, after some rescaling of the size of the mission space, we further assume that the speed is constrained by $0 \leq u_n(t) \leq 1$, $n = 1, \ldots, N$. Each agent is represented as a particle in the 2D space, thus we ignore the case of two or more agents colliding with each other.

We associate with every point $[x, y] \in \Omega$ a function $p_n(x, y, s_n)$ that measures the probability that an event at location $[x, y]$ is detected by agent $n$. We also assume that $p_n(x, y, s_n) = 1$ if $[x, y] = s_n$, and that $p_n(x, y, s_n)$ is monotonically nonincreasing in the Euclidean distance $D(x, y, s_n) \equiv ||[x, y] - s_n||$ between $[x, y]$ and $s_n$, thus capturing the reduced effectiveness of a sensor over its range which we consider to be finite and denoted by $r_n$ (this is the same as the concept of sensor footprint" commonly used in the robotics literature.) Therefore, we set $p_n(x, y, s_n) = 0$ when $D(x, y, s_n) > r_n$. Our analysis is not affected by the precise sensing model $p_n(x, y, s_n)$, but we mention here as an example the linear decay model used in (Cassandras et al., 2013):

\[ p_n(x, y, s_n) = \begin{cases} 
    \frac{1}{C}(1 - \frac{D(x, y, s_n)}{r_n}), & \text{if } D(x, y, s_n) \leq r_n \\
    0, & \text{if } D(x, y, s_n) > r_n
\end{cases} \tag{3.2} \]

where $C$ is a normalization constant. Next, consider a set of points $\{[\alpha_i, \beta_i], i = 1, \ldots, M\}$, $[\alpha_i, \beta_i] \in \Omega$, and associate a time-varying measure of uncertainty with each point $[\alpha_i, \beta_i]$, which we denote by $R_i(t)$. The set of points $\{[\alpha_1, \beta_1], \ldots, [\alpha_M, \beta_M]\}$ may be selected to
contain specific points of interest" in the environment, or simply to sample points in the mission space. Alternatively, we may consider a partition of $\Omega$ into $M$ rectangles denoted by $\Omega_i$ whose center points are $[\alpha_i, \beta_i]$. We can then set $p_n(x, y, s_n(t)) = p_n(\alpha_i, \beta_i, s_n(t))$ for all $\{(x, y) | [x, y] \in \Omega_i, [\alpha_i, \beta_i] \in \Omega_i\}$, i.e., for all $[x, y]$ in the rectangle $\Omega_i$ that $[\alpha_i, \beta_i]$ belongs to. In order to avoid the uninteresting case where there is a large number of agents who can adequately cover the mission space, we assume that for any $s(t)$, there exists some point $[x, y] \in \Omega$ with $P(x, y, s(t)) = 0$. This means that for any assignment of $N$ agents at time $t$, there is always at least one point in the mission space that cannot be sensed by any agent. Therefore, the joint probability of detecting an event at location $[\alpha_i, \beta_i]$ by all the $N$ agents (assuming detection independence) is

$$P_i(s(t)) = 1 - \prod_{n=1}^{N} [1 - p_n(\alpha_i, \beta_i, s_n(t))]$$

where we set $s(t) = [s_1(t), \ldots, s_N(t)]^T$. Similar to the 1D analysis in (Cassandras et al., 2013), we define uncertainty functions $R_i(t)$ associated with the rectangles $\Omega_i$, $i = 1, \ldots, M$, so that they have the following properties: (i) $R_i(t)$ increases with a prespecified rate $A_i$ if $P_i(s(t)) = 0$, (ii) $R_i(t)$ decreases with a fixed rate $B$ if $P_i(s(t)) = 1$ and (iii) $R_i(t) \geq 0$ for all $t$. It is then natural to model uncertainty so that its decrease is proportional to the probability of detection. In particular, we model the dynamics of $R_i(t)$, $i = 1, \ldots, M$, as follows:

$$\dot{R}_i(t) = \begin{cases} 0 & \text{if } R_i(t) = 0, \quad A_i \leq BP_i(s(t)) \\ \dot{R}_i(t) \leq \frac{B}{A_i} P_i(s(t)) & \text{otherwise} \end{cases} \quad (3.3)$$

where we assume that initial conditions $R_i(0)$, $i = 1, \ldots, M$, are given and that $B > A_i > 0$ for all $i = 1, \ldots, M$; thus, the uncertainty strictly decreases when there is perfect sensing $P_i(s(t)) = 1$.

The goal of the optimal persistent monitoring problem we consider is to control through $u_n(t), \theta_n(t)$ in (3.1) the movement of the $N$ agents so that the cumulative uncertainty over all sensing points $\{[\alpha_1, \beta_1], \ldots, [\alpha_M, \beta_M]\}$ is minimized over a fixed time horizon $T$. Thus, setting $u(t) = [u_1(t), \ldots, u_N(t)]$ and $\theta(t) = [\theta_1(t), \ldots, \theta_N(t)]$ we aim to solve the
following optimal control problem P3.1:

\[
P3.1 : \min_{u(t), \theta(t)} J = \int_0^T \sum_{i=1}^M R_i(t) dt \tag{3.4}
\]

subject to the agent dynamics (3.1), uncertainty dynamics (3.3), control constraint \(0 \leq u_n(t) \leq 1, 0 \leq \theta_n(t) \leq 2\pi, t \in [0, T]\), and state constraints \(s_n(t) \in \Omega\) for all \(t \in [0, T]\), \(n = 1, \ldots, N\).

**Remark 3.1.** The modeling of the uncertainty value \(R_i(t)\) in a 2D environment is a direct extension of (Cassandras et al., 2013) in the 1D environment setting in Chapter 2 where it was described how persistent monitoring can be viewed as a polling system, with each rectangle \(\Omega_i\) associated with a virtual queue" where uncertainty accumulates with inflow rate \(A_i\). Each agent acts as a server" visiting these virtual queues with a time-varying service rate given by \(BP_i(s(t))\), controllable through all agent positions at time \(t\). Metrics other than (3.4) are of course possible, e.g., maximizing the mutual information or minimizing the spectral radius of the error covariance matrix (Zhang et al., 2010) if specific point of interest" are identified with known properties.

### 3.2 Optimal Control Solution

We first characterize the optimal control solution of problem P3.1. We define the state vector \(x(t) = [s_1^x(t), s_1^y(t), \ldots, s_N^x(t), s_N^y(t), R_1(t), \ldots, R_M(t)]^T\) and the associated costate vector \(\lambda(t) = [\mu_1^x(t), \mu_1^y(t), \ldots, \mu_N^x(t), \mu_N^y(t), \lambda_1(t), \ldots, \lambda_M(t)]^T\). In view of the discontinuity in the dynamics of \(R_i(t)\) in (3.3), the optimal state trajectory may contain a boundary arc when \(R_i(t) = 0\) for any \(i\); otherwise, the state evolves in an interior arc (Bryson and Ho, 1975). This follows from the fact, proved in (Cassandras et al., 2013) and (Lin and Cassandras, 2013) that it is never optimal for agents to reach the mission space boundary. We analyze the system operating in such an interior arc and omit the state constraint \(s_n(t) \in \Omega, n = 1, \ldots, N, t \in [0, T]\). Using (3.1) and (3.3), the Hamiltonian is

\[
H = \sum_i R_i(t) + \sum_i \lambda_i \dot{R}_i(t) + \sum_n \mu_n^x(t) u_n(t) \cos \theta_n(t) + \sum_n \mu_n^y(t) u_n(t) \sin \theta_n(t) \tag{3.5}
\]
and the costate equations $\dot{\lambda}_i = \frac{\partial H}{\partial x}$ are

$$
\dot{\lambda}_i(t) = \frac{\partial H}{\partial R_i} = 1 \quad (3.6)
$$

$$
\dot{\mu}_n^x(t) = \frac{\partial H}{\partial s_n^x} = \sum_i \frac{\partial}{\partial s_n^x} \lambda_i \dot{R}_i(t) \\
= \sum_{[\alpha_i, \beta_i] \in \mathcal{R}(s_n)} \frac{B \lambda_i(s_n^{\alpha_i})}{r_n D(\alpha_i, \beta_i, s_n(t))} \prod_{d \neq n} [1 \ p_d(\omega_i, s_d(t))] \quad (3.7)
$$

$$
\dot{\mu}_n^y(t) = \frac{\partial H}{\partial s_n^y} = \sum_i \frac{\partial}{\partial s_n^y} \lambda_i \dot{R}_i(t) \\
= \sum_{[\alpha_i, \beta_i] \in \mathcal{R}(s_n)} \frac{B \lambda_i(s_n^{\beta_i})}{r_n D(\alpha_i, \beta_i, s_n(t))} \prod_{d \neq n} [1 \ p_d(\omega_i, s_d(t))] \quad (3.8)
$$

where $\mathcal{R}(s_n) \equiv \{[\alpha_i, \beta_i] | D(\alpha_i, \beta_i, s_n) \leq r_n, R_i > 0 \}$ identifies all points $[\alpha_i, \beta_i]$ within the sensing range of the agent using the model in (3.2). Since we impose no terminal state constraints, the boundary conditions are $\lambda_i(T) = 0, i = 1, \ldots, M$ and $\mu_n^x(T) = 0, \mu_n^y(T) = 0, n = 1, \ldots, N$. The implication of (3.6) with $\lambda_i(T) = 0$ is that $\lambda_i(t) = T-t$ for all $t \in [0, T], i = 1, \ldots, M$ and that $\lambda_i(t)$ is monotonically decreasing starting with $\lambda_i(0) = T$. However, this is only true if the entire optimal trajectory is an interior arc, i.e., all $R_i(t) \geq 0$ constraints for all $i = 1, \ldots, M$ remain inactive. We have shown in (Cassandras et al., 2013) that $\lambda_i(t) \geq 0, i = 1, \ldots, M, t \in [0, T]$ with equality holding only if $t = T$, or $t = t_0$ with $R_i(t_0) = 0, R_i(t') > 0$, where $t' \in [t_0 \quad \delta, t_0], \delta > 0$. Although this argument holds for the 1D problem formulation, the proof can be directly extended to this 2D environment. However, the actual evaluation of the full costate vector over the interval $[0, T]$ requires solving (3.7) and (3.8), which in turn involves the determination of all points where the state variables $R_i(t)$ reach their minimum feasible values $R_i(t) = 0, i = 1, \ldots, M$. This generally involves the solution of a TPBVP.
From (3.5), after some algebraic operations, we get

\[
H = \sum_i R_i(t) + \sum_i \lambda_i \dot{R}_i(t) + \sum_n u_n(t) \left[ \mu_n^x(t) \cos \theta_n(t) + \mu_n^y(t) \sin \theta_n(t) \right]
\]

\[
= \sum_i R_i(t) + \sum_i \lambda_i \dot{R}_i(t) + \sum_n \text{sgn}(\mu_n^y(t)) \sqrt{\left(\mu_n^x(t)\right)^2 + \left(\mu_n^y(t)\right)^2}
\times u_n(t) \left[ \frac{\text{sgn}(\mu_n^y(t)) \mu_n^x(t) \cos \theta_n(t) + |\mu_n^y(t)| \sin \theta_n(t)}{\sqrt{\left(\mu_n^x(t)\right)^2 + \left(\mu_n^y(t)\right)^2}} \right]
\]

(3.9)

where \text{sgn}(\cdot) is the sign function. Combining the trigonometric function terms, we obtain

\[
H = \sum_i R_i(t) + \sum_i \lambda_i \dot{R}_i(t) + \sum_n \text{sgn}(\mu_n^y(t)) u_n(t) \sqrt{\left(\mu_n^x(t)\right)^2 + \left(\mu_n^y(t)\right)^2} \sin(\theta_n(t) + \psi_n(t))
\]

(3.10)

and \psi_n(t) is defined so that \tan \psi_n(t) = \frac{\mu_n^y(t)}{\mu_n^x(t)} for \mu_n^y(t) \neq 0 and

\[
\psi_n(t) = \begin{cases} 
\frac{\pi}{2}, & \text{if } \mu_n^x(t) < 0 \\
\frac{\pi}{2}, & \text{if } \mu_n^x(t) > 0
\end{cases}
\]

for \mu_n^y(t) = 0. In what follows, we exclude the case where \mu_n^x(t) = 0 and \mu_n^y(t) = 0 at the same time for any given \( n \) over any finite singular interval. Applying the Pontryagin minimum principle to (3.10) with \( u_n^*(t), \theta_n^*(t), t \in [0, T) \), denoting optimal controls, we have

\[
H(x^*, \lambda^*, u^*, \theta^*) = \min_{u \in [0,1]^N, \lambda \in [0,2\pi]^N} H(x, \lambda, u, \theta)
\]

and it is immediately obvious that it is necessary for an optimal control to satisfy:

\[
u_n^*(t) = 1
\]

(3.11)

and

\[
\begin{cases} 
\sin (\theta_n^*(t) + \psi_n(t)) = 1, & \text{if } \mu_n^y(t) < 0 \\
\sin (\theta_n^*(t) + \psi_n(t)) = -1, & \text{if } \mu_n^y(t) > 0
\end{cases}
\]

(3.12)

Note \( u_n(t) = 0 \) is not an optimal solution, since we can always set control \( \theta_n(t) \) to enforce \text{sgn}(\mu_n^y(t)) \sin(\theta_n(t) + \psi_n(t)) < 0. \) Thus, we have

\[
\begin{cases} 
\theta_n^*(t) = \frac{\pi}{2}, & \psi_n(t), \text{ if } \mu_n^y(t) < 0 \\
\theta_n^*(t) = \frac{3\pi}{2}, & \psi_n(t), \text{ if } \mu_n^y(t) > 0
\end{cases}
\]

(3.13)
Clearly, when $\mu^y_n(t) < 0$, the $n$th agent heading is $\theta_n^*(t) = \frac{1}{2}\pi$, $\psi_n(t) \in (0, \pi)$ and the agent will move upward in $\Omega$; similarly, when $\mu^y_n(t) > 0$ the agent will move downward. When $\mu^y_n(t) = 0$, we have

$$
\psi_n(t) = \begin{cases} 
\frac{\pi}{2}, & \text{if } \mu_n^x(t) < 0 \\
\frac{\pi}{2}, & \text{if } \mu_n^x(t) > 0
\end{cases}
$$

$$
\theta_n^*(t) = \begin{cases} 
0, & \text{if } \mu_n^x(t) < 0 \\
\pi, & \text{if } \mu_n^x(t) > 0
\end{cases}
$$

so that the $n$th agent will move horizontally. By symmetry, the agent will move towards the right when $\mu_n^x(t) < 0$, towards the left when $\mu_n^x(t) > 0$, and vertically when $\mu_n^x(t) = 0$. Note that this is analogous to the 1D problem in (Cassandras et al., 2013) where the costate $\lambda_{sn}(t) < 0$ implies $u_n(t) = 1$ and $\lambda_{sn}(t) > 0$ implies $u_n(t) = 1$.

Returning to the Hamiltonian in (3.5), the optimal heading $\theta_n^*(t)$ can be obtained by requiring $\frac{\partial H}{\partial \theta_n^*} = 0$:

$$
\frac{\partial H}{\partial \theta_n^*} = \mu_n^x(t) u_n(t) \sin \theta_n(t) + \mu_n^y(t) u_n(t) \cos \theta_n(t) = 0
$$

which gives:

$$
\tan \theta_n^*(t) = \frac{\mu_n^y(t)}{\mu_n^x(t)}
$$

Applying the tangent operation to both sides of (3.13), we can see that (3.13) and (3.15) are equivalent to each other.

Since we have shown that $u_n^*(t) = 1, n = 1, \ldots, N$ in (3.11), we are only left with the task of determining $\theta_n^*(t), n = 1, \ldots, N$. This can be accomplished by solving a standard TPBVP involving forward and backward integrations of the state and costate equations to evaluate $\frac{\partial H}{\partial \theta_n^*}$ after each such iteration and using a gradient descent approach until the objective function converges to a (local) minimum. Clearly, this is a computationally intensive process which scales poorly with the number of agents and the size of the mission space. In addition, it requires discretizing the mission time $T$ and calculating every control at each time step which adds to the computational complexity.

First we discretize the operation time $T$ into $Z$ time slots, assign a series of initial control $\theta = [\theta^1, \theta^2, \ldots, \theta^Z]$ randomly for each time slot, where $\theta^m = [\theta^m_1, \ldots, \theta^m_N], m =$
We then apply \( \theta \) to obtain the system states \( R_{ij}(t), s_x(t), s_y(t) \) using the system dynamics (3.1) and (3.3). After having all the states, we can calculate all the costates \( \lambda_{ij}(t), \lambda_x(t), \lambda_y(t) \) backward from time \( T \) to 0 using the costate dynamics (3.6)-(3.8) and the costate termination conditions. Substitute all the control and costate values into (3.14), we have \( \frac{\partial H}{\partial \theta} \) for each time slot. Then we can use \( \frac{\partial H}{\partial \theta} \) to update \( \theta \) using the gradient descent approach to achieve optimal control solution. Details of the algorithm are in Alg. 2.

