Adaptive Sampling and Forecasting With Mobile Sensor Networks

by

Han-Lim Choi
B.S., Korea Advanced Institute of Science and Technology (2000)
M.S., Korea Advanced Institute of Science and Technology (2002)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
February 2009

Author .....
Department of Aeronautics and Astronautics
September 30, 2008

Certified by .....
Jonathan P. How
Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by .....
Nicholas Roy
Assistant Professor of Aeronautics and Astronautics

Certified by .....
James A. Hansen
Lead Scientist, Naval Research Laboratory

Certified by .....
Emilio Frazzoli
Associate Professor of Aeronautics and Astronautics

Accepted by .....
David L. Darmofal
Chairman, Department Committee on Graduate Students

Adaptive Sampling and Forecasting With Mobile Sensor Networks

by

Han-Lim Choi

Submitted to the Department of Aeronautics and Astronautics
on September 30, 2008, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis addresses planning of mobile sensor networks to extract the best information possible out of the environment to improve the (ensemble) forecast at some verification region in the future. To define the information reward associated with sensing paths, the *mutual information* is adopted to represent the influence of the measurement actions on the reduction of the uncertainty in the verification variables. The sensor networks planning problems are posed in both discrete and continuous time/space, each of which represents a different level of abstraction of the decision space.

In the discrete setting, the *targeting* problem is formulated to determine the sequence of information-rich waypoints for mobile sensors. A *backward formulation* is developed to efficiently quantify the information rewards in this combinatorial decision process. This approach computes the reward of each possible sensing choice by propagating the information backwards from the verification time/space to the search space/time. It is shown that this backward method provides an equivalent solution to a standard forward approach, while only requiring the calculation of a *single* covariance update. This work proves that the backward approach works significantly faster than the forward approach for the ensemble-based representation.

In the continuous setting, the *motion planning* problem that finds the best steering commands of the sensor platforms is posed. The main difficulty in this continuous decision lies in the quantification the mutual information between the *future* verification variables and a *continuous history* of the measurement. This work proposes the *smoother form* of the mutual information inspired by the conditional independence relations, and demonstrates its advantages over a simple extension of the state-of-the-art: (a) it does not require integration of differential equations for long time intervals, (b) it allows for the calculation of accumulated information on-the-fly, and (c) it provides a legitimate information potential field combined with spatial interpolation techniques.

The primary benefits of the presented methods are confirmed with numerical experiments using the Lorenz-2003 idealistic chaos model.

Thesis Supervisor: Jonathan P. How
Title: Professor of Aeronautics and Astronautics

Acknowledgments

First of all, I would like to thank my advisor Prof. Jonathan How for letting me to explore this interesting research and for being a model of a good/hard-working advisor himself. I am thankful to Prof. Nicholas Roy for the series of informative comments and inputs he has provided in the biweekly DDDAS meetings. I appreciate Dr. Jim Hansen for guiding me to the weather world and explaining me all the jargons I was unfamiliar with. Many thanks to Prof. Emilio Frazzoli for insightful comments that helped me set a firm standpoint to the big picture of my work. I also thank my thesis readers, Prof. Karen Willcox and Prof. Hamsa Balakrishnan, for providing me good sources of inputs to improve the presentation of the work in this thesis.

I would like to thank my former advisor Prof. Min-Jea Tahk at KAIST for his invaluable advices on both research and a real life. I thank my former advisor at MIT, Prof. Eric Feron now in Georgia Tech, who helped me easily settle down at MIT. I appreciate Prof. Paul Barton in Chemical Engineering at MIT for his insightful inputs on the optimization work in the appendix of my thesis. I am grateful to Prof. Jean-Jacques Slotine for his help in my individual research on the nonlinear observer design.

Many of my fellow labmates in the Aerospace Controls Laboratory have been a good source of ideas; especially, I would like to thank Luca Bertuccelli and Emily Craparo for all the critical (cross-)readings and insightful comments on my papers. The help of Kathryn Fischer in organizing the trips (adapted to my preference) should be acknowledged.

I am grateful to my friends including Korean community in MIT Aero/Astro, KAISTians at MIT, and KSHS Class 12 in Korea. Especially, I would like to thank Sang-il Lee and Jaemyung Ahn who started MIT Aero/Astro lives at the same time, and the five best friends of mine, "Eagle Five."

I especially want to thank my family in Korea: my parents who have supported and believed in me for 30 years, my brother Yun-Lim who has been my best friend and a model of the active, my parents-in-law who already started calling me Dr. Choi three years ago and have provided endless support, my brother-in-law Changbum who has so believed in me, and my sister-in-laws Dajeong and Anna.

Most importantly, I would like to thank my wife Chorong, the source of every pleasure and happiness of mine. Without her unbelievable love, encouragement, and patience, I could not make it to this point – I love you. Finally, thanks our baby boy, Robin.

This research was funded by NSF CNS-0540331 as part of the DDDAS program. The author also gratefully acknowledges funding from the Samsung Scholarship.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis presents new algorithms for planning of mobile sensor networks to extract the best information out of the environment. The design objective of maximizing some *information reward* is addressed in both discrete and continuous time and space in order to represent different levels of abstraction in the decision space. The sensor *targeting* problem that locates a sequence of information-rich waypoints for mobile sensors is formulated in the discrete domain; the *motion planning* problem that determines the best steering commands of the sensor platforms is posed in the continuous setup. Key technological aspects taken into account are: definition and effective quantification of the information reward, and design mechanism for vehicle paths considering the mobility constraints.

This work addresses the above planning problems in the context of *adaptive sampling* for numerical weather prediction. The goal of adaptive sampling is to determine when and where to take supplementary measurements to improve the weather forecast in the specified verification region, given a fixed observation network (See Figure 1-1). The complex characteristics of the weather dynamics, such as being chaotic, uncertain, and of multiple time- and length-scales, requires the use of a large sensor network [1–5]. Expanding the static observation network is limited by geographic aspects; thus, an adaptive sensor network incorporating mobile sensor platforms (manned or possibly unmanned) has become an attractive solution to effectively construct larger networks.

The primary challenges in this adaptive sampling are:

- Spatial and temporal coupling of the impact of sensing on the uncertainty reduction: Since a decision by one sensing agent affects the information reward for decisions by other sensing agents, the selection of the best set of measurements should take into account these coupling effects. Due to this type of coupling, the adaptive sampling problem is **NP**-hard [6, 7].

- Enormous size of systems: A typical state dimension of the weather system is $\mathcal{O}(10^6)$ and a typical measurement dimension of the sensor networks is $\mathcal{O}(10^3)$ [4]; therefore, the adaptive sampling is a very large-scale combinatorial decision making.

- Long forecast horizon: The quantity of interest in the adaptive sampling problem is the weather variables over the verification region in the far future; thus, computation of the impact of sensing on the verification region typically incurs a computationally expensive forward (in time) inference procedure. Therefore, a solution method should feature an effective forward inference.

- Nonlinearity in dynamics: The weather system is highly nonlinear; therefore, incorporation of nonlinear estimation schemes are essential in adaptive sampling.

- Interest in a specified subset of the entire state space: The fact that the verification region is a small subset of a large state space brings difficulty in the adoption of well-established techniques [6, 8–10] for the selection problems with submodular reward functions.

Previously, there were two representative types of adaptive sampling approaches: One approach locates the sites where a perturbation from them propagated through a linearized dynamics exhibits the largest growth of errors in the verification region [1–3]. This approach characterizes the direction of error growth embedded in the dynamics, but cannot take into account specific details of the data assimilation scheme such as the uncertainty level of the state estimates. The other approach quantified

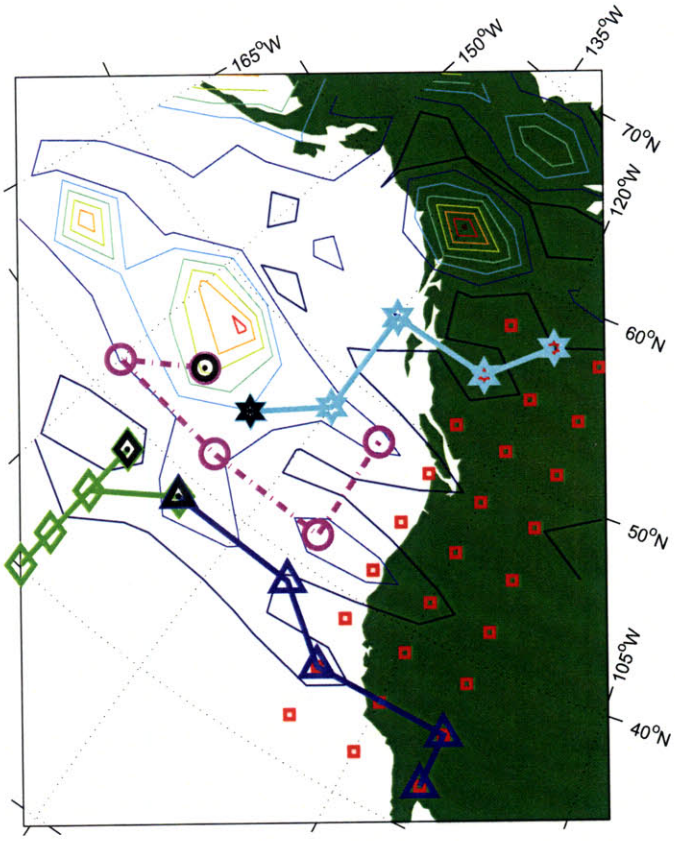

Figure 1-1: Adaptive sampling for weather forecast improvement: Paths for four UAV sensor platforms (cyan ★, green ◊, blue △, magenta ◯) over a 12hr mission horizon starting at black markers are selected to improve the weather forecast over California (red □) in 3 days.

the variation in the future forecast error based on an ensemble approximation of the extended Kalman filter [4, 5], and sought for the sensing points that reduce the trace of the covariance matrix of the future verification variables. This type of approach can incorporate the uncertainty information into the sampling decision. However, due to the challenges listed above, only simple selection strategy has been tractable. For instance, two flight paths were *greedily* selected out of 49 predetermined paths in [4].

This thesis addresses these challenges by considering two levels of decision making – the targeting problem and the motion planning problem. The targeting problem posed in the discrete time/space domain concerns a longer time- and length-scale decision for sensor platforms where the nonlinear weather dynamics is represented by an ensemble forecast system. The motion planning problem posed in the contin-

uous time/space domain deals with a shorter time- and length-scale decision when the linear system approximates a short-term local behavior of the original nonlinear dynamics. This hierarchical approach enables better incorporation of the multi-scale aspect of the weather system with providing better computational efficiency. Moreover, this work takes an information-theoretic standpoint to define and quantify the impact of a measurement on the uncertainty reduction of the verification variables, taking advantage of some important properties of the mutual information that facilitate computationally efficient decision making.

1.1 Information Rewards

1.1.1 Discrete-Time Mutual Information [Chapter 3]

The notion of *mutual information* has been frequently [6, 8–14] used to define the information reward for discrete selection, since it represents the degree of dependency between the sensor selection and the quantity of interest. For the problems of tracking (moving) targets, the mutual information between the target states and the measurement sequence has been used to represent the improvement of the estimation performance by the measurement selection [10, 12–14]. In addition, in [6, 8, 9, 11], mutual information was used to quantify the uncertainty reduction of the temperature field over the unobserved region by the sensor selection. Since the use of mutual information usually incurs substantial computational cost, all the algorithms have been developed in ways that either approximate the mutual information using other simple heuristics [12] or introduce some greediness in the selection decision [6, 8–11, 14].

Relevant approaches to finding the globally optimal solution to a mathematically equivalent problem to sensor selection were addressed in the context of environmental monitoring. Anstreicher et al. [15] presented a 0-1 nonlinear program called a maximum-entropy remote sampling problem and proposed a branch-and-bound algorithm to solve this problem with global optimality. Although the authors of that paper did not employ the same terminology, they used linear algebraic properties to

derive a formulation that is equivalent to the sensor selection based on commutativity of mutual information. But the problem sizes considered in [15] were much smaller than the ones usually encountered in a targeting problem in a large-scale sensor network. The observation that the global optimal solutions to the sensor selection decisions have been available only for small-size problems points out the importance of effective computation of the information reward in designing a large-scale sensor network.

This thesis provides an efficient way of quantifying the mutual information to be used for a large-scale sensor targeting problem. Mutual information is used as a measure of uncertainty reduction and it is computed from the ensemble approximation of the covariance matrices. The commutativity of mutual information is exploited to address the computational challenge resulting from the expense of determining the impact of each measurement choice on the uncertainty reduction in the verification site. This enables the contribution of each measurement choice to be computed by propagating information *backwards* from the verification space/time to the search space/time. This backward computation is shown to significantly reduce the required number of ensemble updates, which is a computationally intensive part of the adaptive planning process, and therefore leads to much faster decision making. In this thesis, theoretical computation time analysis proves the computational benefit of the backward approach, which is also verified by numerical simulations using an idealized weather model.

1.1.2 Continuous-Time Mutual Information [Chapter 5]

Although *mutual information* in the continuous-time domain does not appear to be as well-known in the robotics and sensor networks communities, in the information theory context, there has been a long history of research about the mutual information between the signal and observation in the continuous-time domain. Duncan [16] showed that the mutual information between the signal history and observation history (i.e. signal during $[0, t]$ and observation during $[0, t]$) is expressed as a function of estimation error for a Gaussian signal through an additive Gaussian channel. Similar

quantification is performed for non-Gaussian signal [17, 18] and fractional Gaussian channel [19]. On the other hand, Tomita et al. [20] showed that the optimal filter for a linear system that maximizes the mutual information between the observation history for $[0, t]$ and the state value at t is the Kalman-Bucy filter; Mayer-Wolf and Zakai [21] related this mutual information to the Fisher information matrix.

Recently, Mitter and Newton [22] presented an expression for the mutual information between the signal path during $[s, t]$, $s < t$ and the observation history, with statistical mechanical interpretation of this expression. Newton [23, 24] extended his previous results by quantifying the mutual information between the future signal path during $[t, T]$, $t \leq T$ and the past measurement history during $[0, t]$ for linear time-varying [23] and nonlinear [24] systems. However, it should be noted that all these previous quantifications have been about the *state* and *observation*. On the contrary, this study deals with the mutual information between the values of a *subset of the state* at T and the observation history during $[0, t]$ when $T > t$.

As an effective mechanism to quantify the mutual information for forecast problems, this thesis proposes the *smoother form*, which regards forecasting as fixed-interval smoothing. Based on conditional independence of the measurement history and the future verification variables for a given present state value, the smoother form is proven to be equivalent to the filter form that can be derived as a simple extension of previous work [20–22]. In contrast to the filter form, the smoother form can avoid integration of matrix differential equations for long time intervals, resulting in better computational efficiency. Moreover, since the smoother form pre-processes the effect of the verification variables and the future process noise, the information accumulated along the path and the rate of information accumulation can be computed on the fly with representing the *pure* impact of sensing on the uncertainty reduction. This facilitates more adaptive planning based on some information potential field.

1.2 Path Planning

1.2.1 Discrete Targeting [Chapter 4]

Discrete path planning of sensor platforms is based on the abstraction of a measurement path as a sequence of waypoints [14, 25–27]. Then, the information reward gathered along a path is represented by the mutual information between a finite number of sensing points distributed over time and a finite number of future verification variables. While the backward formulation enables computation of this mutual information value, the primary question in the discrete path planning is how to incorporate the constrained vehicle motions in an effective manner, and how to maintain computational tractability and the performance level in large-scale decisions.

In this thesis, the aspect of the vehicle mobility constraint is dealt with by using the concept of the *reachable search space* and the *action space search*. These two allow for reduction of the search space dimension by excluding inadmissible solution candidates. Also, a cut-off heuristics utilizing a simplistic approximation of the mutual information provides further computational efficiency.

To handle computational burden in the optimal targeting of multiple sensing platforms, decomposition schemes that break down a large-scale targeting problem into a series of smaller problems are presented. The comparative study of different decomposition topologies indicates the crucialness of the coordinated modification of covariance information to achieve good overall team performance.

1.2.2 Continuous Motion Planning [Chapter 5]

The spatially continuous feature of the vehicle path can be addressed by using spatial interpolation techniques such as Kriging [28] and Gaussian Processes Regression [29] that predict a value of the quantity of interest at an arbitrary point in continuous space as a function of values at a finite number of grid points. Using this technique, the measurement action along a continuous path can be modeled as evolution of the observation function – the observation matrix in a linear system, which appears in

the integrand of the matrix differential equation for the smoother-form information reward. Thus, continuous variation of the vehicle location will change the integrand of the matrix differential equation through the observation matrix, and this ultimately leads to the change in the information reward value.

Utilizing this representation of continuous paths, many standard path planning methods are explored. The optimal path planning and a real-time gradient-ascent steering based on information potential fields are presented.

1.3 Implementation Issues

1.3.1 Sensitivity Analysis of Targeting [Chapter 6]

In real weather prediction systems, due to the computational expense of integrating forward a large-scale nonlinear system, and storing large ensemble data sets, the ensemble size that can be used for adaptive sampling is very limited [30–32]. In spite of limitation of ensemble diversity in real weather prediction systems, there have been little studies on the sensitivity of the ensemble-based adaptive sampling to the ensemble size. As an essential step toward real implementation of the presented targeting mechanism, this thesis performs a sensitivity analysis from both experimental and theoretical points of view. Monte-Carlo experiments characterize important sensitivity properties anticipated with small ensemble size, and analytical formulas that relate the ensemble size and the expected performance variation are derived based on the statistics theory for sample entropy estimation.

1.4 Summary of Contributions

- **Discrete-Time Mutual Information** A similar backward formulation based on commutativity of mutual information was first presented in [15] in the process of formulating a 0-1 program, and previous work [10, 14] dealing with the task of tracking a moving target recognized the convenience and efficiency it gives in representing and calculating the associated covariance information. However, a

key contribution of this thesis is to clearly show the computational advantages of the backward form caused by the reduction of the number of covariance updates, by expanding the benefits in [10] to much more general sensor selection problems. In addition, with two key examples, this thesis shows that the backward selection performs never slower than the forward selection and also clearly identifies the types of systems for which the backward offers a substantial computational improvement over the forward. This newly-identified advantage of the backward selection, which is amplified for large-scale decisions, was not addressed in previous work [10, 14], and is a unique contributions of this thesis.

- **Continuous-Time Mutual Information** Despite well-established theory on mutual information between the states and observations in the continuous-time domain, it is a unique and novel contribution of this thesis to derive the expressions for the forecast problem. In particular, the smoother form is proposed to well account for distinctive characteristics of forecasting, i.e., interest in a particular subset of the states in the far future. Also, this work clearly identifies the advantages of the smoother form over the filter form in the aspects of computational efficiency, robustness to modeling errors, and concurrent knowledge of information accumulation.

- **Multi-Sensor Platform Targeting** On the backbone of the backward selection formulation, this work presents the overall algorithmic procedure for multi-sensor platform targeting. In the process, techniques offering further efficiency are proposed, which reduce the dimension of the search space and the number of calculations of information rewards. Moreover, the importance of coordinated information sharing is clearly pointed out based on numerical studies on various decomposition strategies.

- **Information Potential Field and Continuous Motion Planning** While most previous information-driven path planning studies have dealt with discrete decisions, the concept of information potential field, and continuous motion planning based on that concept, was proposed in [33, 34]. This thesis offers a

similar information potential field for the forecast problem by deriving a formula for the rate of change of the smoother form information reward. Despite similarity to the previous work [33, 34], the ability of the smoother form to figure out the pure impact of sensing on the uncertainty reduction of the variables of ultimate interest, enables construction of the true map of information distribution that takes into account the diffusion of information through the future process noise.

- **Sensitivity Analysis** There have been studies to figure out and mitigate the effect of small ensemble size on the quality of forecasting and data assimilation [30, 31]; however, no intensive research has addressed the sensitivity analysis for the adaptive sampling problems. A contribution of this thesis is to quantify the performance variation with respect to the ensemble size with both numerical experiments and theoretical analysis.

Chapter 2

Preliminaries

2.1 Entropy and Mutual Information

The entropy that represents the amount of information hidden in a random variable is defined as [35]

$$\begin{aligned}\mathcal{H}(A_1) &= -\mathbb{E}\left[\log\left(f_{A_1}(a_1)\right)\right] \\ &= \begin{cases} -\sum\limits_i f_{A_1}(a_1^i)\log(f_{A_1}(a_1^i)), & \text{for a discrete random variable } A_1 \\ -\int_{-\infty}^{\infty} f_{A_1}(a_1)\log(f_{A_1}(a_1))da_1, & \text{for a continuous random variable } A_1 \end{cases}\end{aligned} \tag{2.1}$$

where $\mathbb{E}[\cdot]$ denotes expectation, and $f_{A_1}(a_1)$ represents the probability mass function (pmf) for a discrete A_1 (i.e. $\texttt{Prob}(A_1 = a_1)$), or the probability density function (pdf) for a continuous A_1 (i.e. $\frac{d}{da_1}\texttt{Prob}(A_1 \leq a_1)$). The joint entropy of two random variables is defined in a similar way:

$$\begin{aligned}\mathcal{H}(A_1, A_2) &= -\mathbb{E}\left[\log\left(f_{A_1,A_2}(a_1, a_2)\right)\right] \\ &= \begin{cases} -\sum\limits_i \sum\limits_j f_{A_1,A_2}(a_1^i, a_2^j)\log(f_{A_1,A_2}(a_1^i, a_2^j)), & \text{for discrete } A_1, A_2 \\ -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{A_1,A_2}(a_1, a_2)\log(f_{A_1,A_2}(a_1, a_2))da_1\, da_2, & \text{for continuous } A_1, A_2 \end{cases}\end{aligned} \tag{2.2}$$

with $f_{A_1,A_2}(a_1, a_2)$ denoting the joint pmf or pdf of A_1 and A_2. This definition can be extended to the case of a random vector consisting of multiple random variables. Specifically, if a random vector is Gaussian, its entropy is expressed in terms of the covariance matrix as

$$\mathcal{H}(A) = \tfrac{1}{2}\log\det(\mathbf{P}(A)) + \tfrac{|A|}{2}\log(2\pi e) \tag{2.3}$$

The notation $\mathbf{P}(A)$ is defined as $\mathbf{P}(A) \triangleq \mathbb{E}\left[(A - \mathbb{E}[A])(A - \mathbb{E}[A])'\right]$ where $'$ denotes the transpose of a matrix, and another notation $\mathbf{P}(A, B)$ will be used in this thesis to represent $\mathbf{P}(A, B) \triangleq \mathbb{E}\left[(A - \mathbb{E}[A])(B - \mathbb{E}[B])'\right]$; $|A|$ denotes the cardinality of A, and e is the base of natural logarithm.

The conditional entropy is defined as

$$\begin{aligned}
&\mathcal{H}(A_1|A_2) \\
&\quad = -\mathbb{E}_{f_{A_1,A_2}}\left[\log\left(f_{A_1|A_2}(a_1|a_2)\right)\right] \\
&\quad = \begin{cases} -\sum_i \sum_j f_{A_1,A_2}(a_1^i, a_2^j)\log(f_{A_1|A_2}(a_1^i|a_2^j)), & \text{for discrete } A_1, A_2 \\ -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{A_1,A_2}(a_1, a_2)\log(f_{A_1|A_2}(a_1|a_2))da_1\, da_2, & \text{for continuous } A_1, A_2, \end{cases}
\end{aligned} \tag{2.4}$$

and the joint entropy and the conditional entropy satisfy the following condition:

$$\begin{aligned}
\mathcal{H}(A_1, A_2) &= \mathcal{H}(A_1) + \mathcal{H}(A_2|A_1) \\
&= \mathcal{H}(A_2) + \mathcal{H}(A_1|A_2).
\end{aligned} \tag{2.5}$$

Namely, the joint entropy of two random variables comprises the entropy of one random variable and the conditional entropy of the other conditioned on the first. In case of multiple random variables (i.e. a random vector), the relation in (2.5) can be

extended as

$$\begin{aligned}
&\mathcal{H}(A_1, A_2, \ldots, A_k) \\
&\quad = \mathcal{H}(A_1) + \mathcal{H}(A_2|A_1) + \mathcal{H}(A_3|A_1, A_2) + \cdots + \mathcal{H}(A_k|A_1, \ldots, A_{k-1}), \\
&\quad = \cdots = \mathcal{H}(A_{i_1}) + \mathcal{H}(A_{i_1}|A_{i_2}) + \mathcal{H}(A_{i_3}|A_{i_1}, A_{i_2}) + \cdots + \mathcal{H}(A_{i_k}|A_{i_1}, \ldots, A_{i_{k-1}}),
\end{aligned} \tag{2.6}$$

where $k = |A|$ and $\{i_1, \ldots, i_k\}$ represents some possible permutation of $\{1, \ldots, k\}$.

The mutual information represents the amount of information contained in one random variable (A_2) about the other random variable (A_1), and can be interpreted as the entropy reduction of A_1 by conditioning on A_2:

$$\mathcal{I}(A_1; A_2) = \mathcal{H}(A_1) - \mathcal{H}(A_1|A_2). \tag{2.7}$$

It is important to note that the mutual information is commutative [35]:

$$\begin{aligned}
\mathcal{I}(A_1; A_2) &= \mathcal{H}(A_1) - \mathcal{H}(A_1|A_2) \\
&= \mathcal{H}(A_1) - (\mathcal{H}(A_1, A_2) - \mathcal{H}(A_2)) \\
&= \mathcal{H}(A_1) - (\mathcal{H}(A_1) + \mathcal{H}(A_2|A_1) - \mathcal{H}(A_2)) \\
&= \mathcal{H}(A_2) - \mathcal{H}(A_2|A_1) \\
&= \mathcal{I}(A_2; A_1).
\end{aligned} \tag{2.8}$$

In other words, the entropy reduction of A_1 by knowledge of A_2 is the same as the entropy reduction of A_2 by knowledge of A_1. Because of this symmetry, the mutual information can also be interpreted as the degree of dependency between two random variables. The conditional mutual information can be defined in a similar way and the commutativity holds as well:

$$\begin{aligned}
\mathcal{I}(A_1; A_2|A_3) &= \mathcal{H}(A_1|A_3) - \mathcal{H}(A_1|A_2, A_3) \\
&= \mathcal{H}(A_2|A_3) - \mathcal{H}(A_2|A_1, A_3) \\
&= \mathcal{I}(A_2; A_1|A_3).
\end{aligned} \tag{2.9}$$

The mutual information between two random vectors $A \triangleq \{A_1, A_2, \ldots, A_k\}$ and $B \triangleq \{B_1, B_2, \ldots, B_l\}$ is defined the same way, and it can be expressed in terms of individual random variables as follows

$$\begin{aligned} \mathcal{I}(A;B) &= \mathcal{I}(A_{i_1};B) + \mathcal{I}(A_{i_2};B|A_{i_1}) + \ldots \mathcal{I}(A_{i_k};B|A_{i_1},\ldots,A_{i_k-1}) \\ &= \mathcal{I}(B_{j_1};A) + \mathcal{I}(B_{j_2};A|B_{j_1}) + \ldots \mathcal{I}(B_{j_l};A|B_{j_1},\ldots,B_{j_l-1}) \end{aligned} \tag{2.10}$$

where and $\{i_1, \ldots, i_k\}$ and $\{j_1, \ldots, j_l\}$ represent possible permutations of $\{1, \ldots, k\}$ and $\{1, \ldots, l\}$, respectively. Using (2.3), the mutual information two random vectors that are jointly Gaussian can be written as

$$\begin{aligned} \mathcal{I}(A;B) \equiv \mathcal{I}(B;A) &= \tfrac{1}{2}\log\det\mathbf{P}(A) - \tfrac{1}{2}\log\det\mathbf{P}(A|B) \\ &= \tfrac{1}{2}\log\det\mathbf{P}(B) - \tfrac{1}{2}\log\det\mathbf{P}(B|A) \end{aligned} \tag{2.11}$$

where $\mathbf{P}(A|B) \triangleq \mathbb{E}[(A - \mathbb{E}[A])(A - \mathbb{E}[A])'|B]$. In this work, the above expression of mutual information will be utilized to represent the uncertainty reduction of verification variables by a finite number of measurements taken by sensor platforms (in Chapter 3, 4, and 6). In case the measurement is taken in a continuous fashion, the uncertainty reduction of the verification variables by a continuous measurement history is the mutual information between a random vector and a *random process.* However, some reformulation based on conditional independence allows for utilizing the expressions in (2.11) for quantification of this type of mutual information as well. Details on this point will be presented in Chapter 5.

2.2 Ensemble Square-Root Filter

In this thesis, the environmental (e.g. weather) variables are tracked by an ensemble forecast system, specifically, the sequential ensemble square-root filter (EnSRF) [36]. Ensemble-based forecasts better represent the nonlinear features of the weather system, and mitigate the computational burden of linearizing the nonlinear dynamics and keeping track of a large covariance matrix [36, 37], compared to the extended

Kalman filter. In an ensemble forecast system, the ensemble matrix $\mathbf{X}$ consists of sample state vectors:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1, & \mathbf{x}_2, & \dots & \mathbf{x}_{L_E} \end{bmatrix} \in \mathbb{R}^{L_X \times L_E} \tag{2.12}$$

where $\mathbf{x}_i \in \mathbb{R}^{L_X}$ represents i-th sample state vector. L_X and L_E denote the number of state variables and the ensemble members. EnSRF carries this ensemble matrix to track the dynamics of the environmental variables. In EnSRF, the state estimate and the estimation error covariance are represented by the ensemble mean and the perturbation ensemble matrix, respectively. The ensemble mean is defined as

$$\bar{\mathbf{x}} \triangleq \frac{1}{L_E} \sum_{i=1}^{L_E} \mathbf{x}_i, \tag{2.13}$$

and the perturbation ensemble matrix is written as

$$\widetilde{\mathbf{X}} \triangleq c_0 \left(\mathbf{X} - \bar{\mathbf{x}} \otimes \mathbf{1}'_{L_E} \right) = c_0 \begin{bmatrix} \mathbf{x}_1 - \bar{\mathbf{x}}, & \mathbf{x}_2 - \bar{\mathbf{x}}, & \dots & \mathbf{x}_{L_E} - \bar{\mathbf{x}} \end{bmatrix} \tag{2.14}$$

where $\otimes$ denotes the Kronecker product, $\mathbf{1}_{L_E}$ is the L_E-dimensional column vector every entry of which is unity, and c_0 is an inflation factor used to avoid underestimation of the covariance [36]. Then, the estimation error covariance is approximated as

$$\mathbf{P}(\mathbf{x}_{\text{true}} - \bar{\mathbf{x}}) \approx \frac{1}{L_E - 1} \widetilde{\mathbf{X}} \widetilde{\mathbf{X}}' \tag{2.15}$$

using the perturbation ensemble matrix. EnSRF works in two steps: the prediction step that propagates the state estimates through the nonlinear dynamics, and the measurement update step that performs Bayesian inference based on a linearized observation under the Gaussian assumption. The prediction step corresponds to the integration

$$\mathbf{X}^f(t + \Delta t) = \mathbf{X}^a(t) + \int_t^{t+\Delta t} \dot{\mathbf{X}} \, dt \tag{2.16}$$

where the superscripts 'f' and 'a' denote the *forecast* and the *analysis*, which are equivalent to *predicted* and *updated* in the Kalman-filter terminology, respectively.

The measurement update for the EnSRF consists of the mean update and the perturbation ensemble update as:

$$\bar{\mathbf{x}}^a = \bar{\mathbf{x}}^f + K_g(\mathbf{z} - H\bar{\mathbf{x}}^f),$$
$$\widetilde{\mathbf{X}}^a = (I - K_g H)\widetilde{\mathbf{X}}^f,$$

where $\mathbf{z}$ and H are the measurement vector and the Jacobian matrix of the observation function – a functional relation between the state and the noise-free measurement; K_g represents the Kalman gain determined by a nonlinear matrix equation of $\mathbf{X}^f$ [36].

In the sequential framework devised for efficient implementation, the ensemble update by the m-th observation is written as

$$\widetilde{\mathbf{X}}^{m+1} = \widetilde{\mathbf{X}}^m - \frac{c_1 c_2}{L_E - 1}\widetilde{\mathbf{X}}^m \xi_i' \xi_i \tag{2.17}$$

with $c_1 = (1 + \sqrt{c_2 R_i})^{-1}, c_2 = (p_{ii} + R_i)^{-1}$, when the i-th state variable is directly measured (i.e, H is the i-th unit row vector) and the sensing noise variance is R_i. ξ_i is the i-th row of $\widetilde{\mathbf{X}}^m$ and $p_{ii} = \xi_i \xi_i'/(L_E - 1)$. c_1 is the factor for compensating the mismatch between the serial update and the batch update, and $c_2 \widetilde{\mathbf{X}}^m \xi_i'$ is equivalent to the Kalman gain.

2.3 Lorenz-2003 Chaos Model

The methodologies presented in this work are numerically validated using the Lorenz-2003 model [38], which is an idealized chaos model that addresses the multi-scale feature of the weather dynamics as well as the basic aspects of weather system, such as energy dissipation, advection, and external forcing that are captured in its precursor, Lorenz-95 model [3]. As such, the Lorenz models (including -95 and -2003) have been successfully implemented for the initial verification of numerical weather prediction algorithms [3, 5]. In this thesis, the original one-dimensional model presented in [3, 38] is extended to two-dimensions to represent the global dynamics of the mid-latitude region (20–70 deg) of the northern hemisphere [32].

The system equations are

$$\begin{aligned}\dot{\phi}_{ij} = &- \zeta_{i-2\rho_x,j}\zeta_{i-\rho_x,j} + \frac{1}{2\lfloor\rho_x/2\rfloor+1}\sum_{k=-\lfloor\rho_x/2\rfloor}^{k=+\lfloor\rho_x/2\rfloor}\zeta_{i-\rho_x+k,j}\phi_{i+\rho_x+k,j} \\ &- \tfrac{2}{3}\eta_{i,j-2\rho_y}\eta_{i,j-\rho_y} + \frac{2/3}{2\lfloor\rho_y/2\rfloor+1}\sum_{k=-\lfloor\rho_y/2\rfloor}^{k=+\lfloor\rho_y/2\rfloor}\eta_{i,j-\rho_y+k}\phi_{i,i+\rho_y+k} \\ &- \phi_{ij} + \phi_0\end{aligned} \tag{2.18}$$

where

$$\zeta_{ij} = \frac{1}{2\lfloor\rho_x/2\rfloor+1}\sum_{k=-\lfloor\rho_x/2\rfloor}^{k=+\lfloor\rho_x/2\rfloor}\phi_{i+k,j}, \quad i = 1,\ldots,L_{on} \tag{2.19}$$

$$\eta_{ij} = \frac{1}{2\lfloor\rho_y/2\rfloor+1}\sum_{k=-\lfloor\rho_y/2\rfloor}^{k=+\lfloor\rho_y/2\rfloor}\phi_{i,j+k}, \quad j = 1,\ldots,L_{at}. \tag{2.20}$$

ϕ_{ij} represents a scalar meteorological quantity, such as vorticity or temperature [3], at the (i,j)-th grid point. i and j are longitudinal and latitudinal grid indices, respectively. The dynamics of the (i,j)-th grid point depends on its longitudinal $2\rho_x$-interval neighbors (and latitudinal $2\rho_y$) through the advection terms, on itself by the dissipation term, and on the external forcing ($\phi_0 = 8$ in this work). In case $\rho_x = \rho_y = 1$, the model reduces to the two-dimension Lorenz-95 model.

There are $L_{on} = 36\rho_x$ longitudinal and $L_{at} = 8\rho_y + 1$ latitudinal grid points. In order to model the mid-latitude area as an annulus, the dynamics in (2.18) are subject to cyclic boundary conditions in the longitudinal direction [32]:

$$\phi_{i+L_{on},j} = \phi_{i-L_{on},j} = \phi_{i,j}, \tag{2.21}$$

and constant advection conditions in the latitudinal direction: i.e., in advection terms,

$$\phi_{i,0} = \cdots = \phi_{i,-\lfloor\rho_y/2\rfloor} = 3; \quad \phi_{i,L_{at}+1} = \cdots = \phi_{i,L_{at}+\lfloor\rho_y/2\rfloor} = 0. \tag{2.22}$$

The length-scale of the Lorenz-2003 is proportional to the inverse of ρ_x and ρ_y in each direction: for instance, the grid size for $\rho_x = \rho_y = 2$ amounts to 347 km $\times$ 347 km.

The time-scale of the Lorenz-2003 system is such that 1 time unit is equivalent to 5 days in real time.

Chapter 3

Sensing Point Targeting

This chapter presents an efficient approach to observation targeting problems that are complicated by a combinatorial number of targeting choices and a large system state dimension. The approach incorporates an ensemble prediction to ensure that the measurements are chosen to enhance the forecast at a separate verification time and location, where the *mutual information* between the verification variables and the measurement variables defines the information reward for each targeting choice. The primary improvements in the efficiency are obtained by computing the impact of each possible measurement choice on the uncertainty reduction over the verification site *backwards*. The commutativity of mutual information enables an analysis of the entropy reduction of the measurement variables by knowledge of verification variables, instead of looking at the change in the entropy of the verification variables. It is shown that this backward method provides the same solution to a traditional forward approach under some standard assumptions, while only requiring the calculation of a single ensemble update.

The results in this chapter show that the backward approach works significantly faster than the forward approach for the ensemble-based representation for which the cost of each ensemble update is substantial, and that it is never slower than the forward one, even for the conventional covariance representation. This assessment is done using analytic estimates of the computation time improvement that are verified with numerical simulations using an idealized weather model. These results are then

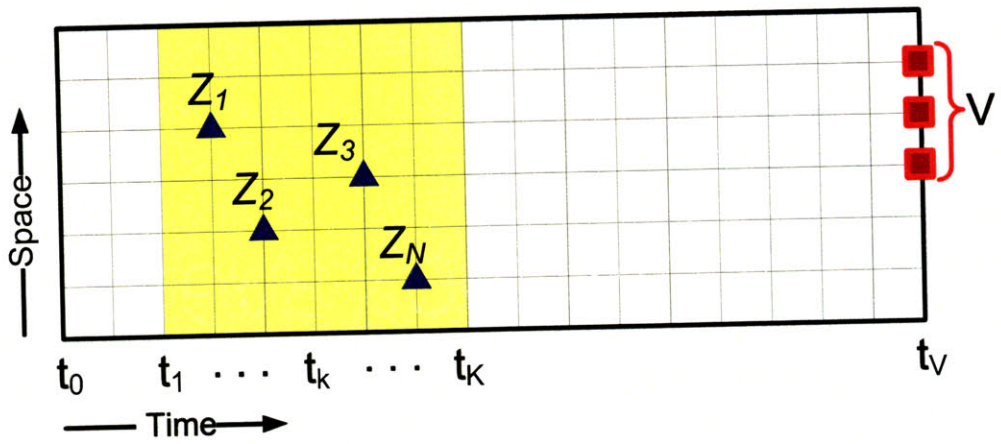

Figure 3-1: Sensor targeting over space and time: decision of deployment of sensors over the search space-time to reduce the uncertainty about the verification region. (N: size of search space, M: size of verification region, n: number of sensing points)

compared with a much smaller-scale sensor selection problem, which is a simplified version of the problem in [10]. This comparison clearly identifies the types of systems for which the backward quantification offers a significant computational improvement over the forward one.

3.1 Sensor Selection Problems

Figure 3-1 illustrates the sensor targeting problem in a spatial-temporal grid space. The objective of sensor targeting is to deploy n sensors in the search space/time (yellow region) in order to reduce the uncertainty in the verification region (red squares) at the verification time t_V. Without loss of generality, it is assumed that each grid point is associated with a single state variable that can be directly measured. Denote the state variable at location s as X_s, and the measurement of X_s as Z_s, both of which are random variables. Also, define $X_{\mathcal{S}} \equiv \{X_1, X_2, \ldots, X_N\}$ and $Z_{\mathcal{S}} \equiv \{Z_1, Z_2, \ldots, Z_N\}$ the sets of all corresponding random variables over the entire search space of size N. Likewise, $V \equiv \{V_1, V_2, \ldots, V_M\}$ denotes the set of random variables representing the states in the verification region at t_V, with M being the size of verification region. With a slight abuse of notation, this thesis does not distinguish a set of

random variables from the random vector constituted by the corresponding random variables. Measurement at location s is subject to an additive Gaussian noise that is uncorrelated with noise at any other location as well as with any of the state variables:

$$Z_s = X_s + W_s, \quad \forall\, s \in \mathcal{S} \triangleq [1, N] \cap \mathbb{Z} \tag{3.1}$$

where $W_s \sim \mathcal{N}(0, R_s)$, and

$$\begin{aligned} \mathbf{P}(W_s, W_p) &= 0, \quad \forall\, p \in \mathcal{S} \setminus \{s\} \\ \mathbf{P}(W_s, Y_p) &= 0, \quad \forall\, Y_p \in X_{\mathcal{S}} \cup V. \end{aligned} \tag{3.2}$$

As mentioned in section 2.1, $\mathbf{P}(A, B) \triangleq \mathbb{E}\left[(A - \mathbb{E}[A])(B - \mathbb{E}[B])'\right]$, and $\mathbf{P}(A) \equiv \mathbf{P}(A, A)$. This work assumes that the distribution of $X_{\mathcal{S}} \cup V$ is jointly Gaussian, or, in a more relaxed sense, that the entropy of any set $Y \subset (X_{\mathcal{S}} \cup V)$ can be well approximated as:

$$\mathcal{H}(Y) = \tfrac{1}{2} \log\det(\mathbf{P}(Y)) + \tfrac{|Y|}{2} \log(2\pi e). \tag{3.3}$$

Also, the covariance matrix $\mathbf{P}(X_{\mathcal{S}} \cup V)$ is assumed to be known in advance of making sensor targeting decisions.