**Algorithm 2 :** Two-Point-Boundary-Value Problem Algorithm for 2-dimensional Persistent Monitoring Problem

1: Pick \( \epsilon > 0 \).
2: repeat
3: Discretize the operation time \( T \) into \( Z \) time slots. Assign an initial control \( \theta = [\theta^1, \theta^2, ..., \theta^Z] \) to the system and each component \( \theta^i \) of the control vector is the control we apply to the \( i^{th} \) time slot.
4: Apply the control \( \theta \) to obtain the states \( R_i(t), s_x(t), s_y(t) \) using the dynamics (3.1) and (3.3) for each time slot in the forward direction from time 0 to \( T \).
5: Obtain the costates \( \lambda_i(t), \lambda_x(t), \lambda_y(t) \) using the dynamics (3.6)-(3.8) and the terminal condition \( \lambda_i(T) = 0, i = 1, ..., M \) and \( \lambda_x(T) = 0, \lambda_y(T) = 0 \) for each time slot in the backward direction from time \( T \) to 0.
6: For \( l = 1 : Z \), substitute the corresponding \( \theta^l \) and \( \lambda^l \) into (3.14) to acquire the partial derivative of \( H \) with respect to \( \theta \) in the \( l^{th} \) time slot \( \frac{\partial H^l}{\partial \theta} \).
7: For \( l = 1 : Z \), update \( \theta^l = \theta^l - \eta \frac{\partial H^l}{\partial \theta} \).
8: until \( \max_l|\frac{\partial H^l}{\partial \theta}| < \epsilon \)
9: END

Here we present a numerical example of 2-dimensional persistent monitoring problems in which agent trajectories are determined using Algorithm 2 for two agent case. These two examples are two-agent experiment with \( L_1 = 19, L_2 = 19, M = 20, N = 20 \) and the remaining sampling points are evenly distributed over the rectangular space. The sensing range is set to \( r = 4 \), the initial values of the uncertainty functions are \( R_{i,j}(0) = 4 \), for \( i = 1, ..., M, j = 1, ..., N \) and the time horizon is \( T = 400 \). In Fig. (3·1) we show results where the two agents start with different initial positions \([0,0] \) and \([0,19] \). The left plot shows the initial trajectories we assigned to the system. For simplicity, we let them to move diagonally cross each other. We use blue and green lines to represent each agent's trajectories respectively. The right plot shows the corresponding optimal trajectories for
the two agents, using the left plot trajectories as the initial input. The final uncertainty value of sampling points are shown by red squares in the space. The darker the color, the higher the uncertainty value is. Final agent positions are represented by black dots in the rectangular space.

### 3.3 Linear vs Elliptical Agent Trajectories

Given the complexity of the TPBVP required to obtain an optimal solution of problem P3.1, we seek alternative approaches which may be suboptimal but are tractable and scalable. The first such effort is motivated by the results obtained in our 1D analysis, where we found that on a mission space defined by a line segment $[0, L]$ the optimal trajectory for each agent is to move at full speed until it reaches some switching point, dwell on the switching point for some time (possibly zero), and then switch directions. Thus, each agent’s optimal trajectory is fully described by a set of switching points $\{\theta_1, \ldots, \theta_K\}$ and associated waiting times at these points, $\{w_1, \ldots, w_K\}$. The values of these parameters can then be efficiently determined using a gradient-based algorithm; in particular, we used Infinitesimal Perturbation Analysis (IPA) to evaluate the objective function gradient as
shown in (Cassandras et al., 2013).

Thus, a reasonable approach that has been suggested is to assign each agent a linear trajectory. The 2D persistent monitoring problem would then be formulated as consisting of the following tasks: (i) finding \( N \) linear trajectories in terms of their length and exact location in \( \Omega \), noting that one or more agents may share one of these trajectories, and (ii) controlling the motion of each agent on its trajectory. Task (ii) is a direct application of the 1D persistent monitoring problem solution, leaving only task (i) to be addressed. However, there is no reason to believe that a linear trajectory is a good choice in a 2D setting. A broader choice is provided by the set of elliptical trajectories which in fact encompass linear ones when the minor axis of the ellipse becomes zero. Thus, we first proceed with a comparison of these two types of trajectories. The main result of this section is to formally show that an elliptical trajectory outperforms a linear one using the average uncertainty metric in (3.4) as the basis for such comparison.

To simplify notation, let \( \omega = [x, y] \in \mathbb{R}^2 \) and, for a single agent, define

\[
\Xi = \{ \omega \in \mathbb{R}^2 | \exists t \in [0, T] \text{ such that } Bp(\omega, s(t)) > A(\omega) \} \tag{3.16}
\]

Note that \( \Xi \) above defines the effective coverage region for the agent, i.e., the region where the uncertainty corresponding to \( R(\omega, t) \) with the dynamics in (3.3) can be strictly reduced given the sensing capacity of the agent determined through \( B \) and \( p(\omega, s) \). Clearly, \( \Xi \) depends on the values of \( s(t) \) which are dependent on the agent trajectory. Let us define an elliptical trajectory so that the agent position \( s(t) = [s^x(t), s^y(t)] \) follows the general parametric form of an ellipse:

\[
\begin{cases}
  s^x(t) = X + a \cos \rho(t) \cos \varphi - b \sin \rho(t) \sin \varphi \\
  s^y(t) = Y + a \cos \rho(t) \sin \varphi + b \sin \rho(t) \cos \varphi
\end{cases} \tag{3.17}
\]

where \([X, Y]\) is the center of the ellipse, \( a, b \) are its major and minor axis respectively, \( \varphi \in [0, \pi) \) is the ellipse orientation (the angle between the x axis and the major ellipse axis) and \( \rho(t) \in [0, 2\pi) \) is the eccentric anomaly of the ellipse. Assuming the agent moves with constant maximal speed \( 1 \) on this trajectory (based on (3.11)), we have \((s^x)^2 + (s^y)^2 = 1\),
Figure 3.2: The red ellipse represents the agent trajectory. The area defined by the black curve is the agent’s effective coverage area. \( \frac{ab}{\sqrt{b^2\cos^2(\theta) + a^2\sin^2(\theta)}} + \gamma(\theta) \) is the distance between the ellipse center and the coverage area boundary for a given \( \theta \).

which gives

\[
\rho(t) = \left[ (a \sin \rho(t) \cos \varphi + b \cos \rho(t) \sin \varphi)^2 + (a \sin \rho(t) \sin \varphi - b \cos \rho(t) \cos \varphi)^2 \right]^{1/2}
\]

In order to make a fair comparison between a linear and an elliptical trajectory, we normalize the objective function in (3.4) with respect to the coverage area in (3.16) and consider all points in \( \Xi \) (rather than discretizing it or limiting ourselves to a finite set of sampling points). Thus, we define:

\[
J(b) = \frac{1}{\Psi_\Xi} \int_0^T \int_{\Xi} R(\omega, t) \, d\omega dt
\]  

(3.18)

where \( \Psi_\Xi = \int_{\Xi} d\omega \) is the area of the effective coverage region. Note that we view this normalized metric as a function of \( b \geq 0 \), so that when \( b = 0 \) we obtain the uncertainty corresponding to a linear trajectory. For simplicity, the trajectory is selected so that \([X, Y]\) coincides with the origin and \( \varphi = 0 \), as illustrated in Fig. 3.2 with the major axis \( a \) assumed fixed. Regarding the range of \( b \), we will only be interested in values which are limited to a neighborhood of zero that we will denote by \( B \). Given \( a \), this set dictates the values that \( s(t) \in \Xi \) is allowed to take. Finally, we make the following assumptions:

**Assumption 1:** \( p(\omega, s) \equiv p(D(\omega, s)) \) is a continuous function of \( D(\omega, s) \equiv ||\omega - s|| \).

**Assumption 2:** Let \( \omega, \omega' \) be symmetric points in \( \Xi \) with respect to the center point of the ellipse. Then, \( A(\omega) = A(\omega') \).

The first assumption simply requires that the sensing range of an agent is continuous
and the second that all points in $\Xi$ are treated uniformly (as far as how uncertainty is measured) with respect to an elliptical trajectory centered in this region. The following result establishes the fact that an elliptical trajectory with some $b > 0$ can achieve a lower cost than a linear trajectory (i.e., $b = 0$) in terms of a long-term average uncertainty per unit area.

**Proposition 3.1.** Under Assumptions 1-2 and $b \in \mathcal{B}$,

$$\lim_{T \to \infty, b \to 0} \frac{\partial J(b)}{\partial b} < 0$$

i.e., switching from a linear to an elliptical trajectory reduces the cost in (3.18).

**Proof.** Since a linear trajectory is the limit of an elliptical one (with the major axis kept fixed) as the minor axis reaches $b = 0$, our proof is based on perturbing the minor axis $b$ away from 0 and showing that we can then achieve a lower average cost $J$ in (3.18), as long as this is measured over a sufficiently long time interval.

Obviously, the effective coverage area $\Psi_\Xi$ depends on the agent’s trajectory and, in particular, on the minor axis length $b$. From the definition of $\Xi$ in (3.16), note that $\Psi_\Xi$ monotonically increases in $b \in \mathcal{B}$, i.e., $\frac{\partial \Psi_\Xi}{\partial b} > 0$ and it immediately follows that:

$$\frac{\partial}{\partial b} \left( \frac{1}{\Psi_\Xi} \right) = \frac{\partial \Psi_\Xi}{\partial b} \frac{1}{\Psi_\Xi^2} < 0 \quad \text{(3.19)}$$

We now rewrite the area integral in (3.18) in a polar coordinate system with $\omega = (\xi, \vartheta) \in \mathbb{R}^2$, where $\xi$ is the polar radius and $\vartheta$ is the polar angle:

$$J(b) = \frac{1}{\Psi_\Xi} \int_0^T \int_0^{2\pi} \int_0^b R(\xi, \vartheta, t) \xi d\xi d\vartheta dt \quad \text{(3.20)}$$

where

$$G(a, b, \vartheta) = \frac{ab}{\sqrt{b^2 \cos^2(\vartheta) + a^2 \sin^2(\vartheta)}} \quad \text{(3.21)}$$

is the ellipse equation in the polar coordinate system and $\gamma(\vartheta)$ is defined for any $(\xi, \vartheta) \in \mathbb{R}^2$ as

$$\gamma(\vartheta) = \sup_{\xi} \{ Bp(\xi, \vartheta, s(t)) > A(\xi, \vartheta) \} \quad G(a, b, \vartheta) \quad \text{(3.22)}$$
where \( \sup_{\xi} \{ Bp(\xi, \vartheta, s(t)) > A(\xi, \vartheta) \} \) is the distance between the ellipse center and the furthest point \((\xi, \vartheta)\), for any given \( \vartheta \), that can be effectively covered by the agent on the ellipse. Taking partial derivatives in (3.20) with respect to \( b \), we get

\[
\frac{\partial J}{\partial b} = \frac{\partial \Psi}{\partial b} \frac{1}{\Psi^2} \int_{0}^{T} \int_{0}^{2\pi} R(\omega, t) d\omega dt + \frac{1}{\Psi} \int_{0}^{T} \int_{0}^{2\pi} [R(G(a, b, \vartheta) + \gamma(\vartheta), \vartheta, t) \cdot (G(a, b, \vartheta) + \gamma(\vartheta)) : \frac{\partial G(a, b, \vartheta)}{\partial b}]
\]

\[
+ \int_{0}^{T} \int_{0}^{2\pi} \frac{\partial R(\xi, \vartheta, t)}{\partial b} \xi d\xi d\vartheta dt \tag{3.23}
\]

Recall that our objective is to show that when we perturb a linear trajectory into an elliptical one, which is achieved by increasing \( b \) from 0 to some small \( b \leq 0 \), we can achieve a lower cost. Thus, we aim to show \( \frac{\partial J}{\partial b} |_{b \to 0} < 0 \). From (3.19), the first term of (3.23) is negative, therefore, we only need to show the second term is non-positive when \( b \to 0 \). By the definition (3.21), observe that when \( b \to 0 \), \( G(a, b, \vartheta) \to 0 \), and \( \frac{\partial G(a, b, \vartheta)}{\partial b} |_{b \to 0} = \frac{1}{\sin \vartheta} \), for \( \vartheta \neq 0 \) and \( \pi \); \( \frac{\partial G(a, b, \vartheta)}{\partial b} |_{b \to 0} = a \) for \( \vartheta = 0 \) or \( \pi \). Thus, the double integral of the second term of (3.23) becomes

\[
\int_{0}^{T} \int_{0}^{2\pi} \left[ \frac{\gamma(\vartheta)}{\sin \vartheta} R(\gamma(\vartheta), \vartheta, t) + \frac{\gamma(\vartheta)}{\sin \vartheta} \frac{\partial R(\xi, \vartheta, t)}{\partial b} \xi d\xi \right] d\vartheta dt \tag{3.24}
\]

By Assumption 2, \( A(\omega) = A(\omega') \), where \( \omega \) and \( \omega' \) are symmetric with respect to the center point of the ellipse, thus \( A(\xi, \vartheta) = A(\xi, \vartheta + \pi) \). Then, for any uncertainty value \( R(\gamma(\vartheta), \vartheta, t) \) satisfying (3.3), we can find \( R(\gamma(\vartheta + \pi), \vartheta + \pi, t) \) which is symmetric to it with respect to the center point of the ellipse. Then, from (3.22) and Fig. 3-2, note that \( \gamma(\vartheta) = \gamma(\vartheta + \pi) \). From the perspective of the point \((\gamma(\vartheta), \vartheta)\), the agent movement observed with an initial position \( \rho(0) = \eta \) (following the dynamics in (3.18)) is the same as the movement observed from \((\gamma(\vartheta + \pi), \vartheta + \pi)\) if the agent starts from \( \rho(0) = \eta + \pi \) when \( T \to \infty \), since the cost in (3.18) is independent of initial conditions as \( T \to \infty \). Thus \( R(\gamma(\vartheta), \vartheta, t) = R(\gamma(\vartheta + \pi), \vartheta + \pi, t) \). Since, in addition, \( \sin \vartheta = \sin(\vartheta + \pi) \), we have \( \gamma(\vartheta) \frac{R(\gamma(\vartheta), \vartheta, t)}{\sin \vartheta} = \).
\[ \gamma(\vartheta + \pi) \frac{R(\gamma(\vartheta + \pi), \vartheta + \pi, t)}{\sin(\vartheta + \pi)} \] and it follows that

\[ \lim_{T \to \infty, b \to 0} \int_0^T \int_0^{2\pi} \frac{\gamma(\vartheta)}{\sin \vartheta} R(\gamma(\vartheta), \vartheta, t) \, d\vartheta \, dt = 0 \quad (3.25) \]

We now turn our attention to the last integral of (3.23). Two cases need to be considered here in view of (3.3):

(i) If \( \exists t' \) such that \( R(\xi, \vartheta, t') = 0 \) for \( t' \in (0, t) \), then let

\[ \tau_f(t) = \sup_{\tau \leq t} \{ \tau : R(\xi, \vartheta, \tau) = 0 \} \quad (3.26) \]

If \( \tau_f(t) < t \), then \( R(\xi, \vartheta, \tau) > 0 \) for all \( \tau \in [\tau_f(t), t) \) and \( \tau_f(t) \) is the last time instant prior to \( t \) when \( R(\xi, \vartheta, \tau) \) leaves an arc such that \( R(\xi, \vartheta, \tau) = 0 \). We can then write

\[ R(\xi, \vartheta, t) = \int_{\tau_f(t)}^t \dot{R}(\xi, \vartheta, \delta) \, d\delta. \]

Therefore,

\[ \frac{\partial R(\xi, \vartheta, t)}{\partial \delta} = \frac{\partial t}{\partial \delta} \dot{R}(\xi, \vartheta, t) + \frac{\partial \tau_f(t)}{\partial \delta} \dot{R}(\xi, \vartheta, \tau_f(t)) + \int_{\tau_f(t)}^t \frac{\partial \ddot{R}(\xi, \vartheta, \delta)}{\partial \delta} \, d\delta \quad (3.27) \]

Clearly, \( \frac{\partial t}{\partial \delta} = 0 \) and since \( \tau_f(t) \) is a time instant when \( R(\xi, \vartheta, t) \) leaves \( R(\xi, \vartheta, t) = 0 \) then, by Assumption 1, \( \dot{R}(\xi, \vartheta, \delta) \) is a continuous function and we have \( \dot{R}(\xi, \vartheta, \tau_f(t)) = 0 \). Therefore, (3.27) becomes

\[ \frac{\partial R(\xi, \vartheta, t)}{\partial \delta} = \int_{\tau_f(t)}^t \frac{\partial \dot{R}(\xi, \vartheta, \delta)}{\partial \delta} \, d\delta \quad (3.28) \]

where, from (3.3), \( \dot{R}(\xi, \vartheta, \delta) = A(\xi, \vartheta) - Bp(\xi, \vartheta, s(\delta)). \)

If, on the other hand, \( \tau_f(t) = t \), then \( R(\xi, \vartheta, t) = 0 \) and we define \( \sigma_f(t) = \sup_{\sigma \leq t} \{ \sigma : R(\xi, \vartheta, \sigma) > 0 \} \). Proceeding as above, we get

\[ \frac{\partial R(\xi, \vartheta, t)}{\partial \delta} = \int_{\sigma_f(t)}^t \frac{\partial \dot{R}(\xi, \vartheta, \delta)}{\partial \delta} \, d\delta \]

where now \( \dot{R}(\xi, \vartheta, \delta) = 0 \) and we get

\[ \frac{\partial R(\xi, \vartheta, t)}{\partial \delta} = 0 \quad (3.29) \]
(ii) $R(\xi, \vartheta, t') > 0$ for all $t' \in (0, t)$. In this case, we define $T(t) = 0$ and we have

$$R(\xi, \vartheta, t) = R(\xi, \vartheta, 0) + \int_{T(t)}^{t} \tilde{R}(\xi, \vartheta, \delta) d\delta,$$

where $\tilde{R}(\xi, \vartheta, \delta) = A(\xi, \vartheta) B \rho(\xi, \vartheta, s(t))$. Thus,

$$\frac{\partial R(\xi, \vartheta, t) }{ \partial b } = \frac{\partial R(\xi, \vartheta, 0) }{ \partial b } + \frac{\partial t}{ \partial b } \tilde{R}(\xi, \vartheta, t) + \int_{T(t)}^{t} \frac{\partial \tilde{R}(\xi, \vartheta, \delta) }{ \partial b } d\delta$$

(3.30)

Clearly, $\frac{\partial t}{ \partial b } = 0$ and $\frac{\partial R(\xi, \vartheta, 0) }{ \partial b } = 0$, since $R(\xi, \vartheta, 0)$ is the initial uncertainty value at $(\xi, \vartheta)$.

Then, (3.30) becomes

$$\frac{\partial R(\xi, \vartheta, t) }{ \partial b } = \int_{T(t)}^{t} \frac{\partial \tilde{R}(\xi, \vartheta, \delta) }{ \partial b } d\delta$$

(3.31)

which is the same result as (3.28).

Let us start by setting aside the much simpler case where (3.29) applies and consider (3.28) and (3.31). Noting that $\frac{\partial A(\xi, \vartheta) }{ \partial b } = 0$ we get

$$\frac{\partial \tilde{R}(\xi, \vartheta, \delta) }{ \partial b } = B \frac{\partial \rho(\xi, \vartheta, s(\delta)) }{ \partial b }$$

(3.32)

Recall that $[X, Y]$ has been selected to be the origin and that $\varphi = 0$. In this case, (3.17) becomes

$$s^x(t) = a \cos \rho(t), \quad s^y(t) = b \sin \rho(t)$$

(3.33)

Observing that $s^x(t)$ is independent of $b$, (3.32) gives

$$\frac{\partial \tilde{R}(\xi, \vartheta, \delta) }{ \partial b } = B \frac{\partial \rho(\xi, \vartheta, s(\delta)) }{ \partial s^y(\delta) } \frac{\partial s^y(\delta) }{ \partial b } = B \frac{\partial \rho(\xi, \vartheta, s(\delta)) }{ \partial D(\xi, \vartheta, s(\delta)) } \frac{\partial D(\xi, \vartheta, s(\delta)) }{ \partial s^y(\delta) } \sin \rho(\delta)$$

(3.34)

where $D(\xi, \vartheta, s(\delta)) = [(s^x(\delta) \xi \cos \vartheta)^2 + (s^y(\delta) \xi \sin \vartheta)^2]^{1/2}$, hence

$$\frac{\partial D(\xi, \vartheta, s(\delta)) }{ \partial s^y(\delta) } = \frac{s^y(\delta) \xi \sin \vartheta}{D(\xi, \vartheta, s(\delta))}$$

(3.35)
Using (3.35), (3.34), (3.28) in the second integral of (3.24), this integral becomes

\[
\int_{0}^{T} \int_{0}^{2\pi} \int_{0}^{\gamma(\vartheta)} \frac{\partial R(\xi, \vartheta, t)}{\partial b} \xi d\xi d\vartheta dt = B \int_{0}^{T} \int_{0}^{2\pi} \int_{0}^{\gamma(\vartheta)} \xi \int_{\tau_f}^{t} \frac{\partial p(\xi, \vartheta, s(\delta))}{\partial D(\xi, \vartheta, s(\delta))} \left( s_y(\delta) \frac{\xi \sin \vartheta}{D(\xi, \vartheta, s(\delta))} \right) \sin \rho(\delta) d\delta d\xi d\vartheta dt
\]

(3.36)

Note that when \( b \to 0 \), we have \( s_y(\delta) \to 0 \). In addition, \( p(\xi, \vartheta, s(\delta)) \) is a direct function of \( D(\xi, \vartheta, s(\delta)) \), so that \( \frac{\partial p(\xi, \vartheta, s(\delta))}{\partial D(\xi, \vartheta, s(\delta))} \) is not an explicit function of \( \xi, \vartheta \) or \( \delta \). Moreover, \( \sin \rho(\delta) \) is not a function of \( \vartheta \). Thus, switching the integration order in (3.36) we get

\[
B \int_{0}^{T} \int_{0}^{2\pi} \int_{0}^{\gamma(\vartheta)} \sin \rho(\delta) \int_{\tau_f}^{t} \frac{\xi^2 \sin \vartheta}{D(\xi, \vartheta, s(\delta))} d\xi d\vartheta d\delta dt
\]

Using Assumption 2, we make a symmetry argument similar to the one regarding (3.25). For any point \( \omega = (\xi, \vartheta) \in \mathbb{R}^2 \), we can find \( (\xi, \vartheta + \pi) \) which is symmetric to it with respect to the center point of the ellipse and Assumption 2 implies that \( A(\xi, \vartheta) = A(\xi, \vartheta + \pi) \).

Then, from the perspective of the point \( (\xi, \vartheta) \), the agent movement observed with an initial position \( \rho(0) = \eta \) (following the dynamics in (3.18)) is the same as the movement observed from \( (\xi, \vartheta + \pi) \) if the agent starts from \( \rho(0) = \eta + \pi \) when \( T \to \infty \), since the cost in (3.18) is independent of initial conditions as \( T \to \infty \). In addition, we again have \( \gamma(\vartheta) = \gamma(\vartheta + \pi) \), so that \( \int_{0}^{\gamma(\vartheta)} \frac{\sin \vartheta}{D(\xi, \vartheta, s(\delta))} = \int_{0}^{\gamma(\vartheta + \pi)} \frac{\sin(\vartheta + \pi)}{D(\xi, \vartheta + \pi, s(\delta))} \). Therefore,

\[
\lim_{T \to \infty} \int_{0}^{2\pi} \int_{0}^{\gamma(\vartheta)} \frac{\xi^2 \sin \vartheta}{D(\xi, \vartheta, s(\delta))} d\xi d\vartheta = 0
\]

(3.37)

and the second term of (3.24) gives

\[
\lim_{T \to \infty, b \to 0} \int_{0}^{2\pi} \frac{\partial R(\xi, \vartheta, t)}{\partial b} \xi dt d\xi = 0
\]

(3.38)

In view of (3.25) and (3.38), we have shown that the second term of (3.23) is 0 and we are
left with the first negative term from (3.19), giving the desired result:

\[
\lim_{T \to \infty, b \to 0} \frac{\partial J(b)}{\partial b} = \frac{\partial \Psi_{\Xi}}{\partial b} \int_0^T \int_{\Xi} R(\omega, t) \, d\omega \, dt < 0 \tag{3.39}
\]

Finally, if (3.29) applies instead of (3.28), then (3.29) and (3.25) immediately imply that the second term of (3.23) is 0, completing the proof. ■

Thus, we have proved that as \( T \to \infty \), when \( b \) is perturbed from 0 to some \( b_\epsilon > 0 \), an elliptical trajectory achieves a lower cost than a linear one. In other words, we have shown that elliptical trajectories are more suitable for a 2D mission space in terms of achieving near-optimal results in solving problem P1.

In other words, Prop. IV.1 shows that elliptical trajectories are more suitable for a 2D mission space in terms of achieving near-optimal results in solving problem P1.

### 3.4 Optimal Elliptical Trajectories

Based on our analysis thus far, we now tackle the problem of determining optimal solutions within the class of elliptical trajectories. Our approach is to associate with each agent an elliptical trajectory, parameterize each such trajectory by its center, orientation and major and minor axes, and then solve P3.1 as a parametric optimization problem. Note that this includes the possibility that two agents share the same trajectory if the solution to this problem results in identical parameters for the associated ellipses. Choosing elliptical trajectories, which are most likely suboptimal relative to a trajectory obtained through a TPBVP solution of P3.1, offers several practical advantages in addition to reduced computational complexity. Elliptical trajectories induce a periodic structure to the agent movements which provides predictability. As a result, it is also easier to handle issues related to collision avoidance.

For an elliptical trajectory, the \( n \)th agent movement is described as in (3.17) by

\[
\begin{align*}
\{ s^e_n(t) = & X_n + a_n \cos \rho_n(t) \cos \varphi_n - b_n \sin \rho_n(t) \sin \varphi_n \\
\{ \delta^e_n(t) = & Y_n + a_n \cos \rho_n(t) \sin \varphi_n + b_n \sin \rho_n(t) \cos \varphi_n
\end{align*}
\]  

(3.40)
where \([X_n, Y_n]\) is the center of the \(n\)th ellipse, \(a_n, b_n\) are its major and minor axes respectively and \(\varphi_n \in [0, \pi)\) is its orientation, i.e., the angle between the horizontal axis and the major axis of the \(n\)th ellipse. Note that the parameter \(\rho_n(t) \in [0, 2\pi)\) is the eccentric anomaly. Therefore, we replace problem \(P3.1\) by the determination of optimal parameter vectors \(Y_n = [X_n, Y_n, a_n, b_n, \varphi_n]^T, n = 1, \ldots, N\), and formulate the following problem \(P3.2\):

\[
P3.2: \quad \min_{Y_n, n=1, \ldots, N} J = \int_0^T \sum_{i=1}^M R_i(Y_1, \ldots, Y_N, t) dt \tag{3.41}
\]

Observe that the behavior of each agent under the optimal ellipse control policy is that of a hybrid system whose dynamics undergo switches when \(R_i(t)\) reaches or leaves the boundary value \(R_i = 0\) (the events" causing the switches). As a result, we are faced with a parametric optimization problem for a system with hybrid dynamics. We solve this hybrid system problem using a gradient-based approach in which we apply IPA to determine the gradients \(\nabla R_i(Y_1, \ldots, Y_N, t)\) on line (hence, \(\nabla J\)), i.e., directly using information from the agent trajectories and iterate upon them.

### 3.4.1 Infinitesimal Perturbation Analysis (IPA)

In our case, the parameter vectors are \(Y_n = [X_n, Y_n, a_n, b_n, \varphi_n]^T\) as defined earlier, and we seek to determine optimal vectors \(Y^*_n, n = 1, \ldots, N\). We will use IPA to evaluate \(\nabla J(Y_1, \ldots, Y_N) = [\frac{\partial J}{\partial Y_1}, \ldots, \frac{\partial J}{\partial Y_N}]^T\). From (3.41), this gradient clearly depends on \(\nabla R_i(t) = \left[\frac{\partial R_i(t)}{\partial Y_1}, \ldots, \frac{\partial R_i(t)}{\partial Y_N}\right]^T\). In turn, this gradient depends on whether the dynamics of \(R_i(t)\) in (3.3) are given by \(\dot{R}_i(t) = 0\) or \(\dot{R}_i(t) = A_i BP_i(s(t))\). The dynamics switch at event times \(\tau_k, k = 1, \ldots, K\), when \(R_i(t)\) reaches or escapes from 0 which are observed on a trajectory over \([0, T]\) based on a given \(Y_n, n = 1, \ldots, N\).

**IPA equations.** We begin by recalling the dynamics of \(R_i(t)\) in (3.3) which depend on the relative positions of all agents with respect to \([\alpha_i, \beta_i]\) and change at time instants \(\tau_k\) such that either \(R_i(\tau_k) = 0\) with \(R_i(\tau_k) > 0\) or \(A_i > BP_i(s(\tau_k))\) with \(R_i(\tau_k) = 0\). Moreover, the agent positions \(s_n(t) = [s_{n1}(t), s_{n2}(t)]\), \(n = 1, \ldots, N\), on an elliptical trajectory are expressed using (3.40). Viewed as a hybrid system, we can now concentrate on all events causing
transitions in the dynamics of $R_i(t), i = 1, \ldots, M$, since any other event has no effect on the values of $\nabla R_i(\Upsilon_1, \ldots, \Upsilon_N, t)$ at $t = \tau_k$.

For notational simplicity, we define $\omega_i = [\alpha_i, \beta_i] \in \Omega$. First, if $R_i(t) = 0$ and $A(\omega_i)BP(\omega_i, s(t)) \leq 0$, applying (2.27) to $R_i(t)$ and using (3.3) gives

$$\frac{d}{dt} \frac{\partial R_i(t)}{\partial \Upsilon_n} = 0$$  \hspace{1cm} (3.42)

When $R_i(t) > 0$, we have

$$\frac{d}{dt} \frac{\partial R_i(t)}{\partial \Upsilon_n} = B \frac{\partial p_n(\omega_i, s_n(t))}{\partial \Upsilon_n} \prod_{d \neq n}^{N} |1 - p_d(\omega_i, s_d(t))|$$  \hspace{1cm} (3.43)

Noting that $p_n(\omega_i, s_n(t)) = p_n(D(\omega_i, s_n(t)))$, we have

$$\frac{\partial p_n(\omega_i, s_n(t))}{\partial \Upsilon_n} = \frac{\partial p_n(D(\omega_i, s_n(t)))}{\partial \Upsilon_n} \frac{\partial D(\omega_i, s_n(t))}{\partial \Upsilon_n}$$  \hspace{1cm} (3.44)

where $D(\omega_i, s_n(t)) = [(s_x^n(t) \alpha_i)^2 + (s_y^n(t) \beta_i)^2]^{1/2}$. For simplicity, we write $D = D(\omega_i, s_n(t))$ and we get

$$\frac{\partial D}{\partial \Upsilon_n} = \frac{1}{2D} \left( \frac{\partial D}{\partial s_x^n} \frac{\partial s_x^n}{\partial \Upsilon_n} + \frac{\partial D}{\partial s_y^n} \frac{\partial s_y^n}{\partial \Upsilon_n} \right)$$  \hspace{1cm} (3.45)

where $\frac{\partial D}{\partial s_x^n} = 2(s_x^n \alpha_i)$ and $\frac{\partial D}{\partial s_y^n} = 2(s_y^n \beta_i)$. Note that $\frac{\partial s_x^n}{\partial \Upsilon_n} = [\frac{\partial s_x^n}{\partial X_n}, \frac{\partial s_x^n}{\partial Y_n}, \frac{\partial s_x^n}{\partial \alpha_n}, \frac{\partial s_x^n}{\partial \beta_n}, \frac{\partial s_x^n}{\partial \phi_n}]^T$ and $\frac{\partial s_y^n}{\partial \Upsilon_n} = [\frac{\partial s_y^n}{\partial X_n}, \frac{\partial s_y^n}{\partial Y_n}, \frac{\partial s_y^n}{\partial \alpha_n}, \frac{\partial s_y^n}{\partial \beta_n}, \frac{\partial s_y^n}{\partial \phi_n}]^T$. From (3.40), for $\frac{\partial s_x^n}{\partial \Upsilon_n}$, we obtain

$$\frac{\partial s_x^n}{\partial X_n} = 1, \quad \frac{\partial s_x^n}{\partial Y_n} = 0$$

$$\frac{\partial s_x^n}{\partial \alpha_n} = \cos \rho_n(t) \cos \varphi_n, \quad \frac{\partial s_x^n}{\partial \beta_n} = \sin \rho_n(t) \sin \varphi_n$$

$$\frac{\partial s_x^n}{\partial \phi_n} = a_n \cos \rho_n(t) \sin \varphi_n \quad b \sin \rho_n(t) \cos \varphi_n$$