The uncertainty metric in this study is entropy; uncertainty reduction over the verification region is the difference between the unconditioned entropy and the conditioned (on the measurement selection) entropy of V. Thus, the selection problem of choosing n grid points from the search space that will give the greatest reduction in the entropy of V can be posed as:

Forward Selection (FS)

$$\begin{aligned} \mathbf{s}_F^\star &= \arg\max_{\mathbf{s} \in \mathbb{S}_n} \mathcal{I}(V; Z_{\mathbf{s}}) \equiv \mathcal{H}(V) - \mathcal{H}(V|Z_{\mathbf{s}}) \\ &= \arg\max_{\mathbf{s} \in \mathbb{S}_n} \tfrac{1}{2} \log\det \mathbf{P}(V) - \tfrac{1}{2} \log\det \mathbf{P}(V|Z_{\mathbf{s}}) \\ &= \arg\min_{\mathbf{s} \in \mathbb{S}_n} \tfrac{1}{2} \log\det \mathbf{P}(V|Z_{\mathbf{s}}) \end{aligned} \tag{3.4}$$

where $\mathbb{S}_n \triangleq \{\mathbf{s} \subset \mathcal{S} : |\mathbf{s}| = n\}$ whose cardinality is $\binom{N}{n}$. Note that $\mathcal{I}(V; Z_{\mathbf{s}})$ is the

mutual information between V and $Z_{\mathbf{s}}$. Since the prior entropy $\mathcal{H}(V)$ is identical over all possible choices of $\mathbf{s}$, the original arg max expression is equivalent to the arg min representation in the last line. Every quantity appearing in (3.4) can be computed from the given covariance information and measurement model. However, the worst-case solution technique to find $\mathbf{s}_F^\star$ requires an exhaustive search over the entire candidate space $\mathbb{S}_n$; therefore, the selection process is subject to combinatorial explosion. Note that N is usually very large for the observation targeting problem for improving weather prediction. Moreover, computing the conditional covariance $\mathbf{P}(V|Z_{\mathbf{s}})$ and its determinant requires a nontrivial computation time. In other words, a combinatorial number of computations, each of which takes a significant amount of time, are required to find the optimal solution using the FS formulation.

Given these computational issues, this thesis suggests an alternative formulation of the selection problem:

Backward Selection (BS)

$$\begin{aligned}
\mathbf{s}_B^\star &= \arg\max_{\mathbf{s}\in\mathbb{S}_n} \mathcal{I}(Z_{\mathbf{s}}; V) \equiv \mathcal{H}(Z_{\mathbf{s}}) - \mathcal{H}(Z_{\mathbf{s}}|V) \\
&= \arg\max_{\mathbf{s}\in\mathbb{S}_n} \tfrac{1}{2}\log\det\mathbf{P}(Z_{\mathbf{s}}) - \tfrac{1}{2}\log\det\mathbf{P}(Z_{\mathbf{s}}|V) \\
&= \arg\max_{\mathbf{s}\in\mathbb{S}_n} \tfrac{1}{2}\log\det\left(\mathbf{P}(X_{\mathbf{s}}) + R_{\mathbf{s}}\right) - \tfrac{1}{2}\log\det\left(\mathbf{P}(X_{\mathbf{s}}|V) + R_{\mathbf{s}}\right).
\end{aligned} \tag{3.5}$$

Instead of looking at the entropy reduction of V by $Z_{\mathbf{s}}$, this backward selection looks at the entropy reduction of $Z_{\mathbf{s}}$ by V; it provides the same solution as FS, since the mutual information is commutative [35]:

Proposition 1. The forward selection and the backward selection create the same solution:

$$\mathbf{s}_F^\star \equiv \mathbf{s}_B^\star, \tag{3.6}$$

since $\mathcal{I}(V; Z_{\mathbf{s}}) \equiv \mathcal{I}(Z_{\mathbf{s}}; V), \ \forall \mathbf{s} \in \mathbb{S}_n$. □

Note that $\mathbf{P}(Z_{\mathbf{s}}|\cdot)$ in (3.5) becomes simply $\mathbf{P}(X_{\mathbf{s}}|\cdot) + R_{\mathbf{s}}$, the first term of which is already embedded in the original covariance structure. This type of simple relation does not exist for $\mathbf{P}(\cdot|Z_{\mathbf{s}})$ and $\mathbf{P}(\cdot|X_{\mathbf{s}})$, which appear in FS. Previous work [10, 14] that

utilized a similar backward concept took advantage of this convenience. However, in the moving target tracking problems addressed in [10, 14], the verification variables are the whole state variables; in other words, $V = X_{\mathcal{S}}$. This leads to $\mathbf{P}(X_{\mathbf{s}}|V) = 0$ with no need of computing the conditional covariances. In contrast, this work mainly considers the case in which $X_{\mathcal{S}}$ and V are disjoint, which requires conditioning processes to compute $\mathbf{P}(X_{\mathbf{s}}|V)$.

Regarding computational scalability, since the worst-case solution technique to find $\mathbf{s}_B^\star$ is still the exhaustive search, BS is also subject to combinatorial explosion. However, the conditional covariance in the BS form can be computed by a single process that scales well with respect to n; BS can be computationally more efficient than FS. More detail on these points are given in section 3.2.

3.1.1 Ensemble-Based Adaptive Observation Targeting

The observation targeting problem for weather prediction concerns improving the weather forecast for the future time t_V broadcast at the nearer future time t_K ($< t_V$), which is called "targeting time" [4], by deploying observation networks from t_1 through t_K. While the *routine* sensor network, whose locations are fixed and known, is assumed to take measurements periodically (every 6 hours in practice), the targeting problem concerns deployment of a supplementary mobile network (see Figure 3-2). What is known a priori at the decision time t_0 is the state estimate and covariance information based on past measurement histories. Since the actual values of the future measurements that affect the actual forecast error at the verification site are not available at t_0, the targeting considers the forecast *uncertainty* reduction (not the forecast error reduction) that can be quantified by estimation error covariance information without relying on the actual values of observations. Thus, the outcome of the targeting is the sensing locations/times that reduce the forecast error the most in the average sense. To address the adaptive observation targeting as a sensor selection problem presented in the previous section, it is necessary to construct the covariance field over the future time based on the a priori knowledge at time t_0. This section presents that procedure in the ensemble-based weather forecasting framework

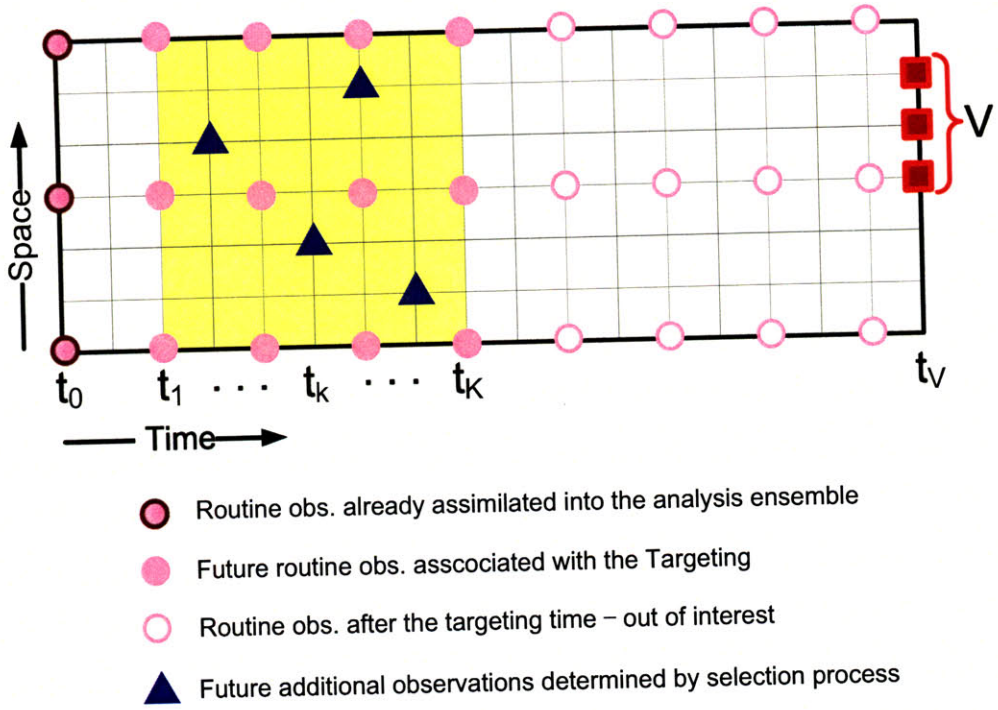

Figure 3-2: Observation structure over the search space and time. The adaptive observation targeting problem decides the best additional sensor network over the time window $[t_1, t_K]$ given the routine network that makes observations over the same time window. The presence of routine observations between $(t_K, t_V]$ is not of a practical concern, because the ultimate interest is in the quality of forecast broadcast at t_K based on the actual measurement taken up to time t_K. [32]

described in section 2.2.

Ensemble Augmentation and Sensor Targeting Formulation

The prior knowledge in the observation targeting problem is the analysis ensemble at t_0. Since the design of the supplementary observation network is conditioned on the presence of the routine network, the covariance field needed for the sensor selection problem is the conditional covariance field conditioned on the impact of future routine observations. Processing the observations distributed over time amounts to an EnSRF ensemble update for the augmented forecast ensemble defined as

$$\mathbf{X}^f_{aug} = \left[(\mathbf{X}^f_{t_1})', \ldots, (\mathbf{X}^f_{t_K})', \ (\mathbf{X}^f_{t_V})'\right]' \in \mathbb{R}^{(K+1)L_X \times L_E}, \tag{3.7}$$

where

$$\mathbf{X}_{t_k}^f = \mathbf{X}_{t_0}^a + \int_{t_0}^{t_k} \dot{\mathbf{X}}(t)dt, \tag{3.8}$$

with $\mathbf{X}_{t_0}^a \in \mathbb{R}^{L_X \times L_E}$ being the analysis ensemble at t_0. For computational efficiency, the impact of the future routine observations is processed in advance, and the outcome of this process provides the prior information for the selection process. The routine observation matrix for the augmented system is expressed as

$$H_{aug}^r = [\ H^r \otimes I_K \ \ \mathbf{0}_{Kn_r \times L_X}], \quad H^r \in \mathbb{R}^{n_r \times L_X} \tag{3.9}$$

where n_r is the number of routine measurements at each time step and $\otimes$ denotes the Kronecker product. Since only the covariance information is needed (and available) for targeting, incorporation of the future routine networks involves only the perturbation ensemble update

$$\widetilde{\mathbf{X}}_{aug}^a = (I - K_{aug}^r H_{aug}^r)\widetilde{\mathbf{X}}_{aug}^f \tag{3.10}$$

without updating the ensemble mean. In the sequential EnSRF scheme, this process can be performed one observation at a time.

$\widetilde{\mathbf{X}}_{aug}^a$ will be utilized to construct the prior covariance matrix for the sensor selection problem. Since the selection problem only concerns the search space and the verification region, it deals with a submatrix of the $\widetilde{\mathbf{X}}_{aug}^a$:

$$\widetilde{\mathbf{X}}_{X_S \cup V} = \left[\xi'_{X_1}, \ldots, \xi'_{X_N},\ \xi'_{V_1}, \ldots, \xi'_{V_M}\right]' \tag{3.11}$$

where $\xi_{(\cdot)} \in \mathbb{R}^{1 \times L_E}$ represents the row vector of $\widetilde{\mathbf{X}}_{aug}^a$ corresponding to the subscribed variable. Using this ensemble, the covariance matrix of $Y \equiv \{Y_1, \ldots, Y_k\} \subset X_S \cup V$ can be evaluated as

$$\mathbf{P}(Y) = \left[\xi'_{Y_1}, \ldots, \xi'_{Y_k}\right]' \left[\xi'_{Y_1}, \ldots, \xi'_{Y_k}\right] / (L_E - 1). \tag{3.12}$$

The distribution of $X_S \cup V$ is not jointly Gaussian in general due to nonlinearity of the dynamics; however, the entropy information is assumed to be well represented by the

expression in (2.3). The numerical results in Section 4.4 will verify this assumption by showing the equivalence of the mutual information values using the FS and BS forms for a simplified weather model.

After obtaining the *prior* ensemble in (3.11), the forward selection can be written in terms of ensembles as

$$\mathbf{s}^{\star}_{F,En} = \arg\min_{\mathbf{s}\in\mathbb{S}_n} \tfrac{1}{2}\log\det\left(\tfrac{1}{L_E-1}\widetilde{\mathbf{X}}_{V|Z_\mathbf{s}}\widetilde{\mathbf{X}}'_{V|Z_\mathbf{s}}\right), \tag{3.13}$$

and the backward ensemble targeting is expressed as

$$\begin{aligned}\mathbf{s}^{\star}_{B,En} = \arg\max_{\mathbf{s}\in\mathbb{S}_n} \; & \tfrac{1}{2}\log\det\left(\tfrac{1}{L_E-1}\widetilde{\mathbf{X}}_{X_\mathbf{s}}\widetilde{\mathbf{X}}'_{X_\mathbf{s}} + R_\mathbf{s}\right) \\ & - \tfrac{1}{2}\log\det\left(\tfrac{1}{L_E-1}\widetilde{\mathbf{X}}_{X_\mathbf{s}|V}\widetilde{\mathbf{X}}'_{X_\mathbf{s}|V} + R_\mathbf{s}\right).\end{aligned} \tag{3.14}$$

The conditional ensembles, $\widetilde{\mathbf{X}}_{V|Z_\mathbf{s}}$ in FS and $\widetilde{\mathbf{X}}_{X_\mathbf{s}|V}$ in BS, can be computed using the sequential ensemble update formula in (2.17). The computational cost of the forward selection in (3.13) and of the backward selection in (3.14) are compared analytically in section 3.2.2 and numerically in section 3.4.1.

3.1.2 Sensor Management for EKF Tracking of a Moving Target

The main objective of this chapter is to present a computationally efficient way of targeting for a large-scale systems such as weather prediction. A secondary objective is to identify the classes of systems for which the selection algorithm offers a significant computational advantage. To this end, this work also considers a smaller-scale sensor selection problem for comparison. The smaller-scale example considered is the single time-step version of the sensor management problem introduced in [10].

The sensor management problem is the decision of which sensors at fixed locations to turn on to best track a moving target under a limited communication budget. The nominally constant-velocity, two-dimensional motion of the target is modeled by the

linear system:

$$\mathbf{x}_{t+1} = F\mathbf{x}_t + \mathbf{w}_t \tag{3.15}$$

where $\mathbf{x} = [p_x\ v_x\ p_y\ v_y]'$ with $p_{(\cdot)}$ and $v_{(\cdot)}$ denoting the position and velocity in each direction, and $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_W)$ is a white Gaussian noise. F and Σ_W are given as

$$F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \otimes I_2,\ \Sigma_W = \bar{w} \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & \Delta t \end{bmatrix} \otimes I_2. \tag{3.16}$$

Denoting the measurement taken by the s-th sensor at time t as z_t^s, the s-th sensor measures the noise-corrupted quasi-distance to the target:

$$z_t^s = \rho_1(||M_p\mathbf{x}_k - l^s||_2^2 + \rho_2)^{-1} + v_t^s. \tag{3.17}$$

v_t^s is a zero-mean white Gaussian noise with variance of R_s, and it is independent of the process noise and the sensing noise for other sensors. $M_p = [1\ 0] \otimes I_2$ and l^s is the location of s-th sensor, and ρ_1 and ρ_2 are selected to model the signal-to-noise ratio (SNR) of the sensor.

The single time-step sensor management decision at time t considers the following *forward* selection:

$$\begin{aligned} \mathbf{s}_{F,Kf}^{\star} &= \arg\max_{\mathbf{s}\in\mathbb{S}_n} \mathcal{H}(\mathbf{x}_t|\mathbf{z}_{0:t-1}) - \mathcal{H}(\mathbf{x}_t|\mathbf{z}_t^{\mathbf{s}}, \mathbf{z}_{0:t-1}) \\ &= \arg\min_{\mathbf{s}\in\mathbb{S}_n} \mathcal{H}(\mathbf{x}_t|\mathbf{z}_t^{\mathbf{s}}, \mathbf{z}_{0:t-1}) \end{aligned} \tag{3.18}$$

where $\mathbf{z}_{0:t-1}$ and $\mathbf{z}_t^{\mathbf{s}}$ denote the measurement sequence up until time $t-1$, and the current measurement taken by the set of sensors $\mathbf{s}$, respectively. In spite of the nonlinearity in the observation model, the entropy of the state is assumed to be well approximated by log det of its covariance estimate based on linearization as it was in [10]:

$$\mathcal{H}(\mathbf{x}_t|\mathbf{z}_t^{\mathbf{s}}, \mathbf{z}_{0:t-1}) = \tfrac{1}{2}\log\det\mathbf{P}(\mathbf{x}_t|\mathbf{z}_t^{\mathbf{s}}, \mathbf{z}_{0:t-1}) + \lambda_{\mathbf{x}} \tag{3.19}$$

where $\lambda_{\mathbf{x}} = 2\log(2\pi e)$. Given the prior covariance matrix $\mathbf{P}(\mathbf{x}_t|\mathbf{z}_{0:t-1})$, the posterior

covariance matrix can be computed by the EKF update equation in Joseph form [39]:

$$\mathbf{P}(\mathbf{x}_t|\mathbf{z}_t^{\mathbf{s}}, \mathbf{z}_{0:t-1}) = (I - K_{\mathbf{s}}H_{\mathbf{s}})\mathbf{P}(\mathbf{x}_t|\mathbf{z}_{0:t-1})(I - K_{\mathbf{s}}H_{\mathbf{s}})' + K_{\mathbf{s}}R_{\mathbf{s}}K_{\mathbf{s}}' \tag{3.20}$$

with the Kalman gain

$$K_{\mathbf{s}} = \mathbf{P}(\mathbf{x}_t|\mathbf{z}_{0:t-1})H_{\mathbf{s}}' \left(H_{\mathbf{s}}\mathbf{P}(\mathbf{x}_t|\mathbf{z}_{0:t-1})H_{\mathbf{s}}' + R_{\mathbf{s}}\right)^{-1}. \tag{3.21}$$

The (linearized) observation matrix $H_{\mathbf{s}} \in \mathbb{R}^{n\times 4}$ consists of rows expressed as

$$H_s = -2\rho_1(||M_p\mathbf{x}_t - l^s||_2^2 + \rho_2)^{-1}(M_p\mathbf{x}_t - l^s)'M_p, \ \forall s \in \mathbf{s}. \tag{3.22}$$

The forward selection performs $\binom{N}{n}$ times of the measurement update process in (3.20).