Similarly, we have

$$\frac{\partial s_y^n}{\partial X_n} = 0, \quad \frac{\partial s_y^n}{\partial Y_n} = 1$$

$$\frac{\partial s_y^n}{\partial \alpha_n} = \cos \rho_n(t) \sin \varphi_n, \quad \frac{\partial s_y^n}{\partial \beta_n} = \sin \rho_n(t) \cos \varphi_n$$

$$\frac{\partial s_y^n}{\partial \phi_n} = a_n \cos \rho_n(t) \cos \varphi_n \quad b \sin \rho_n(t) \sin \varphi_n$$
Using $\frac{\partial \tau_i}{\partial \tau_n}$ and $\frac{\partial \tau_i^+}{\partial \tau_n}$ in (3.45) and then (3.44) and back into (3.43), we can finally obtain $\frac{\partial R_i(t)}{\partial \tau_n}$ for $t \in [\tau_k, \tau_{k+1}]$ as

$$\frac{\partial R_i(t)}{\partial \tau_n} = \frac{\partial R_i}{\partial \tau_n} \tau_i^+ + \begin{cases} 0 & \text{if } R_i(t) = 0, \\
A_i \leq BP_i(s(t)) & \text{otherwise} \end{cases}$$

where the integral above is obtained from (3.42)-(3.44). Thus, it remains to determine the components $\nabla R_i(\tau_i^+)$ in (3.46) using (2.28). This involves the event time gradient vectors $\nabla \tau_k = \frac{\partial \tau_k}{\partial \tau_n}$ for $k = 1, \ldots, K$, which will be determined through (2.29). There are two possible cases regarding the events that cause switches in the dynamics of $R_i(t)$:

**Case 1:** At $\tau_k$, $\dot{R}_i(t)$ switches from $\dot{R}_i(t) = 0$ to $\dot{R}_i(t) = A_i \ BP_i(s(t))$. In this case, it is easy to see that the dynamics $R_i(t)$ are continuous, so that $f_k(\tau_k) = f_k(\tau_k^+)$ in (2.28) applied to $R_i(t)$ and we get

$$\nabla R_i(\tau_k^+) = \nabla R_i(\tau_k), \quad i = 1, \ldots, M \tag{3.47}$$

**Case 2:** At $\tau_k$, $\dot{R}_i(t)$ switches from $\dot{R}_i(t) = A_i \ BP_i(s(t))$ to $\dot{R}_i(t) = 0$, i.e., $R_i(\tau_k)$ becomes zero. In this case, we need to first evaluate $\nabla \tau_k$ from (2.29) in order to determine $\nabla R_i(\tau_k^+)$ through (2.28). Observing that this event is endogenous, (2.29) applies with $g_k = R_i = 0$ and we get

$$\nabla \tau_k = \frac{\nabla R_i(\tau_k)}{A(\omega_i) \ BP(\omega_i, s(\tau_k))} \tag{3.48}$$

It follows from (2.28) that

$$\nabla R_i(\tau_k^+) = \nabla R_i(\tau_k) \left[ \frac{A(\omega_i) \ BP(\omega_i, s(\tau_k))}{A(\omega_i) \ BP(\omega_i, s(\tau_k))} \nabla R_i(\tau_k^+) \right] = 0 \tag{3.49}$$

Thus, $\nabla R_i(\tau_k^+)$ is always reset to 0 regardless of $\nabla R_i(\tau_k^+)$.  

**Objective Function Gradient Evaluation.** Based on our analysis, we first rewrite $J$ in (3.41) as

$$J(T_1, \ldots, T_N) = \sum_{i=1}^M \sum_{k=0}^K \int_{\tau_k(T_1, \ldots, T_N)}^{\tau_{k+1}(T_1, \ldots, T_N)} R_i(T_1, \ldots, T_N, t)dt$$
and (omitting some function arguments) we get

$$\nabla J = \sum_{i=1}^{M} \sum_{k=0}^{K} \left( \int_{\tau_k}^{\tau_{k+1}} \nabla R_i(t) \, dt + R_i(\tau_{k+1}) \nabla(\tau_{k+1}) - R_i(\tau_k) \nabla(\tau_k) \right)$$

Observing the cancelation of all terms of the form $R_i(\tau_k) \nabla(\tau_k)$ for all $k$ (with $\tau_0 = 0$, $\tau_{K+1} = T$ fixed), we finally get

$$\nabla J(Y_1, \ldots, Y_N) = \sum_{i=1}^{M} \sum_{k=0}^{K} \int_{\tau_k}^{\tau_{k+1}} \nabla R_i(t) \, dt \tag{3.50}$$

This depends entirely on $\nabla R_i(t)$, which is obtained from (3.46) and the event times $\tau_k$, $k = 1, \ldots, K$, given initial conditions $s_n(0)$ for $n = 1, \ldots, N$, and $R_i(0)$ for $i = 1, \ldots, M$. In (3.46), $\frac{\partial R_i(\tau_k)}{\partial Y_n}$ is obtained through (3.47)-(3.49), whereas $\frac{d}{dt} \frac{\partial R_i(t)}{\partial Y_n}$ is obtained through (3.42)-(3.45).

**Remark 3.2.** Observe that the evaluation of $\nabla R_i(t)$, hence $\nabla J$, is independent of $A_i$, $i = 1, \ldots, M$, i.e., the values in our uncertainty model. In fact, the dependence of $\nabla R_i(t)$ on $A_i$, $i = 1, \ldots, M$, manifests itself through the event times $\tau_k$, $k = 1, \ldots, K$, that do affect this evaluation, but they, unlike $A_i$ which may be unknown, are directly observable during the gradient evaluation process. Thus, the IPA approach possesses an inherent robustness property: there is no need to explicitly model how uncertainty affects $R_i(t)$ in (3.3). Consequently, we may treat $A_i$ as unknown without affecting the solution approach (the values of $\nabla R_i(t)$ are obviously affected). We may also allow this uncertainty to be modeled through random processes $\{A_i(t)\}, i = 1, \ldots, M$; in this case, however, the result of Proposition 3.1 no longer applies without some conditions on the statistical characteristics of $\{A_i(t)\}$ and the resulting $\nabla J$ is an estimate of a stochastic gradient.

**Remark 3.3.** Note that the number of agents affects the number of derivative components in (3.50), so the complexity of $\nabla J(Y_1, \ldots, Y_N)$ in (3.50) grows linearly in the number of agents $N$. In addition, the calculation of $\nabla J(Y_1, \ldots, Y_N)$ in (3.50) grows linearly in $T$, as a longer operation time only implies more events at whose occurrence times $\tau_k$ the objective function gradient is updated. In other words, solving the problem using IPA is scalable with
respect to the number of agents and the operation time.

### 3.4.2 Objective Function Optimization

We now seek to obtain $[T_1, \ldots, T_N]$ minimizing $J(T_1, \ldots, T_N)$ through a standard gradient-based optimization algorithm of the form

$$[Y_1^{t+1}, \ldots, Y_N^{t+1}] = [Y_1^t, \ldots, Y_N^t] - \eta^t [\nabla J(T_1^t, \ldots, T_N^t)]$$  \hspace{1cm} (3.51)$$

where $\{\eta_n^l\}$, $l = 1, 2, \ldots$ are appropriate step size sequences and $\nabla J(T_1^t, \ldots, T_N^t)$ is the projection of the gradient $\nabla J(T_1, \ldots, T_N)$ onto the feasible set, i.e., $s_n(t) \in \Omega$ for all $t \in [0, T]$, $n = 1, \ldots, N$. The optimization algorithm terminates when $|\nabla J(T_1^t, \ldots, T_N^t)| < \varepsilon$ (for a fixed threshold $\varepsilon$) for some $[T_1^*, \ldots, T_N^*]$. When $\varepsilon > 0$ is small, $[T_1^t, \ldots, T_N^t]$ is believed to be in the neighborhood of the local optimum, then we set $[T_1^*, \ldots, T_N^*] = [T_1^t, \ldots, T_N^t]$. However, in our problem the function $J(T_1, \ldots, T_N)$ is non-convex and there are actually many local optima depending on the initial controllable parameter vector $[T_1^0, \ldots, T_N^0]$. In the next section, we propose a stochastic comparison algorithm which addresses this issue by randomizing over the initial points $[T_1^0, \ldots, T_N^0]$. This algorithm defines a process which converges to a global optimum under certain well-defined conditions.

### 3.5 Stochastic Comparison Algorithm for global optimality

Gradient-based optimization algorithms are generally efficient and effective in finding the global optimum when one is uniquely specified by the point where the gradient is zero. When this is not the case, to seek a global optimum one must resort to several alternatives which include a variety of random search algorithms. In this section, we use the Stochastic Comparison algorithm in (Bao and Cassandras, 1996) to find the global optimum. As shown in (Bao and Cassandras, 1996), for a stochastic system, if $(i)$, the cost function $J(T)$ is continuous in $T$ and $(ii)$, for each estimate $\hat{J}(T)$ of $J(T)$ the error $W(T) = \hat{J}(T) - J(T)$ has a symmetric pdf, then the Markov process $\{T_k\}$ generated by the Stochastic Comparison algorithm will converge to an $\epsilon$ optimal interval of the global optimum for arbitrarily
small $\epsilon > 0$. In short, \( \lim_{k \to \infty} P[\mathcal{T}^k \in \mathcal{T}_{\epsilon}^*] = 1 \), for any $\epsilon > 0$, where $\mathcal{T}_{\epsilon}^*$ is defined as $\mathcal{T}_{\epsilon}^* = \{ \mathcal{T} | J(\mathcal{T}) \leq J(\mathcal{T}^*) + \epsilon \}$. Using the Continuous Stochastic Comparison (CSC) Algorithm developed in (Bao and Cassandras, 1996) for a general continuous optimization problem, consider $\mathcal{T} \in \Phi$ to be a controllable vector, where $\Phi$ is the bounded feasible controllable parameter space. The Stochastic Comparison Algorithm is presented in Algorithm 3. In

**Algorithm 3**: Continuous Stochastic Comparison (CSC) Algorithm.

1: Initialize $\mathcal{T}^0 = \phi^0$, $k = 0$.
2: For a given $\mathcal{T}^k = \phi^k$, sample the next candidate point $Z^k$ from $\Phi$ according to a uniform distribution over $\Phi$.
3: For a given $Z^k = \zeta^k$, set

$$
\mathcal{T}^{k+1} = \begin{cases} 
Z^k, & \text{with probability } p^{k,}\ 
\mathcal{T}^k, & \text{with probability } 1 - p^{k,}
\end{cases}
$$

where $p^k = \{ P[\hat{J}(\zeta^k) < \hat{J}(\phi^k)] \}^{L_k}$. 
4: Replace $k$ by $k + 1$, and go to Step 2.

the CSC algorithm, the probability $p^k$ is actually not calculable, since we do not know the underlying probability functions. However, it is realizable in the following way: both $\hat{J}(\zeta^k)$ and $\hat{J}(\phi^k)$ are estimated $L_k$ times for an appropriately selected increasing sequence $\{L_k\}$. If $\hat{J}(\zeta^k) < \hat{J}(\phi^k)$ every time, we set $\mathcal{T}^{k+1} = Z^k$. Otherwise, we set $\mathcal{T}^{k+1} = \mathcal{T}^k$.

As discussed in Remark 3.3, the persistent monitoring problem P3.2 becomes a stochastic optimization problem if $A_i(t)$, $i = 1, \ldots, M$, are stochastic processes. However, for the deterministic setting in which all $A_i$ are constant, the observed value $\hat{J}$ coincides with the actual value $J$ and a one-time comparison $\hat{J}(\zeta^k) < \hat{J}(\phi^k)$ is sufficient to replace $\phi^k$ with $\zeta^k$ for $\mathcal{T}^{k+1}$. In this case, step 3 in Algorithm 3 becomes, for a given $Z^k = \zeta^k$:

$$
\mathcal{T}^{k+1} = \begin{cases} 
Z^k, & \text{if } J(\zeta^k) < J(\phi^k) \\
\mathcal{T}^k, & \text{otherwise}
\end{cases}
$$

and the CSC algorithm in this deterministic setting reduces to a comparison algorithm with multi-starts over the 6-dimensional controllable vector $\mathcal{T}_n = [X_n, Y_n, a_n, b_n, \varphi_n, \rho_n]^T$, for each ellipse associated with agent $n = 1, \ldots, N$. 
Algorithm 4: IPA-based Optimization Algorithm using CSC to find $Y_n$, $n = 1, \ldots, N$.

1: Set $\epsilon > 0$, $k = 0$. Initialize $Y^0 = \phi^0$, where $\phi^0 = [Y^0_1, \ldots, Y^0_N]$. Initialize $L_0$, where $\{L_k\}$ is an appropriately selected increasing sequence.

2: while $k < K$, do
3: For a given $Y^k = \phi^k$,
4: repeat
5: Compute $s_n(t)$, $t \in [0, T]$ using (3.40) and $\phi^k$ for $n = 1, \ldots, N$
6: Compute $\tilde{J}^k(\phi^k)$, $\tilde{\nabla}J(\phi^k)$ and update $\phi^k$ through (4.12).
7: until $|\tilde{\nabla}J(\phi^k)| < \epsilon$
8: Sample the next candidate point $Z^k$ from $\Phi$ according to a uniform distribution over $\Phi$. For a given $Z^k = \zeta^k$,
9: repeat
10: Compute $s_n(t)$, $t \in [0, T]$ using (3.40) and $\zeta^k$ for $n = 1, \ldots, N$
11: Compute $\tilde{J}(\zeta^k)$, $\tilde{\nabla}J(\zeta^k)$ and update $\zeta^k$ through (4.12).
12: until $|\tilde{\nabla}J(\zeta^k)| < \epsilon$
13: Set
$$Y^{k+1} = \begin{cases} Z^k, & \text{with probability } p^k, \\ Y^k, & \text{with probability } 1 - p^k, \end{cases}$$
with $p^k = \{P[\tilde{J}(\zeta^k) < \tilde{J}(\phi^k)]\}L_k$.
14: Replace $k$ by $k + 1$.
15: end while
16: Set $Y^* = Y^K$.

3.6 Numerical Experiments

We begin with a two-agent example in which we solve P3.2 by assigning elliptical trajectories using the gradient-based approach in Section V.B (without the CSC Algorithm 4).

The environment setting parameters used are: $r = 4$ for the sensing range of agents; $L_1 = 20$, $L_2 = 10$, for the mission space dimensions; and $T = 200$. All sampling points $[\alpha_i, \beta_i]$ are uniformly spaced within $L_1 \times L_2$, $i = 1, \ldots, M$ where $M = (L_1 + 1)(L_2 + 1) = 231$. Initial values for the uncertainty functions are $R_i(0) = 2$ and $B = 6$, $A_i = 0.2$ for all $i = 1, \ldots, M$ in (3.3). The results are shown in Fig. 3.3. Note that the initial conditions were set so as to approximate linear trajectories (red ellipses), thus illustrating Proposition IV.1: we can see that larger ellipses achieve a lower total uncertainty value per unit area. Moreover, observe that the initial cost is significantly reduced, indicating the importance of optimally selecting the ellipse sizes, locations and orientations. The cost associated with the final blue elliptical trajectories in this case is $J_e = 6.93 \times 10^4$. 
Using the same initial trajectories as in Fig. 3.3(a), we also used a TPBVP solution algorithm for P3.1. The results are shown in Fig. 3-4. The TPBVP algorithm is computationally expensive and time consuming (about 800,000 steps to converge). Interestingly, the solution corresponds to a cost $J_{TPBVP} = 7.15 \times 10^4$, which is higher than that of Fig. 3-3 where solutions were restricted to the set of elliptical trajectories. This is an indication of the presence of locally optimal trajectories.

Next, we solve the same two-agent example with the same environment setting using the CSC Algorithm 4. For simplicity, we select the ellipse center location $[X_n, Y_n]$ as the only two (out of six) multi-start components: for a given number of comparisons $Q$, we sample the ellipse center $[X_n, Y_n] \in L_1 \times L_2$, $n = 1, \ldots, N$, using a uniform distribution while $a_n = 5, b_n = 2, \varphi_n = \frac{\pi}{4}, \rho_n = 0$, for $n = 1, 2$ are randomly assigned but initially fixed parameters during the number of comparisons $Q$ (thus, it is still possible that there are local minima with respect to the remaining four components $[a_n, b_n, \varphi_n, \rho_n]$, but, clearly, all six components in $Y_n$ can be used at the expense of some additional computational cost.) In Fig. 3-5, the red elliptical trajectories on the left show the initial ellipses and the blue trajectories represent the corresponding resulting ellipses the CSC Algorithm 4 converges to. Figure 3.5(b) shows the cost vs. number of iterations of the CSC algorithm. The resulting cost for $Q = 300$ is $J_{CSC} = 6.57 \times 10^4$, where "Det" stands for a deterministic environment. It is clear from Fig. 3.5(b) that the cost of the worst local minimum is much higher than that of the best local minimum. Note also that the CSC Algorithm 4 does improve the original pure gradient-based algorithm performance $J_e = 6.93 \times 10^4$.

In Fig. 3-6, the values of $A_i$ are allowed to be random, thus dealing with a persistent monitoring problem in a stochastic mission space, where we can test the robustness of the IPA approach as discussed in Remark 3.2. In particular, each $A_i$ is treated as a piecewise constant random process $\{A_i(t)\}$ such that $A_i(t)$ takes on a fixed value sampled from a uniform distribution over $(0.195, 0.205)$ for an exponentially distributed time interval with mean 5 before switching to a new value. The sequence $\{M_k\}$ defining the number of cost comparisons made at the $k$th iteration is set so as to grow sublinearly with $M_k =$
\[ 10 \log k, k = 2, \ldots, Q. \] Note that the system in this case is very similar to that of Fig. 3.5 where \( A_i = 0.2 \) for all \( i \) without any change in the way in which \( \nabla J(T_1, \ldots, T_N) \) is evaluated in executing (3.51). As already pointed out, this exploits a robustness property of IPA which makes the evaluation of \( \nabla J(T_1, \ldots, T_N) \) independent of the values of \( A_i \). All other parameter settings are the same as in Fig. 3.5. In Fig. 3.6(a), the red elliptical trajectories show the initial ellipses and the blue trajectories represent the corresponding resulting ellipses the CSC Algorithm 4 converges to. The resulting cost for \( Q = 300 \) in Fig. 3.6(b) is \( J_{\text{CSC}}^{\text{Sto}} = 6.60 \times 10^4 \), where "Sto" stands for a stochastic environment. This cost is almost the same as \( J_{\text{CSC}}^{\text{Det}} = 6.57 \times 10^4 \), showing that the IPA approach is indeed robust to a stochastic environment setting.

Finally, Fig. 3.7 shows the TPBVP algorithm result when using the optimal (blue) ellipses in Fig. 3.5(a) as the initial trajectories. The trajectories the TPBVP solver converges to are shown in red and green respectively for each agent. The corresponding cost in Fig. 3.7(b) is \( J_{\text{TPBVP}} = 6.07 \times 10^4 \), which is an improvement compared to \( J_{\text{CSC}}^{\text{Det}} = 6.57 \times 10^4 \) obtained for elliptical trajectories from the CSC Algorithm 4. Compared to the computationally expensive TPBVP algorithm, the CSC Algorithm 4 using IPA is inexpensive and scalable with respect to \( T \) and \( N \). Thus, a combination of the two provides the benefit of offering the optimal elliptical trajectories obtained through the CSC Algorithm 4 (the first fast phase of a solution approach) as initial trajectories for the TPBVP algorithm (the second much slower phase.) This combination is faster than the original TPBVP algorithm and can also achieve a lower cost compared to CSC Algorithm 4.

### 3.7 Summary

We have shown that an optimal control solution to the 1D persistent monitoring problem does not easily extend to the 2D case. In particular, we have proved that elliptical trajectories outperform linear ones in a 2D mission space. Therefore, we have sought to solve a parametric optimization problem to determine optimal elliptical trajectories. Numerical examples indicate that this scalable approach (which can be used on line) provides solu-
(a) Red ellipses are the initial trajectories and (b) Cost as a function of algorithm iterations. blue ellipses are the final trajectories. $J_e = 6.93 \times 10^4$.

**Figure 3-3:** Optimal elliptical trajectories for two agents (without using the CSC algorithm.)

---

(a) Red and green trajectories obtained from TP- BVP solution. (b) Cost as a function of algorithm iterations. $J_{TPBVP} = 7.15 \times 10^4$.

**Figure 3-4:** Optimal trajectories using TPBVP solver for two agents. Initial trajectories are red curves in Fig. 3.3(a).

---

(a) Red ellipses: initial trajectories. Blue ellipses: (b) Cost as a function of algorithm iterations. optimal elliptical trajectories $J_{CSC} = 6.57 \times 10^4$.

**Figure 3-5:** Two agent example for the deterministic environment setting using the CSC Algorithm 1 for $Q = 300$ trials.
Figure 3.6: Two-agent example for a stochastic environment setting using the CSC Algorithm 1 for $Q = 300$ trials, where $A_i(\Delta t_i)^{-1} U (0.195, 0.205)$, $\Delta t_i = 0.2e^{-0.2t}$.

Figure 3.7: Left plot: elliptical trajectories (blue curve) obtained in Fig. 3.5(a) used as initial trajectories for the TPBVP solver.
tions that approximate those obtained through a computationally intensive TPBVP solver. Moreover, since the solutions obtained are generally locally optimal, we have incorporated a stochastic comparison algorithm for deriving globally optimal elliptical trajectories. Ongoing work aims at alternative approaches for near-optimal solutions and at distributed implementations.
Chapter 4

Trajectories Characterized by General Functions for 2-dimensional Persistent Monitoring Problem

We address the persistent monitoring problem in 2-dimensional mission spaces, where the objective is to control the movement of multiple cooperating agents to minimize an uncertainty metric. In a 1-dimensional mission space, it has been shown in Chapter 2 that the optimal solution is for each agent to move at maximal speed and switch direction at specific points, possibly waiting some time at each such point before switching. In a 2-dimensional mission space, such simple solutions can no longer be derived. In Chapter 3, we formulate a 2D persistent monitoring problem as one of determining optimal elliptical trajectories for a given number of agents. In this chapter, we tackle the same 2D persistent monitoring problem by representing an agent trajectory in terms of general function families characterized by a set of parameters that we can optimize. We then show that the problem of determining optimal parameters for these trajectories can be solved using IPA to determine gradients of the objective function with respect to these parameters evaluated on line so as to adjust them through a standard gradient-based algorithm. We have applied this approach to the family of Lissajous functions as well as a Fourier series representation of an agent trajectory. Numerical examples indicate that this scalable approach provides solutions that are near-optimal relative to those obtained through a computationally intensive two point boundary value problem solver.

Our contribution in this chapter is to represent an agent trajectory in terms of general function families characterized by a set of parameters that we seek to optimize, given a persistent monitoring objective function (we note that this includes the possibility that one or more agents share the same trajectory). In particular, we study two families of
functions: Lissajous functions (Cundy and Rollett, 1989) and a Fourier series representation of a trajectory. Motivated by the simple oscillatory optimal trajectory structure in the 1-dimensional problem, we consider Lissajous functions because of their property to systematically describe complex harmonic motion in a 2-dimensional space. Trajectories based on a Fourier series representation, on the other hand, are used to approximate any arbitrary trajectory and are more suitable when the mission space is irregular (i.e., its shape is complex or the weights and distribution of sampling points in the mission space are inhomogeneous). We derive suitable parameterizations for both trajectory representations and show that the problem of determining optimal parameters can be explicitly solved using similar IPA techniques as in our 1-dimensional analysis and the 2-dimensional analysis limited to elliptical trajectories. This is done through IPA gradients of the objective function with respect to these parameters evaluated on line so as to adjust them towards optimality.

The remainder of the chapter is organized as follows. We skip the problem formulation and the Hamiltonian analysis as they are shown in Section 3.1 and Section 3.2. Section 4.1 formulates and solve the problem of determining optimal trajectories based on general function representations using a gradient-based algorithm using IPA. In Section 4.2, we concentrate on two particular function families applying the general analysis. Section 4.3 provides numerical results and Section 4.4 concludes the chapter.

4.1 Trajectory Optimization Using General Function Families Formulation

The goal of the optimal persistent monitoring problem we consider is still to control through \( u_n(t), \theta_n(t) \) in (3.1) the movement of the \( N \) agents so that the cumulative uncertainty over all sensing points \( \{[\alpha_1, \beta_1], \ldots, [\alpha_M, \beta_M]\} \) is minimized over a fixed time horizon \( T \). Thus, setting \( u(t) = [u_1(t), \ldots, u_N(t)] \) and \( \theta(t) = [\theta_1(t), \ldots, \theta_N(t)] \) we aim to solve optimal control problem \( \text{P3.1} \), subject to the agent dynamics (3.1), uncertainty dynamics (3.3), control constraint \( 0 \leq u_n(t) \leq 1, 0 \leq \theta_n(t) \leq 2\pi, t \in [0, T] \), and state constraints \( s_n(t) \in \Omega \) for all \( t \in [0, T], n = 1, \ldots, N \).
Following the same Hamiltonian analysis, we have shown that \( u_n^*(t) = 1 \) \( n = 1, \ldots, N \), in (3.11), we are only left with the task of determining \( \theta_n^*(t), n = 1, \ldots, N \). As introduced in Alg. 2, this can be accomplished by solving a standard TPBVP problem involving forward and backward integrations of the state and costate equations to evaluate \( \frac{\partial H}{\partial \theta_n} \) after each such iteration and using a gradient descent approach until the objective function converges to a (local) minimum. As we stated before, this is a computationally intensive process which scales poorly with the number of agents and the size of the mission space. In addition, it requires discretizing the mission time \( T \) and calculating every control at each time step which adds to the computational complexity.

Given the complexity of the TPBVP required to obtain an optimal solution of problem \( P3.1 \), we seek alternative approaches which may be suboptimal but are tractable and scalable. The first such effort was motivated by the results obtained in the 1-dimensional analysis (Cassandras et al., 2013), where we found that on a mission space defined by a line segment \([0, L]\) the optimal trajectory for each agent is to move at full speed until it reaches some switching point, dwell on the switching point for some time (possibly zero), and then switch directions. Thus, each agent’s optimal trajectory is fully described by a set of switching points \( \{\theta_1, \ldots, \theta_K\} \) and associated waiting times at these points, \( \{w_1, \ldots, w_K\} \). The values of these parameters can then be efficiently determined using a gradient-based algorithm with IPA to evaluate the objective function gradient. In the 2-dimensional case, our approach here is to view each agent’s trajectory as represented by parametric equations

\[
\begin{align*}
    s_n^x(t) &= f(\Upsilon_n, \rho_n(t)) \\
    s_n^y(t) &= g(\Upsilon_n, \rho_n(t))
\end{align*}
\]

for all \( n = 1, \ldots, N \). In (4.1), \( \Upsilon_n = [\Upsilon_n^1, \Upsilon_n^2, \ldots, \Upsilon_n^T] \) is the vector of parameters through which we control the shapes and locations of the \( n \)th agent trajectory and is this vector’s dimension. The agent position over time is controlled by a function \( \rho_n(t) \). Note that we suppress the dependence of \( f(\cdot) \) and \( g(\cdot) \) on the inputs \( u_n(t) \) and \( \theta_n(t) \) in (3.1) and stress instead its dependence on the parameter vector \( \Upsilon_n \) which may generally affect both \( u_n(t) \) and \( \theta_n(t) \).
The 2-dimensional persistent monitoring problem can then be formulated through the following tasks: (i) find \( N \) parametric trajectories in terms of their shape and exact location in \( \Omega \), noting that one or more agents may share one of these trajectories, and (ii) control the motion of each agent on its trajectory through \( \rho_n(t) \). Since we have shown that \( u^*_n(t) = 1 \), \( n = 1, \ldots, N \), it follows that \((s^*_h)^2 + (s^*_v)^2 = 1\), which gives from (4.1):

\[
\left( \frac{\partial f(T_n, \rho_n(t))}{\partial \rho_n(t)} \dot{\rho}_n(t) \right)^2 + \left( \frac{\partial g(T_n, \rho_n(t))}{\partial \rho_n(t)} \dot{\rho}_n(t) \right)^2 = 1
\]  

(4.2)

Note that each \( T_n \) is a constant parameter vector independent of \( t \). From (4.2), we obtain the dynamics of \( \rho(t) \) as

\[
\dot{\rho}_n(t) = \left[ \left( \frac{\partial f(T_n, \rho_n(t))}{\partial \rho_n(t)} \right)^2 + \left( \frac{\partial g(T_n, \rho_n(t))}{\partial \rho_n(t)} \right)^2 \right]^{1/2}
\]  

(4.3)

Initially, we set \( \rho_n(0) = 0 \). Thus, task (ii) is simple, since we always let each agent move at full speed on its assigned trajectory by controlling \( \rho_n(t) \) through (4.3) with initial condition \( \rho_n(0) = 0 \).

We are now left with only task (i) to be addressed. Optimal values of the parameters in \( T_n \) can be obtained by solving an optimization problem using on-line gradient information obtained through IPA, similar to the approach used in (Lin and Cassandras, 2013) where we limited ourselves to elliptical trajectories. Note that this includes the possibility that two or more agents share the same trajectory if the solution to this problem results in identical parameters for the associated parametric curve. The optimization problem for our persistent monitoring setting is \( P_{3.1} \), where we replace \( u(t), \theta(t) \) by the parameter vectors \( T_n, n = 1, \ldots, N \) and obtain the following parametric optimization problem \( P_{4.1} \):

\[
\min_{T_n, n = 1, \ldots, N} J = \int_0^T \sum_{i=1}^M R_i(T_1, \ldots, T_N, t) dt
\]  

(4.4)

We solve this problem using a gradient-based approach to determine the gradients \( \nabla R_i(T_1, \ldots, T_N, t) \) (hence, also \( \nabla J \)). We proceed in which using IPA to evaluate these gradients on line, i.e., by directly using information from the agent trajectories and iterate
upon them.

In our case, the parameter vectors are $Y_n = [Y_n^1, Y_n^2, \ldots, Y_n^n]^T, n = 1, \ldots, N$ as defined earlier, and we seek to determine optimal vectors $Y_n^*$. We will use IPA to evaluate $\nabla J(Y_1, \ldots, Y_N) = [\frac{dJ}{dT_1}, \ldots, \frac{dJ}{dT_N}]^T$. From (4.4), this gradient clearly depends on $\nabla R_i(t) = \left[ \frac{\partial R_i(t)}{\partial T_1}, \ldots, \frac{\partial R_i(t)}{\partial T_N} \right]^T$. In turn, this gradient depends on whether the dynamics of $R_i(t)$ in (3.3) are given by $\dot{R}_i(t) = 0$ or $\dot{R}_i(t) = A_i BP_i(s(t))$. The dynamics switch at event times $\tau_k, k = 1, \ldots, K$, when $R_i(t)$ reaches or escapes from 0 which are observed on a sample path over $[0, T]$ based on a given $Y_n, n = 1, \ldots, N$.

**IPA equations.** We begin by recalling the dynamics of $R_i(t)$ in (3.3) which depend on the relative positions of all agents with respect to $\omega_i$ and change at time instants $\tau_k$ such that either $R_i(\tau_k) = 0$ with $R_i(\tau_k) > 0$ or $A_i > BP_i(s(\tau_k))$ with $R_i(\tau_k) = 0$. Moreover, the agent positions $s_n(t) = [s_n^x(t), s_n^y(t)]$, $n = 1, \ldots, N$, on an parametric trajectories are expressed using (4.1). Viewed as a hybrid system, we can now concentrate on all events causing transitions in the dynamics of $R_i(t), i = 1, \ldots, M$, since any other event has no effect on the values of $\nabla R_i(Y_1, \ldots, Y_N, t)$ at $t = \tau_k$.