The equivalent *backward* formulation for (3.18) is posed as:

$$\mathbf{s}_{B,Kf}^{\star} = \arg\max_{\mathbf{s}\in\mathbb{S}_n} \ \mathcal{H}(\mathbf{z}_t^{\mathbf{s}}|\mathbf{z}_{0:t-1}) - \mathcal{H}(\mathbf{z}_t^{\mathbf{s}}|\mathbf{x}_t, \mathbf{z}_{0:t-1}). \tag{3.23}$$

The prior entropy of $\mathbf{z}_t^{\mathbf{s}}$ is presented as

$$\begin{aligned}\mathcal{H}(\mathbf{z}_t^{\mathbf{s}}|\mathbf{z}_{0:t-1}) &= \tfrac{1}{2}\log\det\mathbf{P}(\mathbf{z}_t^{\mathbf{s}}) + \lambda_{\mathbf{s}} \\ &= \tfrac{1}{2}\log\det(H_{\mathbf{s}}\mathbf{P}(\mathbf{x}_t|\mathbf{z}_{0:t-1})H_{\mathbf{s}}' + R_{\mathbf{s}}) + \lambda_{\mathbf{s}}\end{aligned}$$

where $\lambda_{\mathbf{s}} = \frac{n}{2}\log(2\pi e)$. Since the current measurement is conditionally independent of the previous measurements for a given current state, the posterior entropy for $\mathbf{z}_t$ simply becomes

$$\mathcal{H}(\mathbf{z}_t^{\mathbf{s}}|\mathbf{x}_t, \mathbf{z}_{0:t-1}) = \tfrac{1}{2}\log\det R_{\mathbf{s}} + \lambda_{\mathbf{s}} = \tfrac{1}{2}\textstyle\sum_{s\in\mathbf{s}}\log R_s + \lambda_{\mathbf{s}}. \tag{3.24}$$

It should be noted that no measurement update is needed for the backward formulation, since the posterior entropy for $\mathbf{z}_t^{\mathbf{s}}$ can be calculated from the measurement noise variance with simple arithmetic; this aspect was exploited in [10] for an efficient

implementation. This thesis identifies another aspect of computational advantages of the backward selection that can apply to more general sensor selection problems.

3.2 Computation Time Analysis

In the previous section, the backward selection was suggested as an alternative of the forward selection that needs to perform covariance updates $\binom{N}{n}$ times to find the optimal solution using an exhaustive search. This section shows that the backward selection requires only one covariance update and this can lead to an reduced computation time compared to the forward selection. The conventional covariance form inference will be first considered for a general discussion on this computational effect, and the EnSRF-based targeting will be specifically dealt with to derive detailed expressions of the computation times of the forward and backward selection methods.

The computation for the prediction step such as the ensemble augmentation procedure in (3.8) will not be considered in this section's analysis, because the primary purpose of this section is comparison of the forward and the backward formulation in addressing the combinatorial sensor selection process. As the prediction step causes the same amount of computational overhead in both the forward and the backward selection, it does not influence the main results. However, it should be pointed out that for a large-scale realistic weather system, the computational cost of the prediction step, which involves integration of nonlinear differential equations with large number of state variables, gives rise to another challenge, and many work are devoted to improve this aspect of the computation time. However, efficiency in the prediction of nonlinear systems is beyond the scope of this thesis.

3.2.1 Conventional Covariance Form

In a conventional linear least-square estimation framework, the conditional covariance matrices needed for FS in (3.4) can be computed as

$$\mathbf{P}(V|Z_{\mathbf{s}}) = \mathbf{P}(V) - \mathbf{P}(V, X_{\mathbf{s}})[\mathbf{P}(X_{\mathbf{s}}) + R_{\mathbf{s}}]^{-1}\mathbf{P}(X_{\mathbf{s}}, V) \tag{3.25}$$

where $\mathbf{P}(V)$ is already known. After the conditional covariance is computed, FS calculates the log det value. Thus, an exhaustive search to find $\mathbf{s}_F^\star$ would perform the update equation in (3.25) followed by a determinant calculation of a $M \times M$ symmetric positive definite matrix a total of $\binom{N}{n}$ times. The resulting computation time for this forward selection process is then

$$T_F \approx \binom{N}{n} \left(\texttt{TimeUpdate}_{M,n} + \texttt{TimeDet}_M \right) \tag{3.26}$$

when other computational overhead, such as memory access and sorting, is ignored. $\texttt{TimeUpdate}_{n,M}$ corresponds to the time taken to calculate the conditional covariance of an M-dimensional Gaussian random vector conditioned on a disjoint n-dimensional vector, and $\texttt{TimeDet}_M$ is the time spent to calculate the determinant of a $M \times M$ symmetric positive definite matrix.

On the other hand, the conditioning process for the backward selection in (3.5) can be written as

$$\mathbf{P}(X_\mathbf{s}|V) = \mathbf{P}(X_\mathbf{s}) - \mathbf{P}(X_\mathbf{s}, V)\mathbf{P}(V)^{-1}\mathbf{P}(V, X_\mathbf{s}) \tag{3.27}$$

with known $\mathbf{P}(V)$. In case $M/n \sim \mathcal{O}(1)$, this update equation takes as $\mathcal{O}(1)$ times long as the forward update in (3.25). However, note that $\mathbf{P}(X_\mathbf{s}|V)$ can be evaluated from an alternative update equation:

$$\mathbf{P}(X_\mathcal{S}|V) = \mathbf{P}(X_\mathcal{S}) - \mathbf{P}(X_\mathcal{S}, V)\mathbf{P}(V)^{-1}\mathbf{P}(V, X_\mathcal{S}), \tag{3.28}$$

which computes the conditional covariance over the entire search space. Having computed $\mathbf{P}(X_\mathcal{S}|V)$, evaluation of $\mathbf{P}(X_\mathbf{s}|V)$ requires simply extracting the corresponding principal minor from $\mathbf{P}(X_\mathcal{S}|V)$, which is a trivial computation. The unconditioned covariance $\mathbf{P}(X_\mathbf{s})$ can also be extracted from the known $\mathbf{P}(X_\mathcal{S})$ in the same way. Afterwards, BS computes the determinants of the unconditioned and the conditioned covariance matrices. Since there are $\binom{N}{n}$ pairs of conditioned and unconditioned covariance matrices, the exhaustive search procedure of the backward selection will

take

$$T_B \approx \texttt{TimeUpdate}_{N,M} + 2\binom{N}{n}\texttt{TimeDet}_n. \tag{3.29}$$

Although the backward selection also has the combinatorial aspect in terms of the determinant calculation, it computes the conditional covariance only once. Since $\texttt{TimeUpdate}_{N,M}$ does not depend on n, it can be first inferred that the covariance update step for BS scales easily to the case of large n.

Scalability

This work first compares the scalability of FS and BS. To analyze asymptotic scalability, the big-$\mathcal{O}$ notations [40] are used. When considering the number of floating-point operations (`flops`) for the covariance updates in (3.26)

$$\begin{aligned}\binom{N}{n}\texttt{TimeUpdate}_{M,n} &\sim \binom{N}{n}\left[\mathcal{O}(n^3 + Mn^2 + M^2n)\right] \\ &\sim \mathcal{O}(N^n n \max\{M,n\}^2),\end{aligned} \tag{3.30}$$

while in (3.29)

$$\begin{aligned}\texttt{TimeUpdate}_{N,M} &\sim \mathcal{O}(M^3) + \mathcal{O}(NM^2) + \mathcal{O}(N^2M) \\ &\sim \mathcal{O}(N^2M)\end{aligned} \tag{3.31}$$

where $N \gg \max\{M,n\}$. Regarding the determinant calculation, $\texttt{TimeDet}_M \sim \mathcal{O}(M^3)$ and $\texttt{TimeDet}_n \sim \mathcal{O}(n^3)$. Thus, the overall complexity of FS and BS will be

$$\begin{aligned}T_F &\sim \mathcal{O}(N^n n \max\{M,n\}^2) + \mathcal{O}(N^n M^3) \\ &\sim \mathcal{O}(N^n \max\{M,n\}^3),\end{aligned} \tag{3.32}$$

and

$$T_B \sim \mathcal{O}(N^2M) + \mathcal{O}(N^n n^3) \sim \mathcal{O}(N^n n^3). \tag{3.33}$$

Note that both T_F and T_B are exponential in the number of sensing points; however, since $\mathcal{O}(\max\{M,n\}^3) \gtrsim \mathcal{O}(n^3)$, the backward selection scales better than (or at least equivalent to) the forward selection, as the problem size grows.

Efficiency

In addition to the complexity, the ratio of the computation times T_F/T_B for the asymptotic case can be approximately obtained by counting the `flops` for each algorithm, where *asymptotic* corresponds to $N \gg \max\{M,n\}$ and $\min\{M,n\} \gg 1$. Inversion of a $p \times p$ symmetric positive matrix requires approximately $\frac{2}{3}p^3$ `flops`, while a determinant calculation using Cholesky factorization requires approximately $\frac{1}{3}p^3$ `flops` [41]. Therefore,

$$T_F/T_B \approx \frac{\binom{N}{n}\left[\frac{2}{3}n^3 + 2Mn^2 + M^2n + \frac{1}{3}M^3\right]}{\frac{2}{3}M^3 + 2NM^2 + N^2M + 2\binom{N}{n}\frac{1}{3}n^3} \tag{3.34}$$

$$\approx 1 + 3\tfrac{M}{n} + \tfrac{3}{2}\tfrac{M^2}{n^2} + \tfrac{1}{2}\tfrac{M^3}{n^3} \tag{3.35}$$

$$\approx \begin{cases} \frac{1}{2}M^3/n^3 \gg 1, & M \gg n \\ 1, & M \ll n. \end{cases} \tag{3.36}$$

Thus, the relative efficiency of BS compared to FS depends on the ratio M/n, but it is at least unity, even in the case $M \ll n$ for which the determinant calculation for BS is very expensive.

Remark 1. Since the potential computational advantage of the backward form over the forward form is due to the expense of calculating the conditional covariance in (3.25), the trend can be different in case the probabilistic relation over $X_S \cup V$ is described as a graphical model. In a graphical model, what is known a priori is the information matrix $\mathbf{J}(X_S \cup V) \triangleq \mathbf{P}(X_S \cup V)^{-1}$ rather than the covariance matrix. Since $\log\det\mathbf{P}(Y) = -\log\det\mathbf{P}(Y)^{-1}$, the following quantities need to be computed in the selection process: $\mathbf{J}(V|Z_\mathbf{s})$ for FS, and $\mathbf{J}(Z_\mathbf{s})$ and $\mathbf{J}(Z_\mathbf{s}|V)$ for BS. For simplicity

of discussion, assume $R_s = 0,\ \forall s \in \mathcal{S}$ for now; thus, $Z_s = X_s,\ \forall s \in \mathcal{S}$. It is noted that in order to compute any of the three information matrices mentioned above, the *marginalization* process to compute $\mathbf{J}(X_\mathbf{s} \cup V)$ is needed. Note that this marginalized information matrix is not a submatrix of $\mathbf{J}(X_\mathcal{S} \cup V)$; instead, it is expressed as

$$\begin{aligned}\mathbf{J}(X_\mathbf{s} \cup V) &\triangleq \begin{bmatrix} \bar{J}(\mathbf{s}) & \bar{J}(\mathbf{s}, V) \\ \bar{J}(V, \mathbf{s}) & \bar{J}(V) \end{bmatrix} \\ &= \begin{bmatrix} J_\mathbf{s} - J_{\mathbf{s},\mathbf{s}^c} J_{\mathbf{s}^c}^{-1} J_{\mathbf{s}^c,\mathbf{s}} & J_{\mathbf{s},V} - J_{\mathbf{s}\mathbf{s}^c} J_{\mathbf{s}^c}^{-1} J_{\mathbf{s}^c,V} \\ J_{V,\mathbf{s}} - J_{V,\mathbf{s}^c} J_{\mathbf{s}^c}^{-1} J_{\mathbf{s}^c,\mathbf{s}} & J_V - J_{V,\mathbf{s}^c} J_{\mathbf{s}^c}^{-1} J_{\mathbf{s}^c,V} \end{bmatrix}\end{aligned} \tag{3.37}$$

where $J_{(\cdot,\cdot)}$ and $J_{(\cdot)}$ denote the corresponding submatrices of the a priori information matrix $\mathbf{J}(X_\mathcal{S} \cup V)$ with $\mathbf{s}^c \triangleq \mathcal{S} \setminus \mathbf{s}$. Once $\mathbf{J}(X_\mathbf{s} \cup V)$ is computed, FS does not require any further marginalization process, because $\mathbf{J}(V|X_\mathbf{s}) = \bar{J}(V)$. In case of BS, the conditioned information matrix is simply obtained by $\mathbf{J}(X_\mathbf{s}|V) = \bar{J}(X_\mathbf{s})$; however, the unconditioned information matrix requires one more marginalization process:

$$\mathbf{J}(X_\mathbf{s}) = \bar{J}(X_\mathbf{s}) - \bar{J}(X_\mathbf{s}, V)\bar{J}(V)^{-1}\bar{J}(V, X_\mathbf{s}). \tag{3.38}$$

In other words, in the graphical model, BS needs to conduct both (3.37) and (3.38) combinatorially many times, while FS only needs to do (3.37) that many times; therefore, FS is computationally more effective than BS in terms of the cost of the covariance updates. Since $\mathbf{J}(X_\mathbf{s}|\cdot)$ is not a submatrix of $\mathbf{J}(X_\mathcal{S}|\cdot)$, the similar type of computational advantage that BS has in the covariance form representation does not exist in the information form. However, since the size of $J_{\mathbf{s}^c}$ is $(N-n) \times (N-n)$, the computational cost of doing (3.37) dominates that of doing (3.38) in case N is very large. Since the covariance updates in (3.25) and (3.27) can be done much faster than the marginalization in (3.37), a more reasonable strategy of addressing sensor selection problems in a graphical model would be to take the inverse of $\mathbf{J}(X_\mathcal{S} \cup V)$ and conduct the backward search in the covariance form.

3.2.2 EnSRF-Based Targeting

As described in section 3.2.1, the backward search is faster than or comparable to the forward search, and becomes substantially faster if $M > n$. This section details the computation time of both search schemes in the EnSRF-based targeting framework. In case of using EnSRF, the computational cost of the covariance update (or equivalently, ensemble update) not only relies on N, n, and M but also on the ensemble size L_E. In this section, the expressions of the computation time for FS and BS will be presented for both purposes of emphasizing the advantage of BS in EnSRF-based targeting and of providing practical (as well as asymptotic) estimates of the actual computation times that can be used to indicate the real tractability of the problem.

For both the forward and the backward formulation in (3.13) and (3.14), respectively, the selection processes involve four computation elements: 1) perturbation ensemble updates to obtain conditional ensemble matrices, 2) covariance matrix computations using the conditional (as well as unconditional for BS) ensembles, 3) determinant calculation of the evaluated covariance matrices, and 4) finally, selection of the best candidate from the reward list. This section describes these four by introducing the following four atomic time units: δ_{L_E}, σ_{L_E}, τ_p, and θ_q.

δ_{L_E} represents the time to update L_E ensemble members associated with one state variable by one observation. In the sequential update framework, the update equation in (2.17) can be done row by row as

$$\xi_j^+ = \xi_j - \frac{c_1 c_2}{L_E - 1} (\xi_j \xi_i') \xi_i \tag{3.39}$$

where the superscript "+" denotes the updated one. δ_{L_E} is the time for conducting a single run of (3.39). It will take $pq\delta_{L_E}$ to update the ensembles of p states with q observations, since (3.39) will be evaluated for each $j \in \{1, \ldots, p\}$ to incorporate a single measurement, and this process should be repeated q times. Also, δ_{L_E} is approximately linear in L_E, as the number of floating-point operations is $3L_E$ for given $\frac{c_1 c_2}{L_E - 1}$ when the scaling of a vector is assumed to need a single operation.

σ_{L_E} is the time to compute the inner product of two vectors with size L_E. Then, the time to multiply a $p \times L_E$ matrix with its transpose on its right, which is needed to evaluate the covariance matrix from the perturbation ensemble matrix, can be approximated as $\frac{1}{2}p(p+1)\sigma_{L_E}$, because a covariance matrix is symmetric. Note that σ_{L_E} is linear in L_E as it needs $2L_E$ `flops`.

τ_p is the time to calculate the determinant of a $p \times p$ symmetric positive definite matrix. τ_p corresponds to $\frac{1}{6}p(p+1)(2p+1)+p$ floating-point operations, which is approximately $\frac{1}{3}p^3$ for sufficiently large p. This work concerns more accurate expression of τ_p for a moderate size p, since the degree of potential advantage of the backward formulation highly depends on the ratio of unit cost of the covariance update and determinant calculation.

θ_q is the time to select the greatest element out of the list with length q, and θ_q requires approximately q `flops`.

For a given measurement choice $\mathbf{s}$ with size n, the forward search first needs to compute the conditional ensemble $\widetilde{\mathbf{X}}_{V|Z_\mathbf{s}}$. It is noted that in the sequential EnSRF framework, ensembles for $X_\mathbf{s}$ (as well as V) also need to be sequentially updated in this conditioning process. This is because the effect of observation at $s_1 \in \mathbf{s}$ on the ensembles for X_{s_2}, $s_2 \in \mathbf{s}$ should be incorporated for later update that considers observation at s_2. Although the most efficient implementation may not incorporate the change in the earlier sensing point s_1 by the later observation at s_2, this work considers the case that every ensemble for $X_\mathbf{s} \cup V$ is updated by the measurement $Z_\mathbf{s}$ to simplify the expressions of the computation time using the previously defined atomic time units. Thus, ensembles of a total of $n+M$ states are updated using n observations taken at $\mathbf{s}$; this procedure will take $n(n+M)\delta_{L_E}$. Once the calculation of the conditional ensemble $\widetilde{\mathbf{X}}_{V|Z_\mathbf{s}}$ is completed, the conditional covariance $\mathbf{P}(V|Z_\mathbf{s}) = \frac{1}{L_E-1}\widetilde{\mathbf{X}}_{V|Z_\mathbf{s}}\widetilde{\mathbf{X}}'_{V|Z_\mathbf{s}}$ is computed; the time taken in this procedure can be expressed as $\frac{1}{2}M(M+1)\sigma_{L_E}$. The next process is determinant calculation of $\mathbf{P}(V|Z_\mathbf{s})$, which will take τ_M. The exhaustive search in the forward form needs to repeat this process consisting of ensemble update, covariance evaluation, and determinant calculation

for every $\mathbf{s} \in \mathbb{S}_n$. Then, a search over a list of information rewards for each $\mathbf{s}$ whose length is $\binom{N}{n}$ will determine the best solution $\mathbf{s}^\star_{F,En}$. Thus, the estimated computation time for FS becomes

$$\widehat{T}_F = \binom{N}{n}\Big[\underbrace{n(n+M)\delta_{L_E}}_{\text{ensemble update}} + \underbrace{\tfrac{1}{2}M(M+1)\sigma_{L_E}}_{\text{cov. comp.}} + \underbrace{\tau_M}_{\text{det.}}\Big] + \theta_{\binom{N}{n}}. \tag{3.40}$$

In case of the backward selection, the conditional ensemble for $X_{\mathcal{S}} \cup V$ needs to be evaluated by the *fictitious* observations taken at the verification sites of size M. This conditioning process will take $M(N+M)\delta_{L_E}$, since ensembles for $N+M$ states are updated; the outcome of this process is $\widetilde{\mathbf{X}}_{X_{\mathcal{S}}\cup V|V}$. For a given $\mathbf{s}$, BS needs to evaluate two covariance matrices: $\mathbf{P}(X_{\mathbf{s}})$ and $\mathbf{P}(X_{\mathbf{s}}|V)$. One way to compute these from the unconditioned and the conditioned ensembles for $X_{\mathcal{S}} \cup V$ is to extract the corresponding rows of those ensembles to have $\widetilde{\mathbf{X}}_{X_{\mathbf{s}}}$ and $\widetilde{\mathbf{X}}_{X_{\mathbf{s}}|V}$, and then compute the covariance matrices by multiplying the these ensemble matrices with their transposes. However, this way involves redundant computations if repeated for all $\mathbf{s} \in \mathbb{S}_n$. Instead, it is more computationally efficient to first compute $\mathbf{P}(X_{\mathcal{S}})$ and $\mathbf{P}(X_{\mathcal{S}}|V)$, and then to extract the corresponding principal submatrices from these $N \times N$ covariance matrices. Computation of two covariances for $X_{\mathcal{S}}$ will take $2 \times \frac{1}{2}N(N+1)\sigma_{L_E}$, but this computation does not need to be repeated for different measurement choices. Having computed $\mathbf{P}(X_{\mathcal{S}})$ and $\mathbf{P}(X_{\mathcal{S}}|V)$, BS starts a combinatorial search to find $\mathbf{s}^\star_{B,En}$. For each $\mathbf{s} \in \mathbb{S}_n$, the backward search 1) extracts the submatrices from the covariance matrices for the entire search space, and 2) computes the determinants of the unconditioned and conditioned covariances. As matrix extraction takes a trivial amount of time, the computation time spent for this process is $2\tau_n$. Once all the rewards are evaluated, a search process taking $\theta_{\binom{N}{n}}$ will determine the best solution. Therefore, the estimated computation time for the backward exhaustive search is written as

$$\widehat{T}_B = \underbrace{M(N+M)\delta_{L_E}}_{\text{ensemble update}} + \underbrace{N(N+1)\sigma_{L_E}}_{\text{cov. comp.}} + 2\underbrace{\binom{N}{n}\tau_n}_{\text{det.}} + \theta_{\binom{N}{n}}. \tag{3.41}$$

Note that $\widehat{T}_B$ does not contain the combinatorial aspect in the ensemble update and the covariance computation, while all the relevant terms in $\widehat{T}_F$ are combinatorial.