First, if $R_i(t) = 0$ and $A_i BP_i(s(t)) < 0$, applying (2.27) to $R_i(t)$ and using (3.3) gives $\frac{d}{dt} \frac{\partial R_i(t)}{\partial Y_n} = 0$. When $R_i(t) > 0$, we have

$$\frac{d}{dt} \frac{\partial R_i(t)}{\partial Y_n} = B \frac{\partial p_n(\omega_i, s_n(t))}{\partial Y_n} \prod_{d \neq n} [1 - p_d(\omega_i, s_d(t))]$$

Adopting the sensing model (3.2) and noting that $p_n(\omega_i, s_n(t)) = p_n(D(\omega_i, s_n(t)))$, where $D(\omega_i, s_n(t)) = [(s_n^x(t) - \alpha_i)^2 + (s_n^y(t) - \beta_i)^2]^{1/2}$ we get

$$\frac{\partial p_n(\omega_i, s_n(t))}{\partial Y_n}$$

$$= \frac{1}{2D(\omega_i, s_n(t))} \left( \frac{\partial D(\omega_i, s_n(t))}{\partial s_n^x} \frac{\partial s_n^x}{\partial Y_n} + \frac{\partial D(\omega_i, s_n(t))}{\partial s_n^y} \frac{\partial s_n^y}{\partial Y_n} \right)$$

where $\frac{\partial D(\omega_i, s_n(t))}{\partial s_n^x} = 2(s_n^x - \alpha_i)$ and $\frac{\partial D(\omega_i, s_n(t))}{\partial s_n^y} = 2(s_n^y - \beta_i)$. From (4.1), we obtain $\frac{\partial s_n^x}{\partial Y_n}$ and $\frac{\partial s_n^y}{\partial Y_n}$ as follows:

$$\begin{cases}
\frac{\partial s_n^x}{\partial Y_n} = \frac{\partial f(T_n, \theta_n(t))}{\partial Y_n} \\
\frac{\partial s_n^y}{\partial Y_n} = \frac{\partial g(T_n, \theta_n(t))}{\partial Y_n}
\end{cases}$$
assuming that \( f(\cdot) \) and \( g(\cdot) \) are selected to be continuously differentiable functions. Using (4.7) and (4.6) in (4.5), we can finally obtain \( \frac{\partial R_i(t)}{\partial Y_n} \) for \( t \in [\tau_k, \tau_{k+1}) \) as

\[
\frac{\partial R_i(t)}{\partial Y_n} = \frac{\partial R_i(\tau_k^+)}{\partial Y_n} + \begin{cases} 
0 & \text{if } R_i(t) = 0, \\
\int_{\tau_k}^t \frac{d}{dt} \frac{\partial R_i(t)}{\partial Y_n} dt & \text{otherwise}
\end{cases}
\]

(4.8)

Thus, it remains to determine the components of \( \nabla R_i(\tau_k^+) \) appearing in (4.8) using (2.28). This involves the gradient of the event times \( \nabla \tau_k = \frac{\partial \tau_k}{\partial Y_n} \) for \( k = 1, \ldots, K \), which will be determined through (2.29). There are two possible cases regarding the events that cause switches in the dynamics of \( R_i(t) \):

**Case 1:** At \( \tau_k \), \( \dot{R}_i(t) \) switches from \( \dot{R}_i(t) = 0 \) to \( \dot{R}_i(t) = A_i \ BP_i(s(t)) \). In this case, it is easy to see that the dynamics \( R_i(t) \) are continuous, so that \( f_k(\tau_k) = f_k(\tau_k^+) \) in (2.28) applied to \( R_i(t) \) and we get

\[
\nabla R_i(\tau_k^+) = \nabla R_i(\tau_k), \quad i = 1, \ldots, M
\]

(4.9)

**Case 2:** At \( \tau_k \), \( \dot{R}_i(t) \) switches from \( \dot{R}_i(t) = A_i \ BP_i(s(t)) \) to \( \dot{R}_i(t) = 0 \), i.e., \( R_i(\tau_k) \) becomes zero. In this case, we need to first evaluate \( \nabla \tau_k \) from (2.29) in order to determine \( \nabla R_i(\tau_k^+) \) through (2.28). Observing that this event is endogenous, (2.29) applies with \( g_k = R_i = 0 \) and we get \( \nabla \tau_k = \frac{\nabla R_i(\tau_k)}{A_i \ BP_i(s(\tau_k))} \). It follows from (2.28) that

\[
\nabla R_i(\tau_k^+) = \nabla R_i(\tau_k) \left[ \frac{A_i}{BP_i(s(\tau_k))} \right] \frac{\nabla R_i(\tau_k)}{A_i \ BP_i(s(\tau_k))} = 0
\]

(4.10)

Thus, \( \nabla R_i(\tau_k^+) \) is always reset to 0 regardless of \( \nabla R_i(\tau_k) \).

**Objective Function Gradient Evaluation.** Based on our analysis, we first rewrite \( J \) in (4.4) as

\[
J(\tau_1, \ldots, \tau_N) = \sum_{i=1}^{M} \sum_{k=0}^{K} \int_{\tau_k}^{\tau_{k+1}} R_i(\tau_1, \ldots, \tau_N, t) dt
\]
and (omitting some function arguments) we get

$$\nabla J = \sum_{i=1}^{M} \sum_{k=0}^{K} \left( \int_{\tau_k}^{\tau_{k+1}} \nabla R_i(t) \, dt + R_i(\tau_{k+1}) \nabla \tau_{k+1} - R_i(\tau_k) \nabla \tau_k \right)$$

Observing the cancelation of all terms of the form $R_i(\tau_k) \nabla \tau_k$ for all $k$ (with $\tau_0 = 0$, $\tau_{K+1} = T$ fixed), we finally get

$$\nabla J(T_1, \ldots, Y_N) = \sum_{i=1}^{M} \sum_{k=0}^{K} \int_{\tau_k}^{\tau_{k+1}} \nabla R_i(t) \, dt$$

(4.11)

This depends entirely on $\nabla R_i(t)$, which is obtained from (4.8) and the event times $\tau_k$, $k = 1, \ldots, K$, given initial conditions $s_n(0)$ for $n = 1, \ldots, N$, and $R_i(0)$ for $i = 1, \ldots, M$. In (4.8), $\frac{\partial R_i(\tau_k^+)}{\partial \tau_n}$ is obtained through (4.9)-(4.10), whereas $\frac{d}{dt} \frac{\partial R_i(t)}{\partial \tau_n}$ is obtained through (4.5)-(4.6). Note that this approach is scalable in the number of events defined by any point where some $R_i(t)$ switches dynamics based on (3.3).

**Objective Function Optimization.** We now seek to obtain $[Y_1^*, \ldots, Y_N^*]$ minimizing $J(Y_1, \ldots, Y_N)$ through a standard gradient-based optimization algorithm of the form

$$[Y_1^{l+1}, \ldots, Y_N^{l+1}] = [Y_1^l, \ldots, Y_N^l] \quad \eta^l \hat{\nabla} J(Y_1, \ldots, Y_N)$$

(4.12)

where $\{\eta^l\}$, $l = 1, 2, \ldots$ are appropriate step size sequences and $\hat{\nabla} J(Y_1, \ldots, Y_N)$ is the projection of the gradient $\nabla J(Y_1, \ldots, Y_N)$ onto the feasible set, i.e., $s_n(t) \in \Omega$ for all $n$, $t \in [0, T]$. The algorithm terminates when $|\hat{\nabla} J(Y_1, \ldots, Y_N)| < \varepsilon$ (for a fixed threshold $\varepsilon$) for some $[Y_1^*, \ldots, Y_N^*]$.

**4.2 Lissajous and Fourier Series Representations of Agent Trajectories**

In (Lin and Cassandras, 2013), we proved that elliptical trajectories outperform linear ones in terms of their respective persistent monitoring performance in 2-dimensional spaces: they achieve lower cost using the average uncertainty metric per unit of area as the basis for the comparison. In this section, we consider a much broader family of trajectory representations by appropriately selecting $f(\cdot)$ and $g(\cdot)$ in the parametric equations (4.1).
Lissajous functions. The motivation for this family of functions (often referred to as Lissajous curves) (Cundy and Rollett, 1989) comes from the fact that in the 1-dimensional space, the optimal agent trajectories are oscillators with agents moving between a set of switching points \( \{\theta_1, \ldots, \theta_K\} \) and possibly dwelling at each such point with waiting time \( \{w_1, \ldots, w_K\} \). Lissajous curves are known to describe complex harmonic oscillatory motion in a 2-dimensional space, which it is reasonable to postulate as the analog of the 1-dimensional case.

For a Lissajous trajectory, the \( n \)th agent movement is described by

\[
\begin{align*}
\dot{s}_n^x(t) &= a_n \sin(\theta_n \rho_n(t) + \phi_n) + X_n \\
\dot{s}_n^y(t) &= b_n \sin(\rho_n(t)) + Y_n
\end{align*}
\]

where \([X_n, Y_n]\) is the center of the \( n \)th Lissajous curve, \( a_n \) and \( b_n \) control the horizontal and vertical sizes of the trajectory, \( \theta_n \in [0, \pi) \) controls the specific shape of the curve, and \( \phi_n \) is the phase difference. For \( \theta_n = 1 \), the curve reduces to an ellipse, with special cases including circles \( (a_n = b_n, \phi_n = \pi/2) \) and straight lines \( (\phi_n = 0) \). Another simple case of a Lissajous curve is a "figure 8" when \( \theta_n = 0.5 \) and \( \phi_n = 0 \). Other ratios produce more complicated curves, which are closed only if \( \theta_n \) is rational. Note that the parameter \( \rho_n(t) \in [0, 2\pi) \) in (4.13) represents the agent position along the trajectory. Since we require the agent to move with \( u_n^* = 1 \) on this trajectory, from (4.3), we have

\[
\rho_n(t) = \left[ a_n^2 \dot{\theta}_n^2 \cos^2(\theta_n \rho_n(t) + \phi_n) + b_n^2 \cos^2 \rho_n(t) \right]^{1/2}
\]

Lissajous functions as in (4.13) may be chosen to define the parametric equations \( f(\cdot) \) and \( g(\cdot) \) in (4.1). The controllable parameter \( \Upsilon_n \) is defined as \( \Upsilon_n = [X_n, Y_n, a_n, b_n, \theta_n, \phi_n] \). Applying (4.7) to the functions in (4.13) and noting that \( \frac{\partial s_n^x}{\partial Y_n} \) etc.,
and \( \frac{\partial s_n^n}{\partial \gamma_n} = \left[ \frac{\partial s_n^n}{\partial X_n}, \frac{\partial s_n^n}{\partial Y_n}, \frac{\partial s_n^n}{\partial \alpha_n}, \frac{\partial s_n^n}{\partial \beta_n}, \frac{\partial s_n^n}{\partial \phi_n}, \frac{\partial s_n^n}{\partial \theta_n} \right]^T \), we obtain

\[
\begin{align*}
\frac{\partial s_n^n}{\partial X_n} &= 1, \quad \frac{\partial s_n^n}{\partial Y_n} = 0 \\
\frac{\partial s_n^n}{\partial \alpha_n} &= \sin(\theta_n \rho_n(t) + \phi_n), \quad \frac{\partial s_n^n}{\partial \beta_n} = 0 \\
\frac{\partial s_n^n}{\partial \phi_n} &= a_n \rho_n \cos(\theta_n \rho_n + \phi_n) \\
\frac{\partial s_n^n}{\partial \theta_n} &= a_n \cos(\theta_n \rho_n + \phi_n)
\end{align*}
\]

Similarly, for \( \frac{\partial s_n^n}{\partial \gamma_n} \)

\[
\begin{align*}
\frac{\partial s_n^n}{\partial X_n} &= 0, \quad \frac{\partial s_n^n}{\partial Y_n} = 1 \\
\frac{\partial s_n^n}{\partial \alpha_n} &= 0, \quad \frac{\partial s_n^n}{\partial \beta_n} = \sin \rho_n(t) \\
\frac{\partial s_n^n}{\partial \phi_n} &= 0, \quad \frac{\partial s_n^n}{\partial \theta_n} = 0
\end{align*}
\]

Using these expressions for \( \frac{\partial s_n^n}{\partial \gamma_n} \) and \( \frac{\partial s_n^n}{\partial \gamma_n} \) in (4.7) and the result in (4.6) and then in (4.5), allows us to evaluate (4.8)-(4.10) and finally obtain \( \nabla R_i(t) \) in (4.11).

**Fourier Series.** In (4.13), when \( \theta_n \) is rational, Lissajous curves are closed and periodic. They are well-suited to cover a rectangular space with equally weighted sampling points (same \( A_i \) and \( R_i(0) \) for all \( i \)) that are uniformly distributed in the space. However, if the distribution of sampling points is nonuniform and irregular shapes are involved, Lissajous curves may not work as well. Instead, we turn to Fourier series trajectory representations which are known to approximate any periodic curves of arbitrary shape by decomposing the periodic function into the sum of simple sinusoid functions. In the frequency domain, a Fourier series provides a broader base than Lissajous functions which include a single frequency in contrast to Fourier series representations which contain multiple frequencies with different amplitudes. Thus, when (i) the shape of the mission space is irregular, (ii) the distribution of sampling points is nonuniform or (iii) the sampling points are not equally weighted (different \( A_i \) or \( R_i(0) \) for some \( i \)), a Fourier series trajectory representation is much more attractive.
For a Fourier series trajectory, the \( n \)th agent movement is described by

\[
\begin{align*}
\dot{s}_n^x(t) &= a_{n,0}^x + \sum_{\gamma=1}^{\frac{x}{n}} a_{n,\gamma}^x \sin \left( 2\pi \gamma f_n^x \rho_n(t) + \phi_{n,\gamma}^x \right) + \dot{\rho}_n(t) \\
\dot{s}_n^y(t) &= a_{n,0}^y + \sum_{\gamma=1}^{\frac{y}{n}} a_{n,\gamma}^y \sin \left( 2\pi \gamma f_n^y \rho_n(t) + \phi_{n,\gamma}^y \right)
\end{align*}
\]  

(4.15)

In the frequency domain, \( f_n^x \) and \( f_n^y \) are the base frequencies, \( a_{n,0}^x, a_{n,0}^y \) are the zero frequency components, \( a_{n,\gamma}^x, a_{n,\gamma}^y \) are the amplitudes for the sinusoid functions with frequency \( \gamma f_n^x \) and \( \gamma f_n^y \), and \( \phi_{n,\gamma}^x, \phi_{n,\gamma}^y \) are the phase differences with respect to the \((\gamma + 1)\)th term for \( s_n^x \) and \( s_n^y \), \( \gamma = 1, \ldots, \frac{x}{n} \) or \( \frac{y}{n} \). Note that the absolute values of \( f_n^x \) and \( f_n^y \) do not matter, since only the ratio \( \frac{f_n^x}{f_n^y} \) determines the shape of the trajectories. Therefore, we only need to parameterize \( f_n^x \) and keep \( f_n^y \) fixed. There are totally \( \frac{x}{n} + 1 \) and \( \frac{y}{n} + 1 \) terms in the Fourier series summation representations for \( s_n^x \) and \( s_n^y \). When \( \frac{x}{n} = \frac{y}{n} = 1 \) and \( \phi_{n,\gamma}^y = 0 \), the Fourier series trajectory reduces to a Lissajous trajectory with \( \vartheta_n = \frac{f_n^x}{f_n^y} \). In essence, a Fourier series representation serves as a broad family of trajectories to be chosen, especially appealing when the mission space is complicated as described above.

Note that the parameter \( \rho_n(t) \in [0, 2\pi) \) represents the agent position along the trajectory. Since, as before, we require the agent to move with \( u_n^s = 1 \) on this trajectory, from (4.3), we have

\[
\dot{\rho}_n(t) = \frac{1}{2\pi} \left[ \left( \frac{f_n^x}{2\pi} \sum_{\gamma=1}^{\frac{x}{n}} a_{n,\gamma}^x \gamma \cos \left( 2\pi \gamma f_n^x \rho_n(t) + \phi_{n,\gamma}^x \right) \right)^2 \right]^{1/2} \\
+ \left( \frac{f_n^y}{2\pi} \sum_{\gamma=1}^{\frac{y}{n}} a_{n,\gamma}^y \gamma \cos \left( 2\pi \gamma f_n^y \rho_n(t) + \phi_{n,\gamma}^y \right) \right)^2
\]

(4.16)

Functions expressed as Fourier series in (4.15) may be chosen to define the parametric equations \( f(\cdot) \) and \( g(\cdot) \) in (4.1). The amplitude parameter vectors for \( s_n^x \) and \( s_n^y \) are \( A_n^x = [a_{n,0}^x, a_{n,1}^x, \ldots, a_{n,\frac{x}{n}}^x]^T \) and \( A_n^y = [a_{n,0}^y, a_{n,1}^y, \ldots, a_{n,\frac{y}{n}}^y]^T \). The phase difference parameter vectors for \( s_n^x \) and \( s_n^y \) are \( \Phi_n^x = [\phi_{n,1}^x, \phi_{n,2}^x, \ldots, \phi_{n,\frac{x}{n}}^x]^T \) and \( \Phi_n^y = [\phi_{n,1}^y, \phi_{n,2}^y, \ldots, \phi_{n,\frac{y}{n}}^y]^T \). Then, the controllable parameter vector \( \Upsilon_n \) is defined as \( \Upsilon_n = [A_n^x, A_n^y, f_n^x, \Phi_n^x, \Phi_n^y]^T \).
Applying (4.7) to (4.15) and noting that 

\[ \frac{\partial s_n^x}{\partial a_n, \gamma} = \begin{cases} 1, & \text{if } \gamma = 0 \\ \sin (2\pi \gamma f_n \rho_n(t) + \phi_n^x), & \text{otherwise} \end{cases} \]  

(4.17)

for \( \gamma = 0, \ldots, \frac{x}{n} \), and 

\[ \frac{\partial s_n^x}{\partial A_n} = 0 \]  

(4.18)

For \( \frac{\partial s_n^x}{\partial A_n} \) and \( \frac{\partial s_n^y}{\partial A_n} \), we have 

\[ \frac{\partial s_n^y}{\partial A_n} = 0 \]  

(4.19)

and 

\[ \frac{\partial s_n^y}{\partial a_n, \gamma} = \begin{cases} 1, & \text{if } \gamma = 0 \\ \sin (2\pi \gamma f_n \rho_n(t) + \phi_n^y), & \text{otherwise} \end{cases} \]  

(4.20)

for \( \gamma = 0, \ldots, \frac{y}{n} \). For \( \frac{\partial s_n^x}{\partial f_n} \) and \( \frac{\partial s_n^y}{\partial f_n} \), we have 

\[ \frac{\partial s_n^x}{\partial f_n} = 2\pi \rho_n(t) \sum_{\gamma=1}^{\frac{x}{n}} a_n, \gamma \cos (2\pi \gamma f_n \rho_n(t) + \phi_n^x) \]  

(4.21)

and 

\[ \frac{\partial s_n^y}{\partial f_n} = 0 \]  

(4.22)

For \( \frac{\partial s_n^x}{\partial \Phi_n} \) and \( \frac{\partial s_n^y}{\partial \Phi_n} \), we have 

\[ \frac{\partial s_n^x}{\partial \Phi_n} = a_n, \gamma \cos (2\pi \gamma f_n \rho_n(t) + \phi_n^x) \]  

(4.23)

for \( \gamma = 1, \ldots, \frac{x}{n} \), and 

\[ \frac{\partial s_n^y}{\partial \Phi_n} = 0 \]  

(4.24)

For \( \frac{\partial s_n^x}{\partial \Phi_n} \) and \( \frac{\partial s_n^y}{\partial \Phi_n} \), we have 

\[ \frac{\partial s_n^x}{\partial \Phi_n} = 0 \]  

(4.25)

and 

\[ \frac{\partial s_n^y}{\partial \Phi_n} = a_n, \gamma \cos (2\pi \gamma f_n \rho_n(t) + \phi_n^y) \]  

(4.26)

for \( \gamma = 1, \ldots, \frac{y}{n} \). Using these expressions for \( \frac{\partial s_n^x}{\partial Y_n} \) and \( \frac{\partial s_n^y}{\partial Y_n} \) in (4.7) and the result in (4.6)
and then in (4.5), allows us to evaluate (4.8)-(4.10) and finally obtain $\nabla R_i(t)$ in (4.11).

4.3 Numerical Experiments

We provide a two-agent example for which we solve P4.1 using both Lissajous functions and Fourier series representations for the agent trajectories. The results are shown Figs. 4.1-4.2 respectively. The same two-agent example is also considered using elliptical trajectories (Lin and Cassandras, 2013) and a TPBVP solver (Lin and Cassandras, 2013) with results shown in Fig. 4.3-4.4 for comparison purposes. In all cases the same environment settings are used: $r = 3, L_1 = 20, L_2 = 10, T = 100$. All sampling points $\omega_i = [\alpha_i, \beta_i]$ are uniformly spaced within $L_1 \times L_2, i = 1, \ldots, M$ and the distance between adjacent sampling points is 2. Note that $M = \frac{L_1}{2} \times \frac{L_2}{2} = 50$. Initial values are $R_i(0) = 30$ and $B = 6, A_i = 0.6$ for all $i = 1, \ldots, M$.

In Figs. 4.1-4.4, the top left plots show the initial trajectories assigned to the two agents for Lissajous, Fourier Series, elliptical trajectories and the TPBVP solver respectively. In all cases, initial trajectories are deployed based on randomly assigned parameters. For the Fourier series trajectories in Fig. 4.2, we used $\frac{n}{x} = \frac{n}{y} = 3$, for $n = 1, 2$. The top right plots show the final optimal trajectories for each case, obtained as the solution of P4.1.
Figure 4.2: Fourier Series trajectories. Top left: initial trajectories assigned for two agents. Top right, final trajectories obtained solving Problem P4.1. Bottom: cost vs. number of iterations. $J^* = 1612$.

Figure 4.3: Elliptical trajectories. Top left: initial trajectories assigned for two agents. Top right, final trajectories obtained solving Problem P4.1. Bottom: cost vs. number of iterations. $J^* = 1660$. 
Figure 4.4: Trajectories by solving the TPBVP problem. Top left: initial trajectories assigned for two agents. Top right, final trajectories obtained solving the TPBVP problem. Bottom: cost vs. number of iterations. $J^* = 1575$.

using the gradient-based optimization algorithm in (4.12) and the initial trajectories in the top left plots in each figure. Finally, the bottom plots show the cost as a function of the number of iterations in (4.12) for each parametric case. Note that the initial costs are significantly reduced in all four cases, indicating the importance of optimally selecting the trajectory parameters. In Fig. 4-1, Lissajous trajectories achieve the optimal cost $J^* = 1612$, with the Fourier series trajectories in Fig. 4-2 achieving about the same optimal cost $J^* = 1612$. On the other hand, in Fig. 4-3 we see that elliptical trajectories achieve a higher optimal cost $J^* = 1660$, while in Fig. 4-4, the TPBVP solver achieves the lowest cost $J^* = 1575$ (we note that this may still be a local minimum). A standard TPBVP requires discretizing the mission time $T$ and calculating every control at each time step which is clearly a computationally intensive process. In addition, solving a TPBVP involves forward and backward integrations of the state and costate equations to evaluate the gradient of the cost function with respect to each control. The number of iterations needed for the algorithm to converge (about $2.8 \times 10^5$ iterations) is dramatically higher than the number of iterations needed for the previous three parametric trajectories (about 3 orders of magnitude). This computationally expensive solver scales poorly with the number of agents and the size of the mission space. We point out that, in all four cases, the costs are only locally optimal.
in general; in order to achieve global optimality, we may use the Stochastic Comparison Algorithm, as in (Lin and Cassandras, 2014), based on randomly selected initial settings. As shown in (Bao and Cassandras, 1996), when this algorithm is applied to stochastic systems and under some technical conditions, the Markov process \( \{T_k\} \) generated by the algorithm will converge to an \( \epsilon \) optimal interval of the global optimum for arbitrarily small \( \epsilon > 0 \).

4.4 Summary

We have addressed the problem of determining optimal agent trajectories for persistent monitoring missions in 2-dimensional spaces by representing an agent trajectory in terms of general function families characterized by a set of parameters that we can optimize. We have considered the family of Lissajous functions as well as a Fourier series representation of a trajectory. We have shown that the problem of determining optimal parameters for these trajectories can be solved using IPA to determine gradients of the objective function with respect to these parameters evaluated on line so as to adjust them through a standard gradient-based algorithm. Numerical examples indicate that this scalable approach (which can be used on line) provides solutions that approximate those obtained through a computationally intensive TPBVP solver.
Chapter 5

Receding Horizon Control for Persistent Monitoring Problem

We propose two receding horizon controllers suitable for dynamic and uncertain environments, where off-line optimal control approach may be infeasible. In addition, for persistent monitoring problems for which optimal control approach is too time consuming, receding horizon controller can provide an efficient and near optimal result. The control scheme dynamically determines agent control by solving a sequence of optimal control over a planning horizon and executing them over a shorter action horizon. For the 1-dimensional persistent monitoring problems, since we can show that the system dynamics, cost function and terminal constraints are not explicit function of time, we are able to prove the control strategy is stationary as long as the system states stay on the optimal control trajectory. By carefully design a terminal constraint for the planning horizon, the optimal control strategy is stationary and we are safe, without missing the switching point, to let the system evolve for the action horizon time duration, according to the optimal control calculated during the planning horizon. Receding horizon control can not only greatly simplify the optimal control problem, it can also make the control strategy adapt to the changing environment, since optimal control is an off-line calculation, while receding horizon control is an on-line one.

In a 1-dimensional mission space, it has been shown in Chapter 2 that the optimal solution is for each agent to move at maximal speed and switch direction at specific points, possibly waiting some time at each such point before switching. In a 2-dimensional mission space, such simple solutions can no longer be derived. In Chapter 3, we formulate a 2D persistent monitoring problem as one of determining optimal elliptical trajectories for a given number of agents. Then in Chapter 4, we further study the problem by representing
an agent trajectory in terms of general function families characterized by a set of parameters that we can optimize. It has shown that the problem of determining optimal parameters for these trajectories can be solved using IPA to determine gradients of the objective function with respect to these parameters evaluated on line so as to adjust them through a standard gradient-based algorithm. In particular, we have applied this approach to the family of Lissajous functions as well as a Fourier series representation of an agent trajectory.

As we introduced in Remark 3.2, The IPA approach possesses an inherent robustness property: there is no need to explicitly model how uncertainty affects uncertainty value in (3.3). Consequently, if \( A_i(t), i = 1, \ldots, M \) are unknown but stationary random processes, the gradient manifests \( A_i(t) \) through the event times \( \tau_k, k = 1, \ldots, K \), which are directly observable during the gradient evaluation process. But when \( A_i(t) \) is non-stationary, IPA may not provide a unbiased estimate of the cost function with respect to the controllable parameters. Thus complicating factors include non-stationary uncertainty modeling and the exact uncertainty model may not always be known in advance. In addition, there may be obstacles in the 2-dimensional mission space, which makes our previous parameterized trajectory optimization solution infeasible. In this Chapter, we propose two receding horizon (RH) controllers suitable for dynamic and uncertain environment setting. The control scheme dynamically determines agent trajectories by solving a sequence of optimization problems over a planning horizon and executing them over a shorter action horizon. The receding horizon control (RHC) algorithm can be implemented centralized or decentralized.

Two RH controllers for 1 and 2-dimensional persistent monitoring problem are presented in Section 5.1 and Section 5.2 respectively. In Section 5.3, numerical experiment results are presented with an extra TPBVP example shown as a comparison.

5.1 RHC for 1D Persistent Monitoring Problem

In this section, we provide a receding horizon control algorithm for 1-dimensional multi-agent persistent monitoring problem. It is capable of obtaining a near-optimal solution on-the-fly.
Let us first consider a receding horizon framework for a single agent $s(t)$ with dynamics (2.1) and (2.6), with initial conditions $s(0)$ and $R_i(0), i = 1, \ldots, M$. We denote $x(t) = [s(t), R_1(t), \ldots, R_M(t)]^T$ as the state of the system.

Now we consider the system at time instance $t$, given the state of the system at time $t$ to be $x(t)$. We aim to propose a receding horizon controller with time horizon $[t, t + H]$. We assume that $H$ is not fixed, but rather a control variable. Moreover, we consider the receding horizon controller where there is one switching between current time $t$ and $H$. We first assume that the control switches from 1 to 1 when $s = \theta$. The position of switching $\theta$ is called the switching point. Therefore, we consider the following optimal control problem at time $t$.

$$\min_{\theta, H} J(\theta, H) = \int_t^{t+H} \sum_i R_i(\xi) d\xi$$

under dynamics

$$\dot{s}(t) = \begin{cases} 1 & \text{for } t \in [t, \theta] \\ 1 & \text{for } t \in (\theta, t + H] \end{cases}$$

and (2.6) with initial conditions $x(t)$ and constraint $H \geq T_f$. We denote this optimal control problem with initial condition $x(t)$ at time $t$ as $\text{Prob}^+(t, x(t))$. Next, we explore the stationarity of the optimal control for $\text{Prob}^+(t, x(t))$. Note that this is the free final-time optimal control.

**Proposition 5.1.** Assume $H^*$ and $\theta^*$ are solutions to $\text{Prob}^+(t, x(t))$. Then given $\tau \in [t, H^*], H^*$ and $\theta^*$ are the solutions to $\text{Prob}^+(\tau, x(\tau))$, where $x^*(\tau)$ is the state of the system evolved under dynamics (5.2) and (2.6) with control $\theta^*$.

**Proof.** From (Bryson and Ho, 1975), we have that if the cost function, dynamics, terminal constraint are not function of time explicitly, then the optimal control is a function of the state. We therefore denote the optimal control at state at $x$ as $\theta^*(x)$. Moreover, from Bellman's principle of optimality, $\theta^*(x(t))$ and $H^*(x(t))$ are optimal for any state on the optimal trajectory (i.e., the trajectory of $x^*(\xi), \xi \in [t, H^*]$ evolved under dynamics (5.2), (2.6) and the optimal control $\theta^*$ with initial condition $x^*(t) = x(t)$. In other words, $\theta^*(x(t)) = \theta^*(x^*(\xi))$ for any $\xi \in [t, H^*]$. Therefore $\theta^*$ and $H^*$ are optimal for $\text{Prob}_1(\tau, x(\tau))$. 

for any $\tau \in [t, H^*]$. 

The proof for the stationarity of the optimal switching point for the agent switch from 1 to 1 is exactly the same. We denote the optimal control problem (5.1) under dynamics

$$\dot{s}(t) = \begin{cases} 1 & \text{for } t \in [t, \theta] \\ 1 & \text{for } t \in (\theta, t + H) \end{cases}$$

as $\text{Prob}(t, x(t))$. Prop. 5.1 proves the stationarity of the optimal switching point for $\text{Prob}^+(t, x(t))$ and $\text{Prob}^-(t, x(t))$. Next we show Prop. 5.2 that gives the optimal planning horizon which equals the terminal constraint $T_f$.

**Proposition 5.2.** For $\text{Prob}^+(t, x(t))$ and $\text{Prob}^-(t, x(t))$, $H^* = T_f$.

**Proof.** $\text{Prob}^+(t, x(t))$ and $\text{Prob}^-(t, x(t))$ are defined with terminal constraint $H^* \geq T_f$. $J(\theta, H) = \int_t^{t+H} \sum_i R_i(\xi) d\xi \geq \int_t^{t+T_f} \sum_i R_i(\xi) d\xi$ because $\sum_i R_i(\xi) \geq 0$ for $\xi \in [0, T]$. Thus $H^* = T_f$.  

The determination of the terminal constraint $T_f$ has a tradeoff: if it is too small, the algorithm is inefficient and most of the time, the agent will find the it's optimal to maintain $u(t)$ for $t \in [t, t + T_f]$ and the optimal switching point $\theta^*$ is at the end of the planning horizon; if it is too big, it's likely there are more than one switching point $\theta^*$ within $[t, t + T_f]$. Assuming only one switching point within $[t, t + T_f]$ while there are actually more than one switching points, we would likely miss at least one switching point. In order to reduce the likelihood of missing switching points, we assign the action horizon as the time interval for the agent to go from current position to the switching point: $h = |\theta^* - s(t)|$. After the agent reaches its optimal switching point within $t \in [t, t + T_f]$, it shifts the current time to $t + h$ and resolves the problem. Thus we call $h = |\theta^* - s(t)|$ as the action horizon. It is now clear that the behavior of the agent under the receding horizon control policy is that at the begin of each step, assume there is only one switching point in $[t, t + T_f]$, calculate this switching point position and go directly to that switching point.

For multi-agent 1-dimensional persistent monitoring system, the action horizon for the system can be obtained in the following way. 1) Decide the agent that has the smallest action horizon $k = \arg \min n |\theta_n - s_n(t)|$, for $n = 1, \ldots, N$. and then 2) set the action
horizon for all agent as $h = |\theta_k^* - s_k(t)|$. Thus all agents are synchronized in the sense that they all calculate their optimal switching point within $[t, t + T_f]$ at the same time.

Our analysis thus far has shown that, under receding horizon control, for some $t \in [0, T]$, all agents calculate their optimal switching points $\Theta^* = [\theta_1^*, \ldots, \theta_N^*]$ within $[t, t + T_f]$, choose the action horizon for all agents as the smallest action horizon among all agents $h = |\theta_k^* - s_k(t)|$, where $k = \arg\min_n |\theta_n^* - s_n(t)|$, for $n = 1, \ldots, N$, move at full speed for the action horizon time interval $h$. At the end of the action horizon, the agent determines the smallest action horizon switches its direction and all the other agents maintain their speed. The calculation of $\Theta^* = [\theta_1^*, \ldots, \theta_n^*, \ldots, \theta_N^*]$ can be obtained from Alg. 1, assume $n = 1, n = 1, \ldots, N$ and set $w = 0, \frac{dw}{dt} = 0$.

We present an event-based receding horizon control algorithm for the 1-dimensional multi-agent problem. Assume $\Theta^* = [\theta_1^*, \ldots, \theta_n^*, \ldots, \theta_N^*]$ is a vector composed of the switching points for agent $1, \ldots, N$. Opposite controls either 1 or 1 are assigned to time duration before and after the switching point. Then choose the action horizon as the smallest distance between all agents current positions and the optimal switching points, $h = |\theta_k^* - s_k(t)|$. Apply the default control to this time horizon to all agents, change the direction for the agent that reaches its switching point and maintain all the other agents direction. Repeat this process until termination time $T$.