It should be pointed out that $L_E/N \gtrapprox 1$ for a large-scale system, since a large number of ensembles are typically needed to accurately estimate a complex system with a large number of state variables. Since δ_{L_E} and σ_{L_E} are approximately proportional to L_E, the computation cost of ensemble update and the covariance computation will dominate the determinant calculation for a large L_E; this enables the backward formulation to remarkably reduce the computation time compared to the forward formulation.

Scalability

The scalability of EnSRF-based FS and BS can be analyzed using big-$\mathcal{O}$ methodology [40]. This analysis considers the asymptotic case in which $N \gg \max\{M,n\}$, $L_E \gg \max\{M,n\}$, and $\min\{M,n\} \gg 1$. The asymptotic complexity will be

$$\begin{aligned}\widehat{T}_F &\sim \binom{N}{n} \mathcal{O}\left(n\max\{M,n\}L_E + M^2L_E + M^3\right) \\ &\sim \mathcal{O}(N^n \max\{M,n\}^2 L_E),\end{aligned} \tag{3.42}$$

and

$$\widehat{T}_B \sim \mathcal{O}(MNL_E + N^2L_E) + \binom{N}{n}\mathcal{O}(n^3) \sim \mathcal{O}(N^n n^3). \tag{3.43}$$

The dependency of $\widehat{T}_F$ on L_E indicates that FS will not scale well as the state dimension increases, because it is typical that $L_E \propto L_X$, i.e., an increase in the state dimension usually leads to an increase in the ensemble size. In contrast, BS does scale easily with this type of problem complexity.

Efficiency

Using the relations $\delta_{L_E} \approx 3L_E$ `flops`, $\sigma_{L_E} \approx 2L_E$ `flops`, $\tau_p \approx \frac{p^3}{3}$ `flops`, and $\theta_q \approx q$ `flops` given in the definitions of atomic time units, the ratio of $\widehat{T}_F/\widehat{T}_B$ for the

asymptotic case is

$$\widehat{T}_F/\widehat{T}_B \approx \frac{\binom{N}{n}\left[3n(n+M)L_E + M(M+1)L_E + \frac{1}{3}M^3 + 1\right]}{3M(N+M)L_E + 2N(N+1)L_E + \binom{N}{n}(\frac{2}{3}n^3+1)} \tag{3.44}$$

$$\approx \frac{\binom{N}{n}[3n^2+3nM+M^2]L_E}{\binom{N}{n}(\frac{2}{3}n^3)} \tag{3.45}$$

$$\approx \tfrac{9}{2}\tfrac{L_E}{n}\cdot\left[1+\tfrac{M}{n}+\tfrac{1}{3}\tfrac{M^2}{n^2}\right], \quad (\gg 1) \tag{3.46}$$

$$\approx \begin{cases} \frac{3}{2}L_E M^2/n^3, & M \gg n \\ \frac{9}{2}L_E/n, & M \ll n. \end{cases} \tag{3.47}$$

These results show that the EnSRF-based BS is computationally more efficient than EnSRF-based FS by a factor of at least $\frac{9}{2}L_E/n$, which is large because typically $L_E \gg n$.

3.3 Sequential Greedy Strategies

3.3.1 Algorithms

As shown in (3.42) and (3.43), the computation times for both FS and BS grow exponentially with the number of targeting points. One typical approach to avoid this exponential growth is a sequential greedy algorithm that selects the best targeting point, one at a time. The sequential greedy algorithm based on the forward selection formulation is stated as follows.

Forward Sequential Greedy Selection (FSGS)

$$\begin{aligned} s_k^{FG\star} &= \arg\max_{s\in\mathcal{S}} \ \mathcal{H}(V|Z_{\mathbf{s}_{k-1}^{FG\star}}) - \mathcal{H}(V|Z_s,\, Z_{\mathbf{s}_{k-1}^{FG\star}}) \\ &= \arg\max_{s\in\mathcal{S}} \ \tfrac{1}{2}\log\det\mathbf{P}(V|Z_{\mathbf{s}_{k-1}^{FG\star}}) - \tfrac{1}{2}\log\det\mathbf{P}(V|Z_s,\, Z_{\mathbf{s}_{k-1}^{FG\star}}) \end{aligned} \tag{3.48}$$

for $k \in \{1, \ldots, n\}$, where $Z_{\mathbf{s}_k^{FG\star}} \equiv \{Z_{s_1^{FG\star}}, \ldots, Z_{s_k^{FG\star}}\}$, and $Z_{\mathbf{s}_0^{FG\star}} = \emptyset$. The selection of the k-th measurement point is made conditioned on the selections up to the $(k-1)$-th step; $\mathbf{P}(V|Z_{\mathbf{s}_{k-1}^{FG\star}})$ is a known quantity at the k-th selection step. To choose $s_k^{FG\star}$, the conditional covariance $\mathbf{P}(V|Z_s, Z_{\mathbf{s}_{k-1}^{FG\star}})$ must be computed for all $s \in \mathcal{S}$, which is followed by a determinant calculation. The computation time for the k-th selection step in (3.48) linearly increases as N increases, and this process should be repeated n times. Thus the overall computation time for FSGS grows linearly in nN, which could still be large for large N. This suggests investigating the backward greedy selection algorithm:

Backward Sequential Greedy Selection (BSGS)

$$\begin{aligned} s_k^{BG\star} &= \arg\max_{s\in\mathcal{S}} \ \mathcal{H}(Z_s|Z_{\mathbf{s}_{k-1}^{BG\star}}) - \mathcal{H}(Z_s|V,\, Z_{\mathbf{s}_{k-1}^{BG\star}}) \\ &= \arg\max_{s\in\mathcal{S}} \ \tfrac{1}{2}\log\left(\mathrm{Var}(X_s|Z_{\mathbf{s}_{k-1}^{BG\star}}) + R_s\right) - \tfrac{1}{2}\log\left(\mathrm{Var}(X_s|V,\, Z_{\mathbf{s}_{k-1}^{BG\star}}) + R_s\right) \end{aligned} \tag{3.49}$$

for $k \in \{1, \ldots, n\}$, where $Z_{\mathbf{s}_k^{BG\star}} \equiv \{Z_{s_1^{BG\star}}, \ldots, Z_{s_k^{BG\star}}\}$, and $Z_{\mathbf{s}_0^{BG\star}} = \emptyset$. BSGS selects the site where the difference between the entropy conditioned on previous selections and that conditioned on the previous selections plus V, is maximized. The known quantities for the k-th selection of BSGS are: $\mathbf{P}(X_{\mathcal{S}})$ in case $k = 1$, and $\mathbf{P}(X_{\mathcal{S}}|Z_{\mathbf{s}_{k-2}^{BG\star}})$ and $\mathbf{P}(X_{\mathcal{S}}|V, Z_{\mathbf{s}_{k-2}^{BG\star}})$ in case $k > 1$. Two aspects that characterize the computational benefits of this algorithm are: 1) BSGS does not involve the computation of the determinant of a large matrix, but instead only a scalar, and 2) at the k-th step $(k > 1)$, only two covariance updates by a single observation $Z_{s_{k-1}^{BG\star}}$ are needed to compute $\mathbf{P}(X_{\mathcal{S}}|Z_{s_{k-1}^{BG\star}}, Z_{\mathbf{s}_{k-2}^{BG\star}})$ and $\mathbf{P}(X_{\mathcal{S}}|V, Z_{s_{k-1}^{BG\star}}, Z_{\mathbf{s}_{k-2}^{BG\star}})$. Note however that BSGS gives the same solution as FSGS:

Proposition 2. The forward sequential greedy selection and the backward sequential greedy selection produce the same solution:

$$s_k^{FG\star} = s_k^{BG\star}, \ \forall\, k \in [1, n] \cap \mathbb{Z}. \tag{3.50}$$

Proof. The proof is by induction. Since Proposition 1 is true for $n = 1$, the above statement is true for $k = 1$. Suppose that $s_k^{FG\star} = s_k^{BG\star}, \forall k \in [1, m] \cap \mathbb{Z}$ with $m < n$. Because $\mathcal{I}(V; Z_s|Y) = \mathcal{I}(Z_s; V|Y)$ for any random vector Y, this identity should be true for $Y = Z_{\mathbf{s}_m^{FG\star}} (= Z_{\mathbf{s}_m^{BG\star}}$, by assumption). Therefore, the objective functions in (3.48) an (3.49) are related as

$$\mathcal{H}(V|Z_{\mathbf{s}_m^{FG\star}}) - \mathcal{H}(V|Z_s, Z_{\mathbf{s}_m^{FG\star}}) = \mathcal{H}(Z_s|Z_{\mathbf{s}_m^{BG\star}}) - \mathcal{H}(Z_s|V, Z_{\mathbf{s}_m^{BG\star}}), \ \forall\ s \in \mathcal{S}, \quad (3.51)$$

then it follows that $s_{m+1}^{FG\star} = s_{m+1}^{BG\star}$. Combining these results yields that $s_k^{FG\star} = s_k^{BG\star}, \ \forall\ k \in [1, n] \cap \mathbb{Z}$. □

3.3.2 Computation Time

Computation time for the EnSRF-based greedy algorithms can also be expressed with the atomic time units defined in section 3.2.2. When neglecting the search space shrinkage by the previous selections, FSGS computes the conditional covariance $\mathbf{P}(V|Z_s, Z_{\mathbf{s}_{k-1}^{FG\star}})$ in (3.48), which takes $M\delta_{L_E} + \frac{1}{2}M(M+1)\sigma_{L_E}$, a total of N times for every k, while it also computes the determinant of that covariance matrix the same many times. Note that once the k-th sensing point is selected, additional ensemble updates must be done to compute the conditional ensemble for $X_{\mathcal{S}} \cup V$ conditioned on $Z_{\mathbf{s}_k}^{FG\star} \equiv Z_{\mathbf{s}_{k-1}}^{FG\star} \cup \{Z_{s_k}^{FG\star}\}$ that will be used for $(k+1)$-th selection. This additional ensemble update requires $(N + M)\delta_{L_E}$ per $k < n$. Therefore, the estimated FSGS computation time is

$$\begin{aligned} \widehat{T}_{FG} =& n\Big[\underbrace{N\left(M\delta_{L_E} + \tfrac{1}{2}M(M+1)\sigma_{L_E} + \tau_M\right)}_{\text{update and det. comp. for current selection}} + \theta_N\Big] \\ &+ (n-1) \times \underbrace{(N+M)\delta_{L_E}}_{\text{ens. updates for next step}}, \end{aligned} \quad (3.52)$$

which shows that most terms are proportional to nN.

In BSGS, the impact of V is only calculated at the first selection step, so the cost of the ensemble update needed to pick the rest of the points is lower than that

required to pick the first. In particular, for $k = 1$, an ensemble update for $X_{\mathcal{S}} \cup V$ by the fictitious measurement of V is needed, while for $k > 1$, two ensemble updates for $X_{\mathcal{S}}$ by the measurement $Z^{BG\star}_{s_{k-1}}$ given ensembles conditioned on $Z^{BG\star}_{\mathbf{s}_{k-2}}$ and on $Z^{BG\star}_{\mathbf{s}_{k-2}} \cup V$, are conducted. Thus, the computation times for the first and the rest of the ensemble updates are $M(N+M)\delta_{L_E}$ and $2N\delta_{L_E}$, respectively. Note that BSGS only computes the diagonal elements of the covariance matrices for $X_{\mathcal{S}}$, which provides $\mathrm{Var}(X_s|\cdot)$, $\forall s \in \mathcal{S}$; this computation takes $N\sigma_{L_E}$ (in contrast to $\frac{1}{2}N(N+1)\sigma_{L_E}$ for a full matrix computation). Then, the estimated computation time for BSGS is

$$\begin{aligned}\widehat{T}_{BG} = &\underbrace{M(N+M)\delta_{L_E}}_{\text{ens. up. for } k=1} + (n-1) \times \underbrace{2N\delta_{L_E}}_{\text{ens. up. for } k>1} \\ &+ n \underbrace{2N\sigma_{L_E}}_{\text{var. comp.}} + n\theta_N.\end{aligned} \tag{3.53}$$

These results show that BSGS scales better than FSGS as n increases, since the terms proportional to n is $\mathcal{O}(NL_E)$ in $\widehat{T}_{BG}$ while those in $\widehat{T}_{FG}$ are $\mathcal{O}(NM^2L_E + NM^3)$.

For the asymptotic case, the complexities of the forward and backward sequential greedy algorithms are

$$\widehat{T}_{FG} \sim \mathcal{O}(nM^2NL_E), \text{ and } \widehat{T}_{BG} \sim \mathcal{O}(\max\{M,n\}NL_E). \tag{3.54}$$

Since $\mathcal{O}(nM^2) > \mathcal{O}(\max\{M,n\})$ in both terms of M and n, BSGS scales better than FSGS as n and/or M increase.

The efficiency of the backward method can be estimated as

$$\widehat{T}_{FG}/\widehat{T}_{BG} \approx \frac{M^2nNL_E}{(3M+10n)NL_E} \approx \frac{M^2}{10+3(M/n)}, \tag{3.55}$$

$$\approx \begin{cases} Mn/3, & M \gg n \\ M^2/10, & M \ll n. \end{cases} \tag{3.56}$$

using the relations between atomic time units and the `flops`. Note that the formula in (3.55) is an increasing function of both M and n. Thus, the minimum value of

Table 3.1: Summary of the asymptotic complexity for the cases where $N \gg \max\{M,n\}$, $L_E \gg \max\{M,n\}$, $\min\{M,n\} \gg 1$

Case (Section #)	Forward Complexity	Backward Complexity
Cov. Exhaustive (3.2.1)	$\mathcal{O}(N^n \max\{M,n\}^3)$	$\mathcal{O}(N^n n^3)$
EnSRF Exhaustive (3.2.2)	$\mathcal{O}(N^n \max\{M,n\}^2 L_E)$	$\mathcal{O}(N^n n^3)$
EnSRF Greedy (3.3.2)	$\mathcal{O}(nM^2 N L_E)$	$\mathcal{O}(\max\{M,n\} N L_E)$

Table 3.2: Summary of relative efficiency for the cases where $N \gg \max\{M,n\}$, $L_E \gg \max\{M,n\}$, $\min\{M,n\} \gg 1$

Case (Section #)	Efficiency (Fwd $\div$ Bwd)	Min. Efficiency
Cov. Exhaustive (3.2.1)	$1 + \frac{3M}{n} + \frac{3M^2}{2n^2} + \frac{M^3}{2n^2}$	≈ 1
EnSRF Exhaustive (3.2.2)	$\frac{9L_E}{2n}(1 + \frac{M}{n} + \frac{M^2}{3n^2})$	$\frac{9L_E}{2n} \gg 1$
EnSRF Greedy (3.3.2)	$M^2/(10 + 3\frac{M}{n})$	≈ 8

the above ratio is at the smallest M and n to which the asymptotic analysis can be applied. For instance, if this smallest value is $M = n = 10$, then it is inferred that $\widehat{T}_{FG}/\widehat{T}_{BG} > 8$ for all larger problems.

3.3.3 Summary of Computation Time Analysis

The asymptotic results of the computational complexity of the presented methods and the relative efficiency of the backward method compared to the forward one are summarized in Table 3.1 and 3.2. The results clearly show that the scalability and efficiency of the backward methods are never worse than the forward ones. In case of the EnSRF-based exhaustive search, the benefit of using the backward form can be significant since typically $L_E \gg n$. Also, the polynomial-time approximation of the backward method, BSGS, works at least about 8 times faster than the forward approximation, FSGS, and this relative efficiency grows as the problem size increases.

3.4 Numerical Results

3.4.1 Ensemble-Based Targeting for Weather Forecast

Several sensor targeting scenarios are considered to numerically validate the computational advantages of the proposed backward scheme. For simulation, the Lorenz-95 model (or Lorenz-2003 model with $\rho_x = \rho_y = 1$) is used. In this model, there are 36 longitudinal grids and 9 latitudinal grids, and the size of each grid is equivalent to 694km×694km. A routine network of size 93 is assumed to already be deployed over the grid space (black ∘ in Figure 3-3). The static network is dense in two portions of the grid space that could represent land, while it is sparse in the other two portions of the space, which represent oceans. It is assumed that measurements are taken every 0.05 time units (equivalent to 6 hours in real time), and the EnSRF data assimilation scheme with ensemble size $L_E = 1024$ is used to generate the initial analysis ensemble at t_0 by incorporating these measurements. The verification region is the leftmost part of the land mass on the right (consisting of $M = 10$ grid points depicted with red □ in Figure 3-3), and the verification time $t_V = 0.55$ ($\sim$ 66hrs). Targeting deploys sensors at a single time instance $t_K = 0.05$ over the search space defined by a total of $N = 108$ grid points in the left ocean, and the sensing noise variance of $R_s = 0.0004$, $\forall s \in \mathcal{S}$ is used. With this setting, the targeting results, with different numbers of targeting points, n, are obtained using the four algorithms: FS/BS/FSGS/BSGS.

First note that the backward algorithm gives the same solution as the forward algorithm for all of the cases, not only in terms of the optimal sensing locations but also the objective function values (within 0.001% error). This agreement supports the validity of the Gaussian assumption in computing the mutual information is reasonable for this problem. In Figure 3-3, the optimal and the sequential greedy solutions are illustrated for $n = 1$, 3, and 5. The shaded contour represents the local reward value for each grid point, $\mathcal{I}(V; Z_s)$, which is the entropy reduction of the verification site by a single measurement taken at location s. The differences between the optimal and sequential greedy solutions are apparent for $n = 5$ in that the optimal

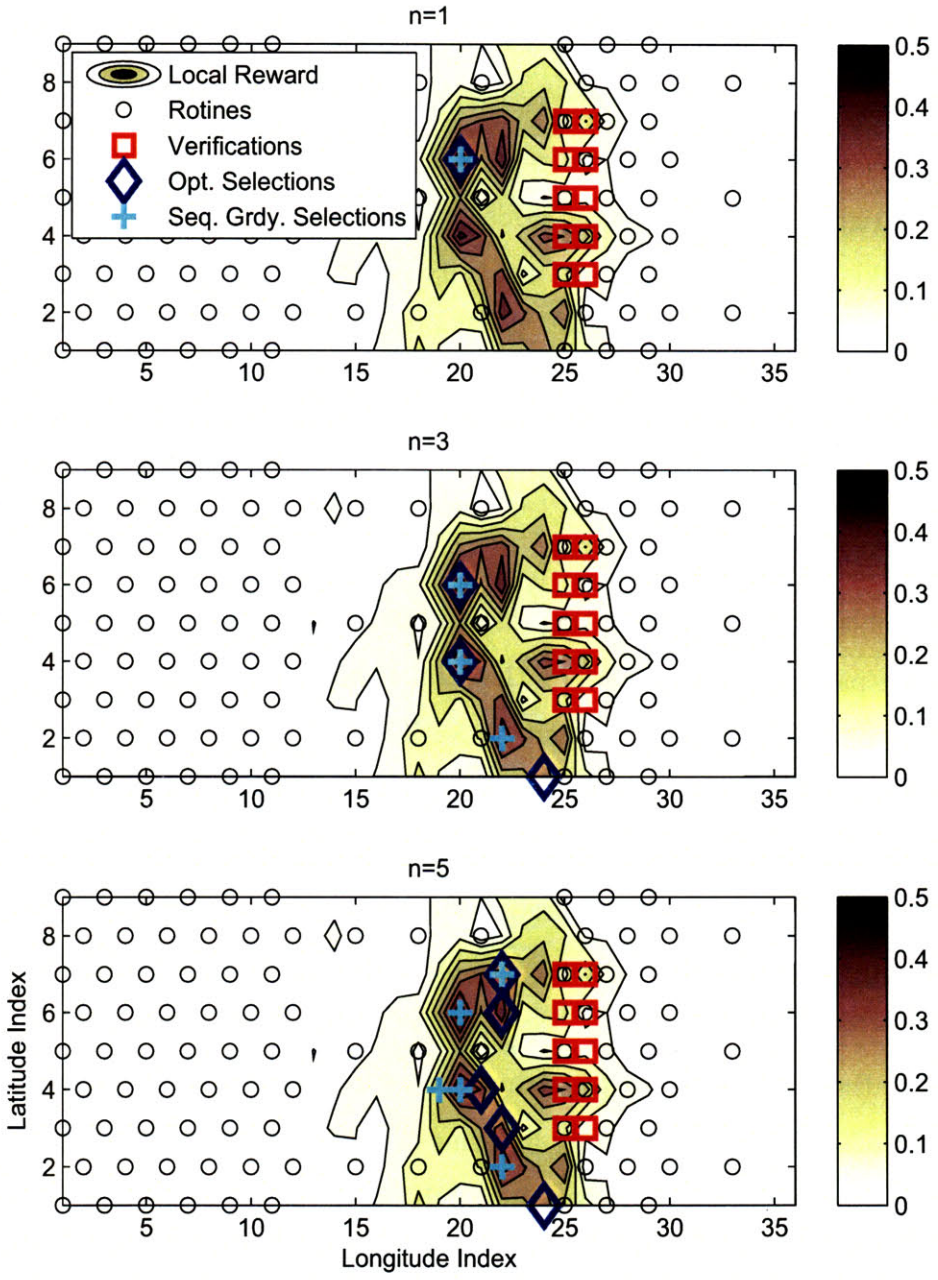

Figure 3-3: Targeted sensor locations determined by the optimal and the sequential greedy methods

result does not even select the two dark points that were selected for $n = 3$. Table 3.3 gives the resulting mutual information values by different the strategies for various n. The results of two other strategies are also shown for comparison with the optimal and the sequential greedy. The *local* greedy strategy simply selects n points in the order of the largest single targeting performance as the following:

$$s_k^{L\star} = \arg \max_{s \in \mathcal{S} \setminus \mathbf{s}_{k-1}^{L\star}} \mathcal{I}(Z_s; V) \equiv \mathcal{H}(Z_s) - \mathcal{H}(Z_s|V) \tag{3.57}$$

where $\mathbf{s}_{k-1}^{L\star} \triangleq \{s_1^{L\star}, \ldots, s_{k-1}^{L\star}\}$. The *myopic* strategy searches for the set of sensing points that represents the largest prior entropy without consideration of the verification region:

$$\mathbf{s}_{myo}^{\star} = \arg \max_{\mathbf{s} \in \mathbb{S}_n} \mathcal{H}(Z_{\mathbf{s}}). \tag{3.58}$$

First, it can be seen that the myopic strategy works never comparable to other strategies; this indicates that simply selecting the most uncertain region in the search space is never a good strategy to reduce the uncertainty in a separate verification region. Amongst other strategies, the performance gap between strategies becomes more distinct as n increases; but note that BSGS (equivalently, FSGS) is always better than the local greedy solution. This performance improvement occurs because, in contrast to the local greedy strategy, the sequential one takes into account the correlation structure over the search space by conducting conditioning processes based on previous selections. The benefit of the sequential decision over the local decision will be addressed in more details in the multi-sensor platform targeting framework in Chapter 4.

Tables 3.4 and 3.5 represent the actual and estimated computation time of each algorithm for different n. The atomic time units for computation time estimation were determined by Monte Carlo simulations in Fortran90 using LAPACK library [42] on a PC with Pentium-4 3.2GHz CPU and 1GB RAM. The atomic time units have the values: $\delta_{L_E} = 60.4$ μs, $\sigma_{L_E} = 36.7$ μs, and $\theta_q = 8.1 \times 10^{-3} q$ μs. Regarding τ_p, the values for $p \leq 10$ are obtained by simulations (e.g. $\tau_3 = 0.95\mu$s and $\tau_{10} = 6.5\mu$s). Because these p values are not large enough, the cubic relation of $\tau_p \propto p^3$ does not

Table 3.3: Mutual information values for the targeting solutions by different targeting strategies (Backwards are all same as Forwards if both are available)

n	BS(≡FS)	BSGS(≡FSGS)	Local Greedy	Myopic
1	0.46	0.46	0.46	0.01
2	0.87	0.84	0.79	0.13
3	1.24	1.19	1.02	0.22
4	1.60	1.55	1.43	0.48
5	1.95	1.86	1.66	0.51

accurately predict the actual values. The results in Tables 3.4 and 3.5 show that the estimated values of the computation time are accurate to within 40% error, which is small enough to support their use as an indicator of the computational tractability of a given problem.

Table 3.4 confirms that BS is a much faster algorithm that scales better with n than FS. Given that a real weather forecast scenario is much more complex than this example, it is clear that FS is not practical to implement for multiple targeting problems. These results show that the backward algorithm should be practical for selecting a few measurement locations (e.g., $n \leq 4$).

Table 3.5 also confirms that the superiority of the backward scheme extends to the sequential greedy case as well. Although the computation time for FSGS exhibits linear growth with n, the BSGS computation time is essentially constant for $n \leq 5$, and $T_{FG}/T_{BG} > 17$ when $n = 10$. Of particular interest is the fact that the forward sequential greedy algorithm is actually slower than the backward exhaustive search algorithm when $n = 2$, which implies the optimal solution could be obtained by the backward scheme for reasonably sized problems without sacrificing significant computational resources.

3.4.2 Sensor Selection for EKF Target Tracking

For numerical experiments of the moving target tracking, the following parameter values are used from [10]: $\Delta t = 0.25$, $\bar{w} = 0.01$, $\rho_1 = 2000$, $\rho_2 = 100$, and $R_s = 1$. A

Table 3.4: Solution time of exhaustive searches for ensemble targeting problems with Lorenz-95 model

N	n	T_F (s)	T_B (s)	$\widehat{T}_F$ (s)	$\widehat{T}_B(s)$
108	1	0.27	0.27	0.29	0.50
108	2	15.6	0.28	20.1	0.51
108	3	646.9	0.81	893.8	0.89
108	4	–	20.6	8.0 hr	16.13
108	5	–	583.4	8.4 day	440.5
108	6	–	–	173 day	2.8 hr
108	10	–	–	17000yr	16 yr

Table 3.5: Solution time of sequential greedy strategies for ensemble targeting problems with Lorenz-95 model

N	n	T_{FG} (s)	T_{BG} (s)	$\widehat{T}_{FG}$ (s)	$\widehat{T}_{BG}(s)$
108	1	0.25	0.06	0.28	0.07
108	2	0.50	0.06	0.58	0.08
108	3	0.75	0.08	0.87	0.10
108	4	0.98	0.08	1.16	0.11
108	5	1.22	0.08	1.45	0.12
108	6	1.47	0.11	1.74	0.14
108	10	2.44	0.14	2.90	0.19

total of $N = 30$ sensors are located at fixed locations determined randomly on a 15×15 two-dimensional space. The initial state value is $\mathbf{x}_0 = [0,\ 2,\ 0,\ 2]'$, which results in the nominal position at the t-th timestep $(0.5t, 0.5t)$. The sensor selection is addressed at time $t = 20$, before which an EKF has used randomly selected $n_0 = 10$ sensor measurements for state estimation at every time step. Simulations are performed with MATLAB7.1®.

Figure 3-4 illustrates the optimal sensor selection for $n = 6$ compared with the *local greedy* solution that does not take into account any correlation amongst the sensors. For this particular example, the sequential greedy solutions are the same as the optimal ones. It can be seen that sensors located in the direction of target's movement provide generally larger information gain than the sensors behind the target. While the optimal solution is identical to the local greedy solution for $n \leq 5$, the

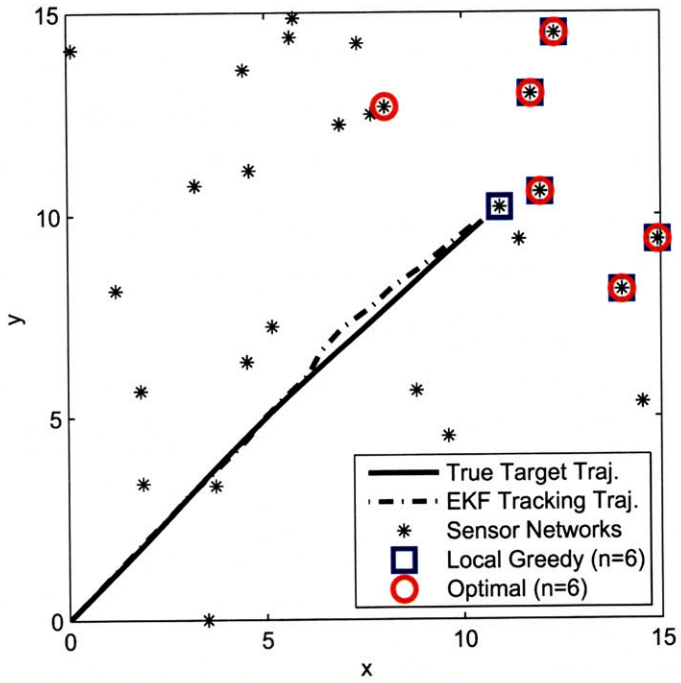

Figure 3-4: Optimal Sensor Selection for EKF Target Tracking

Table 3.6: Solution time for forward and backward methods for EKF sensor management ($N = 30$)

n	3	4	5	6	7	8	9
T_F (s)	0.52	2.77	15.0	64.7	228.8	679.7	1745.4
T_B (s)	0.08	0.64	3.52	15.4	55.1	164.5	421.6

optimal strategy chooses a sensor that is distant from the previously selected ones as the sixth choice. This avoids selecting the sensor right in front of the target that is very close to (therefore, expected to be highly correlated with each other) one of the previously selected sensors.

Table 3.6 shows the computation time needed for the forward and backward exhaustive searches for a range of selected sensors, i.e., $n = 3, \ldots, 9$. The results confirm that the backward search is always faster than the forward search, as it does not require any measurement updates. Also note that T_B is only about 4 times smaller than T_F.

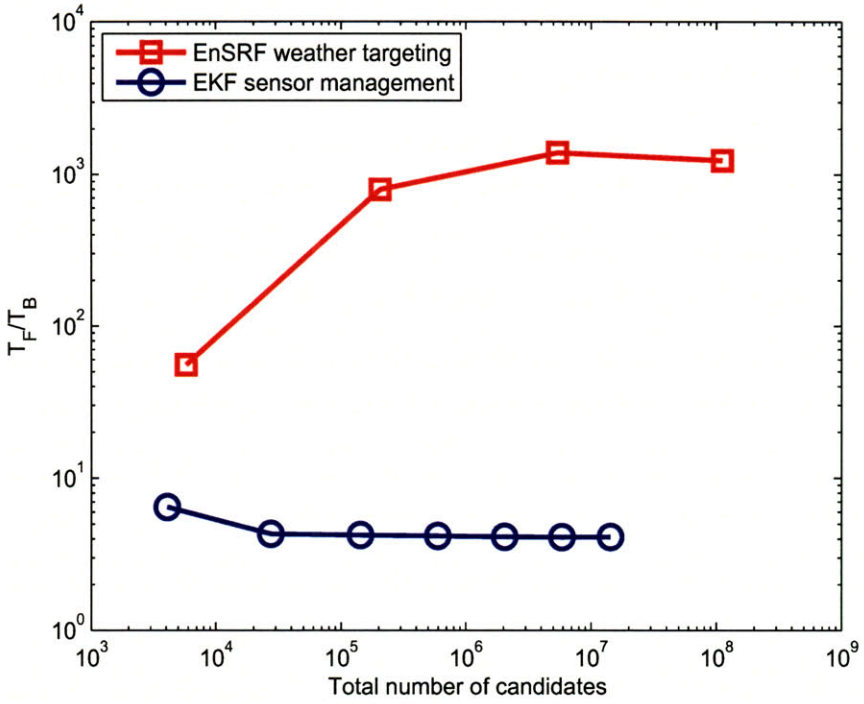

Figure 3-5: Comparison of relative computational advantage of the backward formulation

3.4.3 Summary

Figure 3-5 compares the ratio T_F/T_B of the ensemble-based targeting case and the EKF target tracking case with respect to the total number of candidates, which is taken here as an indication of the complexity of the problem. Four ensemble targeting cases ($n = 2, \ldots, 5$ with $N = 108$) and seven EKF sensor management cases ($n = 3, \ldots, 9$ with $N = 30$) are considered, and the estimated computation time is used if the actual computation time is not available. Notice that T_F/T_B is much bigger for the ensemble targeting than for the EKF sensor management; this confirms the analysis in section 3.2. In other words, in the problem of targeting sensor networks in an environment governed by a large-scale nonlinear dynamics, the backward formulation is preferable over the forward formulation since the cost of covariance update is significant for the large number of ensembles needed to accurately represent the complex system.

Note also that the T_F/T_B values can be predicted with reasonable accuracy. Using

$\tau_p \approx \frac{1}{6}p(p+1)(2p+1) + p$ `flops`, which applies better for small values of p, and the other parameters as given previously, T_F/T_B for the EKF tracking example ($n = 6$) is estimated as 3.6 while the actual ratio is 4.2. Likewise, for the EnSRF targeting case, using $n = 3$, $\widehat{T}_F/\widehat{T}_B$ is estimated as 1347 where the actual ratio is 796, which corresponds to 70% error. This difference is mainly caused by error in estimating τ_n for small n since `flops` counting does not consider the overhead computation expense such as memory access. However, this level of accuracy is sufficient to indicate the order of magnitude of improvement of the backward selection compared to the forward. For larger-scale problems in which the asymptotic assumptions are more accurate, the efficiency can be accurately estimated using the expressions given in Table 3.1.

3.5 Conclusions

This chapter presented a *backward formulation* for the large-scale sensor targeting problem using ensemble-based filtering. It was shown that this backward formulation provides an equivalent solution to a standard forward approach under some standard assumptions, but significantly reduces the number of ensemble updates that must be performed. The analysis of the computational efficiency of two approaches clearly showed that the backward selection is provably faster than the forward selection, and is particularly advantageous for large-scale systems.

The accomplishment in this chapter is expanded in the later chapters in the following ways: the targeting algorithms for coordination of multiple sensor platforms is developed in Chapter 4, quantification of the mutual information in the continuous-time domain is addressed in Chapter 5, and a sensitivity analysis of the ensemble-based targeting in this chapter is given in Chapter 6.

Chapter 4

Multi-Sensor Platform Targeting

This chapter addresses the coordination of multiple sensor platforms with constrained mobility in the targeting problems. The information reward for sensing paths is defined by the mutual information between the measurements at a finite set of sensing points and the future verification variables. Using the backward formulation presented in Chapter 3, this information reward can be computed without having to do a combinatorial number of computationally expensive ensemble updates.

The key aspects focused on in this chapter are: (a) an effective way of incorporating the vehicle mobility constraints into the backward selection process developed in Chapter 3, and (b) further improvements in efficiency to maintain computational tractability for large-scale decisions.

The limited vehicle mobility is addressed using the concept of the *reachable search space*, which is defined as the set of sensing points in space/time that can be reached by any admissible sequence of control actions, and utilizing the *action space search*, which enumerates possible control sequences instead of waypoint sequences. These two reduce the dimension of the search space leading to better computational efficiency.

For better efficiency in the decision making process, a cut-off heuristic utilizing a simple cost-to-go function is also introduced. The cost-to-go function based on some sparsity assumption of covariance matrices leads to a reduction of the number of calculations of the mutual information values. In addition, approximation techniques based on decomposition of a large-scale problem into small subproblems are presented

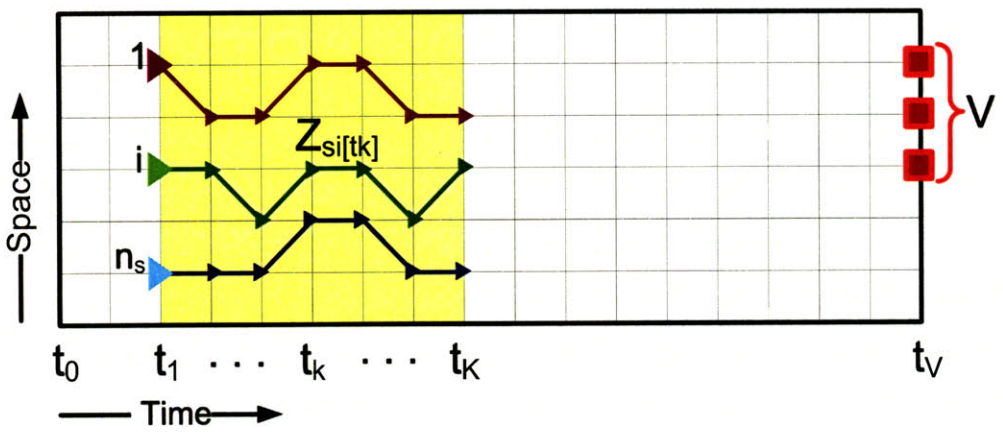

Figure 4-1: Multi-UAV targeting in the grid space-time: In contrast to the sensing-point targeting in Chapter 3, the motions of sensor platforms are constrained.

for better scalability.

Combining all these elements, this chapter presents an algorithmic framework for the multi-sensor platform targeting problems. In addition, numerical studies for comparison of various decomposition topologies demonstrate that a sequential decision topology better accounts for the spatial coupling of the information rewards compared to a local decision topology.

4.1 Problem Formulation

The multi-sensor platform targeting in this chapter considers the gridded space-time of finite dimension (Figure 4-1). In this grid space-time, the location of a vehicle can be described as a positive integer by having the index set defined such that

$$r = s + \Delta t \times n_G \tag{4.1}$$

when r and s represent two points that are spatially identical but temporally apart from each other by Δt time steps, which is positive when representing posteriority; n_G denotes the spatial dimension of the grid representation. In this chapter, the

search space-time $\mathcal{S} \subset \mathbb{Z}$ is defined be the set of location indices at which a sensor platform may be located.[1] Similar to the sensing point selection problem in section 3.1, it is assumed that a grid point s is associated with a single state variable X_s that can be directly measured; the measurement at this location is denoted as Z_s and $Z_s = X_s + W_s$ where W_s is the white Gaussian noise with variance R_s that is independent of sensing noise at other locations and of any state variable. Likewise, V represents the set of verification variables at verification time t_V. Denote the location index of i-th sensor platforms at time t_k as $s_i[t_k] \in \mathbb{Z}$, and the number of sensor platforms as n_s; also, let the set of locations of all the sensor platforms at time instance t_k be denoted as $\mathbf{s}[t_k] \in \mathbb{Z}^{n_s}$. The transition of a sensor platform's location at one time step to the next time step can be written as

$$s_i[t_{k+1}] = s_i[t_k] + u_i[t_k] \tag{4.2}$$

where $u_i[t_k] \in \mathcal{U} \subset \mathbb{Z}$ is the control action taken by the sensor platform i at time step t_k. The set $\mathcal{U}$ defines all the possible control options that can be represented as positive integer values in the grid space-time. Since a vehicle has limited capability of motion, $\mathcal{U}$ should be a finite set. Also, the control vector is defined as $\mathbf{u}[t_k] \triangleq [u_1[t_k],\ u_2[t_k],\ \ldots,\ u_{n_s}[t_k]]'$.

The goal of the multi-sensor platform targeting is to find the optimal control sequences $\mathbf{u}[t_k],\ k \in [1, K] \cap \mathbb{Z}$ (or equivalently optimal waypoint sequences $\mathbf{s}[t_k],\ \forall k \in [1, K] \cap \mathbb{Z}$) to maximize the mutual information between the verification variables V and the measurement sequences by all the sensor platforms:

$$\max_{\mathbf{u}[t_1],\cdots,\mathbf{u}[t_K]} \mathcal{I}(V; Z_{\mathbf{s}[t_1]}, \cdots, Z_{\mathbf{s}[t_K]}) \tag{4.3}$$

[1]This definition of $\mathcal{S}$ is equivalent to the definition in Chapter 1 with appropriate re-indexing of locations.

subject to

$$\mathbf{s}[t_{k+1}] = \mathbf{s}[t_k] + \mathbf{u}[t_k], \qquad \forall k \in [1, K-1] \cap \mathbb{Z} \tag{4.4}$$

$$\mathbf{s}[t_1] = \mathbf{s}_o = \text{given}, \tag{4.5}$$

and

$$s_i[t_k] \in \mathcal{S}, \qquad \forall i \in [1, n_s] \cap \mathbb{Z}, \ \forall k \in [1, K] \cap \mathbb{Z} \tag{4.6}$$

There are two types of constraints in this decision: The condition (4.4) describes the vehicle dynamics, when initial (at t_1) locations of vehicles are assumed to be given as in (4.5). With regard to the control options of the sensor platforms, this work particularly considers limited mobility of a sensor platform; thus, a legitimate control action leads the vehicle to one of its neighboring locations at the next time step. The condition (4.6) means that vehicles should remain in the grid space by their control actions - call it the *admissibility* condition.