**Algorithm 5 : Centralized Receding Horizon Controller for the 1-dimensional Persistent Monitoring Problem**

1: Pick $T_f > 0$ and Set $t = 0, u_n = 1, n = 1, \ldots, N$.
2: repeat
3: Compute $\Theta^* = [\theta_1^*, \ldots, \theta_n^*, \ldots, \theta_N^*]$ as $\dot{s}_n(t) = u_n$ for $\tau \in [t, \theta_n]$ and $\dot{s}_n(t) = u_n$ for $\tau \in [\theta_n, t + T_f]$.
4: $k = \arg\min_n |\theta_n - s_n(t)|$, for $n = 1, \ldots, N$. Set the action horizon $h = |\theta_k^* - s_k(t)|$.
5: Set control $u_n = u_n(t)$ for $n = 1, \ldots, N$, for $h$ time units long.
6: Set $t$ to be the time when $s_k = s(\theta_k^*)$.
7: $u_k(t) = u_k(t)$.
8: until $t \geq T$
9: END

All agents are synchronized in the sense that the action horizon for all agents are set to be homogenous as the smallest action horizon. The algorithm is in essence a centralized
RHC controller where there exist a central server to communicate with all agents. The server is responsible for collecting each agent's action horizon, picking the smallest one and distributing the smallest action horizon to all agents. The centralized RHC may not be feasible in the real world, as it requires wireless communication to be reliable. With the increasing importance of energy management in wireless environments, batteries are playing a significant role for autonomous agents' tasks. Wireless communication is extremely energy consuming and we want to minimize unnecessary wireless communication as much as possible. So there is a tradeoff between longer and shorter action horizon $h$. With short $h$, RHC is more prudent and can achieve lower cost, but with a price of high energy consumption in real implementation.

5.2 RHC for 2D Persistent Monitoring Problem

In this section, we discuss receding horizon control for 2-dimensional multi-agent persistent monitoring problem. In principle, one can invoke dynamic programming as a solution approach, but this is computationally intractable even for relatively simple mission control settings, i.e., more agents and longer operation time. Because of the complexity of the overall problem, it is natural to decompose it into various subproblems at different levels. In Chapter 3 and Chapter 4, we simplify the control space by parameterizing agent trajectories using Elliptical, Lissajous trajectories and ones characterized by Fourier series function representation. An alternative to this function decomposition approach is one based on time decomposition. This is aiming developing online controllers suitable for uncertain environment where combinatorially complex assignment algorithms are infeasible.

We have shown in 3.11 that $u^*(t) = 1$ for $n = 1, \ldots, N$, we are only left with the task of determining $\theta^*_n(t) \in [0, 2\pi), n = 1, \ldots, N$. First we discretize the controllable space $[0, 2\pi)$ and let $\theta_n(t)$ take $D$ discrete values $\{0, \frac{1}{D}2\pi, \frac{2}{D}2\pi, \ldots, \frac{D}{D}2\pi\}$. We then define the planning horizon $H$ for all agent as

$$H(\theta) = \min_n [B_n(\theta_n) \cdot s_n(t)] \quad (5.4)$$

where $\theta = [\theta_1, \ldots, \theta_N]$ and $B_n(\theta_n)$ is the point lies on the environment boundary if agents
n moves with heading $\theta_n$. Then $[B_n(\theta_n), s_n(t)]$ is the distance between the agent current position $s_n(t)$ and $B_n(\theta_n)$ under the heading $\theta_n$. We use $H(\theta)$ as the homogeneous planning horizon for all agents such that the centralized RHC is synchronized. We calculate $J(\theta) = \frac{1}{H(\theta)} \int_t^{t+H(\theta)} \sum_i R_i(t) dt$ under current control $\theta$, uncertainty dynamics (3.3).

We set

$$\theta^*(t) = \arg\min_{\theta(t)} J(\theta)$$

(5.5)

and set $H^* = H(\theta^*)$ accordingly. The action horizon $h$ is set to be a fixed small value $h_0$. At each time step, the central server compares all the possible combinational control within the planning horizon $H$ and choose the one that generates the smallest time average cost to implement for action horizon $h_0$. Thus all agents implement their control synchronously.

The algorithm is shown in Alg. 6

**Algorithm 6**: Synchronized Centralized Receding Horizon Control for 2-dimensional Persistent Monitoring Problem

1. Pick $h = h_0$.
2. Discretize each agent to have $D$ headings within $[0, 2\pi]$. Each heading $u_n$ can take value of $0, \frac{1}{D}2\pi, \frac{2}{D}2\pi, \ldots, \frac{D-1}{D}2\pi$.
3. repeat
   4. Given current system states $s_n(t), n = 1, \ldots, N$ and $R_i(t), i = 1, \ldots, M$, define the planning horizon $H$ for each combinatorial control $\theta(t)$ as $H(\theta(t))$. Set $H(\theta(t)) = \min_n [B_n(\theta_n), s_n(t)]$, where $B_n(\theta_n)$ is the point lies on the environment boundary if agents $n$ moves with heading $\theta_n$.
   5. Compute $J(\theta(t)) = \frac{1}{H(\theta(t))} \int_t^{t+H(\theta(t))} \sum_i R_i(t) dt, n = 1, \ldots, N$.
   6. Define $\theta^*(t) = \arg\min_{\theta(t)} J(\theta)$ and set $H^* = H(\theta^*(t))$.
   7. Implement $\theta^*$ for $h_0$ time units and obtained the system states $R_i(t+h_0)$, $i = 1, \ldots, M$ and $s_n(t+h_0), n = 1, \ldots, N$.
   8. Set $t = t + h_0$.
9. until $t \geq T$
10. END

The planning horizon $H(\theta(t))$ is determined as the earliest time when some agent would hit the environment boundary, if all agents maintain the current headings $\theta(t)$. The system becomes an event driven hybrid system where interesting events are defined as some agents reach the environment boundary. We could also define other interesting event times such as the earliest time instant some uncertainty value reaches 0. This implementation requires
a precise knowledge of the uncertainty dynamics for every sampling points. In addition, the central server needs to track all the uncertainty values. These complexities add more difficulty if the central server has limited online computation power.

5.3 Numerical Experiments

5.3.1 RHC for 1-dimensional Mission Space

Here we present a comparison of two numerical examples of 1-dimensional persistent monitoring problem. One experiment is using the receding horizon controller and the other is determined by the optimal controller using IPA algorithm. Here we omit the waiting time parameters $w$ and only consider the switching point $\theta$. These two examples are two-agent experiment with $L = 40, M = 41$ and the remaining sampling points are evenly distributed over the rectangular space. The sensing range is set to $r = 4$, the initial values of the uncertainty functions are $R_i(0) = 3$, for $i = 1, \ldots, M$, and the time horizon is $T = 200$. The increasing speed for the uncertainty value is $A = 0.1$ and the decreasing speed for the uncertainty value is $B = 3$. The initial positions for the two agents in these two examples are $s_1(0) = 10$ and $s_1(0) = 30$. In Fig. 5.1(a) we show two agent trajectories using receding horizon control and we can achieve cost $J = 33.6230$, while in Fig. 5.1(b) we show two agent trajectories using optimal control and we can achieve the optimal cost $J = 32.2895$, which is a little bit lower than the receding horizon control. Thus we have shown that our receding horizon controller can achieve a cost which is very close to the optimal cost, while maintaining computational simplicity.

Noted that the switching points marked by the red circle in Fig. 5.1(a) are not real switching point. The agent stop at this point at the last planning horizon and find the next optimal $\theta^* = s(t)$, meaning the agent changes its speed directly without moving forward using its original speed. The action horizon at these points is zero.
5.3.2 RHC for 2-dimensional Mission Space

Two-agent example of solving the 2-dimensional environment persistent monitoring problem using Alg. 6 are shown in Fig. 5.4. Two two-agent example of solving the 2-dimensional environment persistent monitoring problem using TPBVP Alg. 2 are shown in Fig. 5.2 and Fig. 5.3.

In all three cases the same environment settings are used: \( r = 2, L_1 = 8, L_2 = 8, T = 100 \). All sampling points \([\alpha_i, \beta_i]\) are uniformly spaced within \( L_1 \times L_2, i = 1, \ldots, M \). Note that \( M = 3 \times 3 = 9 \). Initial values are \( R_i(0) = 2 \) and \( B = 6, A_i = 0.2 \) for all \( i = 1, \ldots, M \). Number of headings for each agent are \( D = 12 \).

Different initial trajectories are given for these two TPBVP experiments. In Fig. 5.2, we use diagonal trajectories as the initial input, while in Fig. 5.3, we use elliptical trajectories as the initial input for the TPBVP algorithms. The two TPBVP with different initial trajectories achieve lower cost compared to cost obtained from the RHC Alg. 6. The TPBVP solution is easy to fall into local optima when the number of sampling points are small. It simply misses and doesn’t even realize the existence of some sampling points. Note
that in Fig. 5.2, the final optimal trajectory misses the left bottom sampling point the up right sampling point. In Fig. 5.3, the final optimal trajectory misses the middle sampling point. The optimal cost for Fig. 5.2 is $J^* = 536$ and for Fig. 5.3 is $J^* = 521$. Again, TPBVP algorithm for one or two cases converge to the local minimum.

A two-agent example of solving the 2-dimensional environment persistent monitoring problem using Alg. 6 and are shown in Fig. 5.4. Number of headings for each agent are $D = 12$. in Alg. 6. The action horizon for $h$ is set to be $h = \frac{H}{3}$. Thus the action horizon $h$ grows proportionally with $H$. When $H$ is small, increasing $H$ gives agents the ability to avoid myopic behavior and thus achieves lower cost. But when $H$ is big, the resulting big $h$ pushes agents to be less prudent. We observe higher cost for increasing $H$. The cost using Alg. 6 is $J^* = 654$, which is a little higher than the cost achieved using TPBVP. Alg. 6 achieves near optimal result.
Figure 5.3: TPBVP results. Upper plot: initial trajectory. Bottom plot: optimal trajectory. $J^* = 521$.

Figure 5.4: RHC trajectories using Alg 6. $J^* = 654$. 
5.4 Summary

We have proposed two receding horizon controllers which are capable of determining agent trajectories for persistent monitoring missions in 1 and 2-dimensional mission space. For the 1-dimensional problem, we have proved that stationarity of the optimal switching point for the free final-time optimal control problem. Moreover, we have shown that the optimal free final time equals the terminal time constraint. Based on these two proofs, we have presented a centralized RH controller, where the planning horizon is fixed and determined by the terminal time constraint and the action horizon is determined by the smallest distance between all agents and their corresponding switching points. For the 2-dimensional mission space, we have proposed a centralized RH controller, where the planning horizon is obtained as the smallest distance between all agents and mission space boundary and action horizon is fixed. We have presented numerical experiments to show that these two RH controllers can obtain near optimal results on the fly.
Chapter 6
Conclusions and Future Work

We have investigated persistent monitoring problem in 1 and 2-dimensional mission space, using both optimal control framework and near optimal RHC to solve these two problems. For the optimal control framework in both cases, the idea is to transfer the optimal control problems into parameterized optimization in hybrid systems. For RHC, we want to find the optimal planning horizon and action horizon.

First, we have formulated an optimal persistent monitoring problem with the objective of controlling the movement of multiple cooperating agents to minimize an uncertainty metric in a given mission space with and without some upper bound constraints for the uncertainty values in the mission space. In a one-dimensional mission space, we have shown that the optimal solution is reduced to the determination of two parameter vectors for each agent: a sequence of switching locations and associated waiting times at these switching points. We have used Infinitesimal Perturbation Analysis (IPA) to evaluate sensitivities of the objective function with respect to all parameters and, therefore, obtain a complete on-line (locally optimal) solution through a gradient-based algorithm. Observe that the evaluation of ∇J is independent of \( A_i, i = 1, \ldots, M \) and the dependence of ∇J on \( A_i \) manifests itself through the event times \( \tau_k, k = 1, \ldots, K \), we have pointed that the IPA approach possesses an inherent robustness property: there is no need to model how uncertainty increases. Consequently, \( A_i \) can be treated unknown and be modeled as stationary random process \{\( A_i(t) \), \( i = 1, \ldots, M \). For the problem with upper bound constraint, the evaluation of the gradient of the cost function with respect to controllable variables has only one more term, which characterize \( R_i(t) \) enters and leaves the upper bound constraint.

Second, for the 2-dimensional mission space, we have shown that an optimal control
solution to the 1-dimensional persistent monitoring problem does not easily extend to the 2-dimensional case. In particular, we have proved that elliptical trajectories outperform linear ones in a 2-dimensional mission space. When we perturb the minor axis from 0 to a small positive value \( v > 0 \), the derivative of the cost function with respect to the minor axis \( b \) is strictly greater than 0. Therefore, we have sought to solve a parametric optimization problem to determine optimal elliptical trajectories. Numerical examples indicate that this scalable approach (which can be used on line) provides solutions that approximate those obtained through a computationally intensive TPBVP solver. Moreover, since the solutions obtained are generally locally optimal, we have incorporated a stochastic comparison algorithm for deriving globally optimal elliptical trajectories. Note that the persistent monitoring problem becomes a stochastic comparison problem if \( A_i(t), i = 1, \ldots, M \) are stochastic processes. However, for the deterministic setting in which all \( A_i \) are constant, one-time comparison of the cost within iteration of the CSC algorithm is sufficient. Thus, the CSC algorithm in this deterministic setting reduces to a multi-start comparison algorithm.

Third, we further approach the 2-dimensional persistent monitoring problem by representing an agent trajectory in terms of general function families characterized by a set of parameters that we can optimize. We have considered the family of Lissajous functions as well as a Fourier series representation of a trajectory. Motivated by the simple oscillatory optimal trajectory structure in the 1-dimensional problem, we consider Lissajous functions because of their property to systematically describe complex harmonic motion in a 2-dimensional space. Trajectories based on a Fourier series representation, on the other hand, are used to approximate any arbitrary trajectory and are more suitable when the mission space is irregular (i.e., its shape is complex or the weights and distribution of sampling points in the mission space are inhomogeneous). We have shown that the problem of determining optimal parameters for these trajectories can be solved using IPA to determine gradients of the objective function with respect to these parameters evaluated on line so as to adjust them through a standard gradient-based algorithm. Numerical examples indicate
that this scalable approach (which can be used online) provides solutions that approximate those obtained through a computationally intensive TPBVP solver.

Last, we have proposed two receding horizon controllers which are capable of determining agent trajectories for persistent monitoring missions in 1 and 2-dimensional mission space. For the 1-dimensional problem, we have proved that stationarity of the optimal switching point for the free final-time optimal control problem. Moreover, we have shown that the optimal free final time equals the terminal time constraint. Based on these two proofs, we have presented a centralized RH controller, where the planning horizon is fixed and determined by the terminal time constraint and the action horizon is determined by the smallest distance between all agents and their corresponding switching points. For the 2-dimensional mission space, we have proposed a centralized RH controller, where the planning horizon is obtained as the smallest distance between all agents and mission space boundary and action horizon is fixed. We have presented numerical experiment to show that these two RH controllers can obtain near optimal results on the fly.
References


CURRICULUM VITAE

Xuchao Lin

Education
Boston University, Boston, MA, USA
M.S. and Ph.D in Systems Engineering, Sep 2009 - Sep 2014
- Adviser: Prof. Christos G. Cassandras
Zhejiang University, Hangzhou, China
B.Eng. in Electrical Engineering, Sep 2005 - Jun 2009
The University of Hong Kong, Hong Kong, China
Exchange Student in Electrical and Electronics Engineering, Sep 2007 - Jun 2008

Research Interests

Professional Experiences
Senior Consultant May 2014 - Present
Altair Management Consultants Corp., Boston, MA
- Retail Banking Pricing Optimization
Research Assistant May 2010 - Apr 2014
Advisor: Prof. Christos G. Cassandras, Control of Discrete Event Systems (CODES) Laboratory, Boston University, Boston, MA
- Optimal Control and Infinitesimal Perturbation Analysis (IPA) for the Persistent Monitoring Problem
  Presented an optimal control framework for persistent monitoring problems. Utilized Infinitesimal Perturbation Analysis (IPA) to obtain a complete on-line solution through a gradient-based algorithm for the Hybrid system.
Awards

1. Travel Award (IEEE Conference on Decision and Control) 2013
2. Dean's Fellowship 2009
3. Graduated with Highest Distinction in Zhejiang Province 2009
4. Graduated with Highest Distinction in Zhejiang University 2009
5. China National Scholarship (< 1%) 2009
6. First Class Honors (3%) 2009
7. Guosong Wang Fellowship (1%) 2008
8. Li & Fung Scholarship 2007
9. Xinshi Fellowship (5%) 2006

Journal Papers


Peer-reviewed Conference Proceedings


Workshop Presentations and Posters