Before moving on to the algorithm description, it needs to be pointed out that the formulation in (4.3) models the sensing process as "dwell and dash." A sensor makes an observation of the environmental variables by loitering over (or slowly passing by) a waypoint; then, it dashes to the next waypoint without sensing in-between. With this sensing model, the targeting problem in 4.3 addresses the decision of finding *information-rich* landmarks, which are sparsely distributed in time-space. This abstraction is valid in which the environmental dynamics is of larger time- and length-scales than vehicle dynamics, and therefore the environmental variables at the endpoints of a path are highly correlated with those in the middle of the path.

4.2 Algorithm

This section presents an algorithm to solve the multi-sensor platform targeting defined in section 4.1. The proposed targeting algorithm features four main aspects: a) the backward selection formulation in computing the mutual information value for the

solution candidates, b) the reachable search space and action space search to effectively incorporate vehicle mobility constraints, c) a cut-off heuristic that reduces the number of solution candidates for which the objective values are actually calculated. Details of these aspects are presented followed by a summary of the algorithm.

4.2.1 Backward Selection

The targeting algorithm employs the backward selection (BS) formulation presented in Chapter 3. Thus, the mutual information value in (4.3) is computed as

$$\begin{aligned} \mathcal{I}(V; Z_{\mathbf{s}[t_1:t_K]}) &= \mathcal{H}(Z_{\mathbf{s}[t_1:t_K]}) - \mathcal{H}(Z_{\mathbf{s}[t_1:t_K]}|V) \\ &= \tfrac{1}{2}\log\det\left(\mathbf{P}(Z_{\mathbf{s}[t_1:t_K]}) + R_{\mathbf{s}[t_1:t_K]}\right) \\ &\quad - \tfrac{1}{2}\log\det\left(\mathbf{P}(Z_{\mathbf{s}[t_1:t_K]}|V) + R_{\mathbf{s}[t_1:t_K]}\right) \end{aligned} \tag{4.7}$$

where $\mathbf{s}[t_1 : t_K] \triangleq \bigcup_{k=1}^{K} \mathbf{s}[t_k]$. In the EnSRF framework, the ensemble augmentation procedure in section 3.1.1 can be used to obtain the prior and the posterior ensembles for the search space $\mathcal{S}$, and the covariance matrices in (4.7) are computed using these ensembles as in (3.12).

4.2.2 Reachable Search Space

The targeting decision in this chapter is not simply a decision of picking a fixed number of points out of a fixed search space, because of the constraints in vehicle motions. In the backward selection framework, an ensemble update is needed to compute the posterior ensemble over the search space by fictitious measurements at V. Because the motions of sensor platforms are limited, not all of the points in $\mathcal{S}$ can be reached by the sensor platforms; therefore, a set smaller than $\mathcal{S}$ effectively works as the search space over which the posterior ensemble is calculated. This reduced search space should be as small as possible, but must be sufficiently exhaustive to consider all of the admissible and reachable candidates.

To find the smallest possible search space, first, define the one-step reachable set that represents the set of all possible location vectors that the team of sensor

platforms at location $\mathbf{s}[t_k]$ at time t_k can reach at the next time step t_{k+1} by applying a legitimate control vector:

$$\mathcal{R}(\mathbf{s}[t_k]) = \left\{ \mathbf{r} \subset \mathcal{S} : \mathbf{r} = \mathbf{s}[t_k] + \mathbf{u}[t_k];\ \mathbf{s}[t_k] \subset \mathcal{S}, \mathbf{u}[t_k] \subset \mathcal{U}^{n_s} \right\}. \tag{4.8}$$

Then, the *reachable search space* denoted as $\mathcal{S}_Z$ consisting of all the points in $\mathcal{S}$ that can be visited by some sensor platform by some control actions can be defined as

$$\mathcal{S}_Z = \bigcup_{k=1}^{K-1} \mathcal{R}^k(\mathbf{s}_o), \tag{4.9}$$

where $\mathbf{s}_o$ is the initial vehicle locations at time t_1 as defined in (4.5), and

$$\mathcal{R}^{k+1}(\mathbf{s}_o) = \mathcal{R}\left(\mathcal{R}^k(\mathbf{s}_o)\right), \qquad \forall k \in [1, K-1] \cap \mathbb{Z}. \tag{4.10}$$

This $\mathcal{S}_Z$ is minimal in the sense that every element in $\mathcal{S}_Z$ will be referred to at least once in computing the mutual information for a feasible solution candidate. Since the right-hand side of (4.9) is a union of disjoint sets, the cardinality of $\mathcal{S}_Z$ becomes

$$|\mathcal{S}_Z| = \sum_{k=1}^{K-1} |\mathcal{R}^k(\mathbf{s}_o)| \leq \sum_{k=1}^{K-1} |\mathcal{U}|^{n_s k} = \frac{|\mathcal{U}|^{n_s K} - |\mathcal{U}|^{n_s}}{|\mathcal{U}|^{n_s} - 1} \sim \mathcal{O}\left(|\mathcal{U}|^{n_s(K-1)}\right). \tag{4.11}$$

Note that the size of $\mathcal{S}_Z$ is exponential in n_s and $K-1$; thus, the optimal targeting problem with many sensor platforms for a long time horizon needs a larger $\mathcal{S}_Z$. Using the atomic time units defined in section 3.2.2, the computation cost of performing the ensemble update over $\mathcal{S}_Z$ takes

$$\texttt{TimeUpdate}_{\mathcal{S}_Z} = (|\mathcal{S}_Z|(|\mathcal{S}_Z| + M)\,\delta_{L_E}. \tag{4.12}$$

As an instance of the vehicle mobility constraints, this work considers the minimum and maximum speed of a sensor platform.

$$v_{\min} \leq \mathcal{D}_M(s_i[t_k], s_i[t_{k+1}]) \leq v_{\max}. \tag{4.13}$$

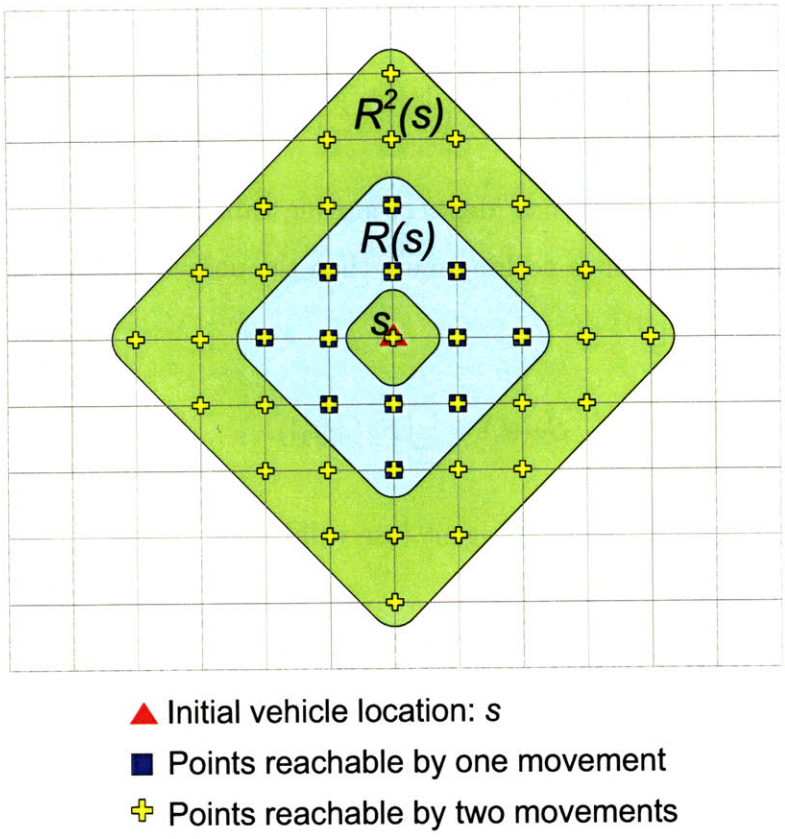

Figure 4-2: Reachable zones by a single agent in two-dimensional grid space

$\mathcal{D}_M$ denotes the Manhattan distance defined as

$$\mathcal{D}_M(s,r) = \sum_{j=1}^{d} |s^j - r^j| \tag{4.14}$$

where d denotes the dimension of spatial coordinates; s^j and r^j denote the j-th spatial coordinate values of the gridpoints s and r in $\mathcal{S}$, respectively. Figure 4-2 illustrates the reachable zones by a single agent from location s in a two-dimensional grid space, in case $v_{min} = 1$ and $v_{max} = 2$. Note in this case that

$$\mathcal{R}^k(s_o) = \Big\{ r \in \mathcal{S}[t_{k+1}] \,:\, \mathcal{D}_M(s,r) \leq k \times v_{max} \Big\}, \qquad \forall k \in [2, K-1] \cap \mathbb{Z} \tag{4.15}$$

where $\mathcal{S}[t_{k+1}] \subset \mathcal{S}$ denotes the set of gridpoints whose time index is t_{k+1}. Thus, the

size of $\mathcal{S}_Z$ for a single sensor in a two-dimensional space becomes

$$|\mathcal{S}_Z|_{single.2d} = \sum_{k=2}^{K-1} \left(1 + 4 \sum_{j=1}^{kv_{max}} j \right) + 4 \sum_{j=v_{min}}^{v_{max}} j \ . \tag{4.16}$$

In case each sensor platform has identical moving capability, the reachable zone by a team of sensor platforms is nothing more than the union of each agent's reachable zone:

$$\mathcal{S}_Z(\mathbf{s}_o) = \bigcup_{i=1}^{n_s} \mathcal{S}_Z(s_i[t_1]). \tag{4.17}$$

Then, the size of $\mathcal{S}_Z$ for the multi-agent case will be

$$|\mathcal{S}_Z| \leq n_s \times |\mathcal{S}_Z|_{single} \tag{4.18}$$

where equality holds when the reachable zones of the agents are disjoint. Note that conflicting assignment is not explicitly prohibited with this $\mathcal{S}_Z$; different sensor platforms might take measurements at the same location at the same time. However, the optimal targeting decision will never select this option, since it is always suboptimal:

$$\begin{aligned} \mathcal{I}(V; Z_s, Z_s) &= \mathcal{I}(V; Z_s) \\ &= \mathcal{H}(V) - \mathcal{H}(V|Z_s) \\ &\leq \mathcal{H}(V) - \mathcal{H}(V|Z_s, Z_q) \\ &= \mathcal{I}(V; Z_s, Z_q), \quad \forall s \neq q. \end{aligned} \tag{4.19}$$

4.2.3 Action Space Search

With fixed initial location vector $\mathbf{s}_o$, the targeting problem finds the optimal $\mathbf{s}[t_2 : t_K]$ consisting of $n_s(K-1)$ grid points. The reachable search space $\mathcal{S}_Z$ defines the set over which the ensemble update needs to be done; every point in $\mathcal{S}_Z$ can be reached by *some* control actions of some sensor platform. However, this does not mean that *every* set $\mathbf{s} \subset \mathcal{S}_Z$ with $|\mathbf{s}| = n_s(K-1)$ comprises some feasible waypoint sequences. Thus, a simple `n-choose-k`-type search considered in Chapter 3, is not an efficiency

way when the vehicle motions are constrained.

For this reason, this work suggests to search over the *action space* rather than the grid index space. In other words, the search process works as

1. Pick a feasible control sequence $\mathbf{u}[t_1 : t_K]$,
2. Find the corresponding location vector sequence $\mathbf{s}[t_1 : t_K]$,
3. Evaluate $\mathcal{I}(Z_{\mathbf{s}[t_1:t_K]}; V)$,

When using the action space search, the total number of solution candidates to consider becomes simply $|\mathcal{U}|^{n_s(K-1)}$; this is much smaller than $\binom{|\mathcal{S}_Z|}{n_s(K-1)}$ for the `n-choose-k`-type search, which could be $\mathcal{O}(|\mathcal{U}|^{n_s^2(K-1)^2})$ in the worst case.

It is conceivable that the action space search allows for dealing with more generalized problems than given in section 4.1. For instance, the problems of finding sensing paths with much denser temporal/spatial resolutions can be addressed in the same action space of dimension $\mathcal{O}(|\mathcal{U}|^{n_s^2(K-1)^2})$, if the control action is parameterized in terms of the start and end points.

4.2.4 Cut-Off Heuristics

Although BS does not involve a combinatorial number of ensemble updates, it still requires the computation of a determinant of

$$\mathbf{P}(Z_{\mathbf{s}[t_2:t_K]}|Z_{\mathbf{s}_o}), \quad \mathbf{P}(Z_{\mathbf{s}[t_2:t_K]}|V, Z_{\mathbf{s}_o}), \tag{4.20}$$

a total of $|\mathcal{U}|^{n_s(K-1)}$ times.[2] For instance, with the mobility constraint in (4.13),

$$|\mathcal{U}|^{n_s(K-1)} = \left[4 \sum_{j=v_{min}}^{v_{max}} j\right]^{n_s(K-1)} = \left[2(v_{max}+v_{min})(v_{max}-v_{min}+1)\right]^{n_s(K-1)}, \tag{4.21}$$

for a two-dimensional problem. Therefore, if v_{max} and/or $n_s(K-1)$ becomes large, the number of covariance matrices in (4.7) to be considered in the decision rapidly

[2]Since the measurement at initial time has common effect on all the solution candidates, the measurement update with $Z_{\mathbf{s}_o}$ can be done in advance of the solution search process.

grows. Moreover, as $n_s(K-1)$ becomes larger, the unit time for computing the determinant of one covariance matrix also increases - proportional to $n^3(K-1)^3$ when utilizing the Cholesky factorization. Thus, in order to address a large-scale problem, it is necessary to reduce the number of solution candidates whose mutual information values (equivalently, the determinants of prior and posterior covariance matrices) are actually calculated.

For this purpose, this thesis proposes a cut-off heuristic that provides an indication of which measurement choice would render a high information reward. For notational simplicity, the following notations are used in this section: $\mathbf{P}(X_{\mathbf{s}[t_2:t_K]}|Z_{\mathbf{s}_o}) \triangleq P^-_{\mathbf{s}[t_2:t_K]}$ and $\mathbf{P}(X_{\mathbf{s}[t_2:t_K]}|V, Z_{\mathbf{s}_o}) \triangleq P^+_{\mathbf{s}[t_2:t_K]}$. It was noted in Burer and Lee [43] that for a legitimate covariance matrix $P \succ 0$,

$$\log\det(P) \leq \log\det(P \circ \Sigma) \tag{4.22}$$

where $\circ$ denotes Haddamard (or, entry-wise) product, and Σ is a symmetric positive definite matrix satisfying

$$\mathtt{diag}(\Sigma) = \mathbf{1}. \tag{4.23}$$

With the simplest choice of $\Sigma = I$, upper bounds for the prior and posterior entropies for the candidate $\mathbf{s}$ will be

$$\begin{aligned}\mathcal{H}(Z_{\mathbf{s}[t_2:t_K]}|Z_{\mathbf{s}_o}) - \frac{n_s(K-1)}{2}\log(2\pi e) &= \frac{1}{2}\log\det\left(P^-_{\mathbf{s}[t_2:t_K]} + R_{\mathbf{s}[t_2:t_K]}\right) \\ &\leq \frac{1}{2}\sum_{i=1}^{n_s(K-1)} \log\left[P^-_{\mathbf{s}[t_2:t_K]}(i,i) + R_{\mathbf{s}[t_2:t_K]}(i,i)\right] \\ &\triangleq \widetilde{\mathcal{H}}^-_{\mathbf{s}[t_2:t_K]}\end{aligned} \tag{4.24}$$

$$\begin{aligned}\mathcal{H}(Z_{\mathbf{s}[t_2:t_K]}|V, Z_{\mathbf{s}_o}) - \frac{n_s(K-1)}{2}\log(2\pi e) &= \frac{1}{2}\log\det\left(P^+_{\mathbf{s}[t_2:t_K]} + R_{\mathbf{s}[t_2:t_K]}\right) \\ &\leq \frac{1}{2}\sum_{i=1}^{n_s(K-1)} \log\left[P^+_{\mathbf{s}[t_2:t_K]}(i,i) + R_{\mathbf{s}[t_2:t_K]}(i,i)\right] \\ &\triangleq \widetilde{\mathcal{H}}^+_{\mathbf{s}[t_2:t_K]}.\end{aligned} \tag{4.25}$$

Note that calculation of $\widetilde{\mathcal{H}}^{-}_{\mathbf{s}[t_2:t_K]}$ and $\widetilde{\mathcal{H}}^{+}_{\mathbf{s}[t_2:t_K]}$ is computationally much cheaper than the original entropy computation, since it only requires simple scalar operations. Based on the above upperbounds, this thesis proposes using

$$\widetilde{\mathcal{I}}_{\mathbf{s}[t_2:t_K]} = \widetilde{\mathcal{H}}^{-}_{\mathbf{s}[t_2:t_K]} - \widetilde{\mathcal{H}}^{+}_{\mathbf{s}[t_2:t_K]} \tag{4.26}$$

as an indication of whether the candidate $\mathbf{s}[t_2 : t_K]$ is worth being considered in the mutual information computation. Suppose that $\mathbf{s}[t_2 : t_K]$ is the m-th candidate in the list, then the cut-off decision for $\mathbf{s}[t_2 : t_K]$ is made by

$$\mathbb{I}_{co}(\mathbf{s}[t_2 : t_K]) = \begin{cases} \texttt{pass}, & \text{if } \widetilde{\mathcal{I}}(\mathbf{s}[t_2 : t_K]) \geq \mathcal{I}_{LBD} \\ \texttt{fail}, & \text{otherwise,} \end{cases} \tag{4.27}$$

where $\mathcal{I}_{LBD}$ is the tightest lowerbound on the optimal information reward based on the previous $(m-1)$ solution candidates. The actual mutual information of candidate $\mathbf{s}[t_2 : t_K]$ is computed, only if $\mathbb{I}_{co}(\mathbf{s}[t_2 : t_K]) = \texttt{pass}$.

It should be noted that the relations in (4.24) and (4.25) can also be described in information-theoretic terms. Suppose that $\mathbf{s}[t_2 : t_K] = [r_1, r_2, \ldots, r_{n_s(K-1)}]'$, then

$$\mathcal{H}(Z_{\mathbf{s}[t_2:t_K]}|Z_{\mathbf{s}_o}) = \mathcal{H}(Z_{r_1}|Z_{\mathbf{s}_o}) + \sum_{i=2}^{n_s(K-1)} \mathcal{H}(Z_{r_i}|Z_{\mathbf{s}_o}, Z_{r_1}, \ldots, Z_{r_{i-1}}) \tag{4.28}$$

$$\leq \mathcal{H}(Z_{r_1}|Z_{\mathbf{s}_o}) + \sum_{i=1}^{n_s(K-1)} \mathcal{H}(Z_{r_i}|Z_{\mathbf{s}_o}), \tag{4.29}$$

because the marginal entropy of a random variable is always greater than equal to its conditional entropy. The similar inequality relation can be derived for $\mathcal{H}(Z_{\mathbf{s}[t_2:t_K]}|V, Z_{\mathbf{s}_o})$ using the information-theoretic notion.

Also note that if

$$\widetilde{\mathcal{I}}_{\mathbf{s}[t_2:t_K]} \leq \mathcal{I}(Z_{\mathbf{s}[t_2:t_K]}; V) \tag{4.30}$$

for all $\mathbf{s}[t_2 : t_K]$, the optimal solution is guaranteed to lie amongst the **pass**-ed candidates. Notice that if the mutual information $\mathcal{I}(Z_{\mathbf{s}}; V)$ is submodular, then (4.30) is satisfied. However, as will be discussed in Appendix A, the information reward

in this work is not submodular. Thus, the search process using the cut-off heuristic in (4.27) does not guarantee that it will find the optimal solution. However, it can be experimentally verified that $\widetilde{\mathcal{I}}_{\mathbf{s}[t_2:t_K]}$ is a sufficiently good approximation of the true mutual information $\mathcal{I}(Z_{\mathbf{s}[t_2:t_K]}; V, Z_{\mathbf{s}_o})$. Thus, this thesis proposes the following modification to the cut-off indicator function to improve the likelihood of finding the optimal solution:

$$\mathbb{I}^{\epsilon}_{co}(\mathbf{s}[t_2:t_K]) = \begin{cases} \texttt{pass}, & \text{if } \widetilde{\mathcal{I}}(\mathbf{s}[t_2:t_K]) \geq \mathcal{I}_{LBD} - \epsilon \\ \texttt{fail}, & \text{otherwise,} \end{cases} \tag{4.31}$$

with a positive relaxation parameter ϵ. Thus, every time a candidate $\mathbf{s}[t_2:t_K]$ is being considered, its approximate cost-to-go $\widetilde{\mathcal{I}}(\mathbf{s}[t_2:t_K])$ and cut-off indicator $\mathbb{I}^{\epsilon}_{co}(\mathbf{s}[t_2:t_K])$ are evaluated; the actual payoff value is computed if it `pass`-es the cut-off criterion in (4.31).

An alternative way of implementing the idea of cut-off is to postpone the cut-off decision until after the calculation of the approximate mutual information values of all the candidates. By evaluating the actual information reward in the order of the largest cost-to-go heuristics, $\mathcal{I}_{LBD}$ is likely to be tighter; therefore, more candidates will be cut-off. However, this way requires more memory space than making the cut-off decision every time the cost-to-go is calculated.

A more accurate approximation of $\mathcal{I}(Z_{\mathbf{s}[t_2:t_K]}; V)$ than $\widetilde{\mathcal{I}}_{\mathbf{s}[t_2:t_K]}$ in (4.26) can be obtained either by utilizing more sophisticated Σ such as block-diagonal and tri-diagonal, or by extending the information-theoretic discussion in (4.29) as

$$\mathcal{H}(Z_{\mathbf{s}[t_2:t_K]}|Z_{\mathbf{s}_o}) = \mathcal{H}(Z_{r_1}|Z_{\mathbf{s}_o}) + \sum_{i=2}^{n_s(K-1)} \mathcal{H}(Z_{r_i}|Z_{\mathbf{s}_o}, Z_{r_1}, \ldots, Z_{r_{i-1}}) \tag{4.32}$$

$$\leq \mathcal{H}(Z_{r_1}|Z_{\mathbf{s}_o}) + \sum_{i=2}^{n_s(K-1)} \mathcal{H}(Z_{r_i}|Z_{\mathbf{s}_o}, Z_{r_{i-1}}). \tag{4.33}$$

The use of more accurate approximations can cut-off more suboptimal solution candidates; but calculation of individual cost-to-go becomes more expensive.

Algorithm 1 Ensemble-Based Multi-Sensor Platform Targeting Algorithm

1: Initialize $\mathcal{I}_{LBD}$ to zero.
2: Determine $\mathcal{S}_Z$ from $\mathbf{s}_o$, v_{min}, and v_{max}
3: Obtain prior perturbation ensembles corresponding to $\mathcal{S}_Z \cup V$, $\widetilde{\mathbf{X}}_{X_{\mathcal{S}_Z \cup V}}$
4: From $\widetilde{\mathbf{X}}_{X_{\mathcal{S}_Z \cup V}}$, extract $\widetilde{\mathbf{X}}_{X_{\mathcal{S}_Z}}$ and compute the posterior ensemble $\widetilde{\mathbf{X}}_{X_{\mathcal{S}_Z}|V}$
5: Update with the prior and posterior ensembles with initial measurement $Z_{\mathbf{s}_o}$.
6: Compute mutual information for the initial measurement $\mathcal{I}(Z_{\mathbf{s}_o}; V)$
7: **for all** $\mathbf{u}[t_1 : t_{K-1}] \in \mathcal{U}^{n_s(K-1)}$ **do**
8: $\quad \mathbf{s}[t_k] \in \mathcal{S}_Z$, $\forall k$? If yes, proceed; otherwise, goto step 7.
9: $\quad$ Evaluate covariance matrices, $\mathbf{P}(Z_{\mathbf{s}[t_2:t_K]}|Z_{\mathbf{s}_o})$ and $\mathbf{P}(Z_{\mathbf{s}[t_2:t_K]}|V, Z_{\mathbf{s}_o})$
10: $\quad$ Compute the heuristic cost-to-go $\widetilde{\mathcal{I}}_{\mathbf{s}[t_2:t_K]}$
11: $\quad$ **if** $\widetilde{\mathcal{I}}_{\mathbf{s}[t_2:t_K]} > \mathcal{I}_{LBD} - \epsilon$ **then**
12: $\quad\quad$ Compute $\mathcal{I}(Z_{\mathbf{s}[t_2:t_K]}; V|Z_{\mathbf{s}_o})$
13: $\quad\quad$ **if** $\mathcal{I}(Z_{\mathbf{s}[t_2:t_K]}; V|Z_{\mathbf{s}_o}) > \mathcal{I}_{LBD}$ **then**
14: $\quad\quad\quad \mathcal{I}_{LBD} = \mathcal{I}(Z_{\mathbf{s}[t_2:t_K]}; V|Z_{\mathbf{s}_o})$
15: $\quad\quad\quad \mathbf{u}_{LBD} = \mathbf{u}[t_1 : t_{K-1}]$
16: $\quad\quad$ **end if**
17: $\quad$ **else**
18: $\quad\quad$ Goto step 7.
19: $\quad$ **end if**
20: **end for**
21: Return $\mathbf{u}^\star[t_1 : t_{K-1}] = \mathbf{u}_{LBD}$ and $\mathcal{I}(Z_{\mathbf{s}^\star[t_1:t_K]}; V) = \mathcal{I}(Z_{\mathbf{s}_o}; V) + \mathcal{I}_{LBD}$

4.2.5 Algorithm Summary

Algorithm 1 demonstrates the overall procedure of the ensemble-based multi-sensor platform targeting presented in this section, specifically the cheapest implementation that uses the cost-to-go in (4.26) and calculates the actual objective value for every pass-ed candidate. Other types of cost-to-go heuristics and cut-off procedures can easily be incorporated with a slight modification of Algorithm 1, although all the numerical results in this thesis will be based on this version of algorithm.

It should be pointed out that other constraints such as communication budget and power budget can also be involved in the targeting decision, and the algorithm in this Chapter can easily be modified to address those constraints.

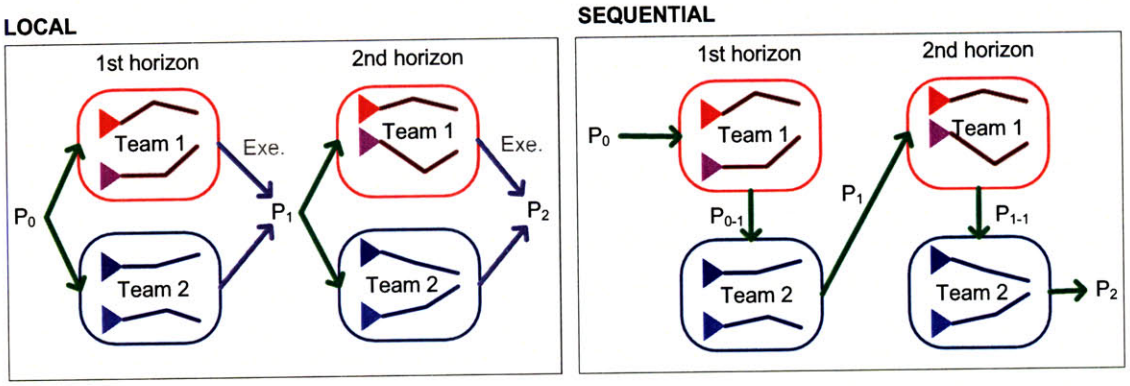

Figure 4-3: Schematics of team-based decomposition with two different communication topologies

4.3 Decomposition Schemes

The proposed cut-off heuristics can reduce the number of computations of matrix determinant; however, it still requires calculation of cost-to-go values combinatorially many times. Thus, a further approximation scheme that breaks down the original problem into a set of small pieces of problems is needed to address a larger-scale problem. This section presents a team-based receding horizon approximation of the original targeting problem. In addition to speeding up the computation, this decomposition provides useful implications on the distributed decision making for the targeting approach.

Recall that the search process of targeting starts with two covariance matrices - $\mathbf{P}(X_{\mathcal{S}_Z})$ and $\mathbf{P}(X_{\mathcal{S}_Z}|V)$; the impact of measurements taken at the initial vehicle locations can be incorporated in advance. Before starting a search process, the following two covariances are available.

$$\mathbf{P}(X_{\mathcal{S}_Z}|Z_{\mathbf{s}_o}), \text{ and } \mathbf{P}(X_{\mathcal{S}_Z}|V, Z_{\mathbf{s}_o}). \tag{4.34}$$

The decomposition approach in this section is applied after computing the above two *a priori* covariances.

Figure 4-3 illustrates the schematics of the problem decomposition in case of two teams. Instead of finding $n_s(K-1)$ control actions at once, a set of subproblems

are considered in which n_t sensor platforms having an n_H-long lookahead window. Assuming that n_T divides n_s and n_H divides $K-1$ for simplicity, the number of subproblems becomes $\frac{n_s(K-1)}{n_T n_H}$.

The approximation scheme starts with solving the first subproblem that determines the optimal control actions of the first n_T platforms during the first n_H time steps (i.e. for the time window $[t_2, t_{n_H+1}]$ as the initial location vector $\mathbf{s}_o$ is fixed.) – decision making for team 1 for the first planning horizon in Figure 4-3. Afterwards, the second subproblem associated with the next n_T agents with the same time window is taken into account. There are a total of n_s/n_T subproblems for the first horizon; as soon as decisions for all the sensor platforms within the first planning window have been made, the "execution step," in which covariance information is updated incorporating those decisions will follow. From the implementation point of view, the existence of the execution step implies that the actual time interval between the first and next planning horizon is sufficiently large to incorporate all of the decisions made in the first horizon and to share the updated information. Thus, the $(\frac{n_s}{n_T}+1)$-th subproblem determines the optimal control actions for agents 1 to n_T for the next time window $[t_{n_H+2}, t_{2n_H+1}]$ based on the updated covariances:

$$\mathbf{P}(X_{\mathcal{S}_Z}|Z_{\mathbf{s}[t_2:t_{n_H+1}]}, Z_{\mathbf{s}_o}), \text{ and } \mathbf{P}(X_{\mathcal{S}_Z}|V, Z_{\mathbf{s}[t_2:t_{n_H+1}]}, Z_{\mathbf{s}_o}). \tag{4.35}$$

This sequence of decision making and execution continues in the temporal direction until the final planning horizon $[t_{K-n_H+1}, t_K]$.

Regarding the transition from the first subproblem to the second subproblem, this work considers two types of transition in this direction: *local* and *sequential* schemes. In the local scheme, a decision of each team is made without any knowledge about the other teams' decisions. In decision making for the planning window $[t_{kn_H+2}, t_{(k+1)n_H+1}]$, every team determines its best control actions by treating the

following matrices as *a priori* ones:

$$\mathbf{P}(X_{\mathcal{S}_Z}|Z_{\mathbf{s}[t_2:t_{kn_H+1}]}, Z_{\mathbf{s}_o}), \text{ and} \tag{4.36}$$

$$\mathbf{P}(X_{\mathcal{S}_Z}|V, Z_{\mathbf{s}[t_2:t_{kn_H+1}]}, Z_{\mathbf{s}_o}). \tag{4.37}$$

On the contrary, in the sequential scheme each team knows the decisions of preceding teams and incorporates them into its decision. In other words, a team of sensors $(mn_T + 1)$ to $(m + 1)n_T$ will compute their best control decisions based on the covariance matrices:

$$\mathbf{P}(X_{\mathcal{S}_Z}|Z_{s_{\{1:mn_T\}}[t_{kn_H+2}:t_{(k+1)n_H+1}]}, Z_{\mathbf{s}[t_2:t_{kn_H+1}]}, Z_{\mathbf{s}_o}), \text{ and} \tag{4.38}$$

$$\mathbf{P}(X_{\mathcal{S}_Z}|V, Z_{s_{\{1:mn_T\}}[t_{kn_H+2}:t_{(k+1)n_H+1}]}, Z_{\mathbf{s}[t_2:t_{kn_H+1}]}, Z_{\mathbf{s}_o}). \tag{4.39}$$

where $Z_{s_{\{1:mn_T\}}[t_{kn_H+2}:t_{(k+1)n_H+1}]} \triangleq \bigcup_{i=1}^{mn_T} Z_{s_i[t_{kn_H+2}:t_{(k+1)n_H+1}]}$.

Figure 4-3 contrasts the information flows for these two spatial decomposition schemes. The distinction between the local and the sequential schemes is related to the communication topology amongst agents. The local scheme assumes that only inner-team communication is available during a given decision horizon, while the sequential scheme assumes the existence of inter-team communication. Note that decisions for members in the same team are free of conflicts due to (4.19), but decisions for members in different teams can be in conflict, in particular, with the local scheme.

4.4 Numerical Simulations

4.4.1 Setup

For validation of the proposed algorithm, numerical simulations with the two-dimensional Lorenz-2003 model with $\rho_x = \rho_y = 2$ are considered. This model has 72 longitudinal grids and 17 latitudinal grids, and the size of each grid corresponds to 347km $\times$ 347km. The routine network with size 186 is assumed to be already deployed over the grid space, which is depicted with black * in Figures 4-4 and 4-5. The static network

is dense in two portions of the grid space representing lands, while it is sparse in the other two portions of the space representing oceans. The routine network takes measurement every 0.05 time unit (6 hrs in real). The leftmost part consisting of 27 grid points in the right land is selected as the verification region, over which the forecast uncertainty reduction 0.6 time units (~3 days) after the targeting time is interested in; it is plotted with red □ in the figures. The planning window $[t_1, t_K]$ is from 0.025 time unit(~3 hours) through 0.375 time unit (~15 hours) after t_0 with $K = 5$; time steps are equally spaced by 0.025 time units (~3 hours). Thus, the targeting problems is posed for 12hr missions from $t_1 = 3$hr to $t_K = 15$hr. The analysis ensemble at t_0 with size 1224, which is same as the size of the grid space, is obtained by running an EnSRF incorporating the routine observations for 1000 cycles. The contour plots in Figures 4-4 and 4-5 shows a typical shape of the error variance field with this routine configuration. The measurement noise variance is $R_{routine} = 0.2^2$ for routines and $R_s = 0.02^2$ for additional observations, assuming that high-resolution sensors will be equipped on the mobile platforms. Flight speed of the sensor platform is limited as $v_{min} = 1$ grid/timestep and $v_{max} = 2$ grids/timestep, which correspond to $v_{\min} = 116$km/hr, and $v_{\max} = 232$km/hr.

4.4.2 Effect of Cut-Off Heuristics

As pointed out in section 4.2.4, the cost-to-go heuristics $\tilde{\mathcal{I}}(\mathbf{s}[t_2 : t_K])$ can underestimate the actual mutual information; the relaxing parameter ϵ is needed for better optimality. In order to verify the computational effectiveness of the heuristics and to figure out the effect of the relaxing parameter on the solution optimality, the following two-sensor targeting problem is considered. Two sensors are initially located at $(45, 7)$ and $(45, 9)$ in the Lorenz-2003 grid space, which are in the middle of the eastern part of the wider ocean (representing the Pacific ocean). A Monte-Carlo simulation with 5 different initial ensembles – five different $\mathbf{X}^a(t_0)$ – is performed.

Table 4.1 shows the average mutual information value, the average number of candidates whose actual rewards are computed, and the computation time, with respect to different values of ϵ. $\epsilon = \infty$ means no cut-off decision, while $\epsilon = 0$ means no

Table 4.1: 5-simulation average performance of the cut-off heuristics for different ϵ

ϵ	$\mathcal{I}(Z_{\mathbf{s}^\star[t_1:t_K]}; V)$	# mutual info comp. ($\times 10^3$)	Comp. time (sec)
0	6.185	454	545
0.1	6.193	782	547
0.2	6.198	1319	551
1.0	6.199	31847	754
∞	6.199	401688	3187

relaxation of the cut-off criterion. It is first noted that the cut-off heuristics enormously reduces the number of candidates to be considered in the mutual information calculation, while optimality degradation is very small. In fact, for four out of five cases, the cut-off heuristics does not cause any performance degradation even with $\epsilon = 0$. As $\epsilon = 0.2$ gives a sufficiently close answer to the optimum without substantial increase of computation time, this value of ϵ is used for later simulations. It is also seen that improvement in the actual computation time is relatively small compared to the reduction in the number of candidates; this is because the cut-off heuristic requires additional computational resources to evaluate the cost-to-go function for every candidate. This implies that more complicated cost-to-go functions might not be preferred, because they could result in longer computational times due to increase in time spent for calculation of individual cost-to-go values.

4.4.3 Comparison of Decomposition Schemes

In order to discuss the validity of the decomposition schemes, simulation studies varying the values of n_T and n_H, and altering the inter-team communication scheme, are conducted. A problem with $n_s = 4$ and $K - 1 = 4$, which would take thousands of years to find the optimal solution by an exhaustive search on a usual PC, is considered.

Regarding the initial locations of the sensor platforms, two configurations in Table 4.2 are considered. The first configuration represents the case where sensors are located relatively close to each other at the initial time, while sensors are more dispersed in the second configuration. Three values of $n_T = 1, 2, 4$ and three values

Table 4.2: Initial Locations

Config #	sensor-1	sensor-2	sensor-3	sensor-4	Avg. initial dist. (km)
1	(45,7)	(45,9)	(46,7)	(46,9)	606
2	(48,3)	(47,7)	(47,11)	(47,15)	2326

Table 4.3: Average value of $\mathcal{I}(Z_{\mathbf{s}^*[t_1:t_K]}; V)$ for 4 sensors with dense configuration ($\mathcal{I}(Z_{\mathbf{s}^{rnd}[t_1:t_K]}; V) = 6.5098$)

	$n_H = 1$	$n_H = 2$	$n_H = K-1 = 4$
$n_T = 1_L$	6.4419	6.1147	6.0655
$n_T = 1_S$	9.9142	9.9749	10.2611
$n_T = 2_L$	7.5253	7.7506	8.2679^{co}
$n_T = 2_S$	9.9947	10.1854	10.5615^{co}
$n_T = n_s = 4$	10.0064	10.3363^{co}	N/A

of $n_H = 1, 2, 4$ are taken into account; in case $n_T = 1, 2$, both local and sequential communication schemes are considered, while $n_T = 4$ means full spatial cooperation across the entire fleet. Thus, a total of 15 different settings with various n_T and n_H, and with either local or sequential are compared with each other as well as with a feasible random strategy; for the cases of $(n_T, n_H) = (4, 2), (2, 4)$, the cut-off heuristics with $\epsilon = 0.2$ is implemented. Recall that the full optimal solution equivalent to $(n_T, n_H) = (4, 4)$ cannot be obtained within a reasonable amount of time. Monte-Carlo simulations are performed using 10 different initial ensembles.

Looking at the results for Configuration 1, Table 4.3 shows the average performance of each strategy. The superscript '*co*' in the table represents the use of the cut-off heuristics, and the subscripts 'L' and 'S' mean the use of local and sequential communication scheme, respectively. It is first noticeable that the local communication causes significant performance degradation. All of the three cases with $n_T = 1_L$ reveal inferior performance even to the reference random strategy. Furthermore, increase of the planning horizon deteriorates optimality in these cases; this can be interpreted as selecting many points based on outdated information is exceedingly suboptimal. Another noticeable comparison is between $n_T = 2_L$ and $n_T = 1_S$: al-

though each team consists of more sensors for $n_T = 2_L$, the lack of inter-team communication results in poorer optimality. In contrast to the cases of $n_T = 1_L$, increase of the team size and/or the horizon size enhances optimality for the cases of sequential schemes.

Figure 4-4 shows one exemplary targeting solution, contrasting two sets of sensing paths corresponding to strategies $n_T = 1_L, n_H = 4$ (left) and $n_T = 2_S, n_H = 4$ (right). These two setups provide the worst and the best performance, respectively. In the left picture, due to the lack of inter-sensor communication all four sensors collapse to one trajectory that must be the single-sensor optimal (but obviously multi-sensor suboptimal) solution, while sensors in the right picture are exploring a relatively large portion of the space.

The average computation time for each strategy is tabulated in Table 4.4. For every case except the three cases with $n_T n_H = 8$, the targeting decision takes just about 3 to 6 seconds. Computation time for the remaining three cases are all about a thousand seconds. Compared with extremely long expected computation time (about thousands of years) for the full optimization problem, it can be verified that decomposition schemes considerably enhance computational effectiveness.

Table 4.5 and Figure 4-5 show the results for the second configuration in which sensors are relatively distant to each other at the initial time. The performance gap between the local strategies and the sequential strategies is smaller in this configuration because the chance of conflicting assignment is reduced because of the initial dispersion of the agents. However, the qualitative message regarding the communication topology and the size of planning horizon is consistent with that of Configuration 1: a) with $n_T = 1_L$, increase of n_H deteriorates performance, and b) $n_T = 1_S$ outperforms $n_T = 2_L$. It is found in Figure 4-5 that multiple sensors are assigned to the same location at the same time – same colors of different markers, in case of the local scheme (on the left).

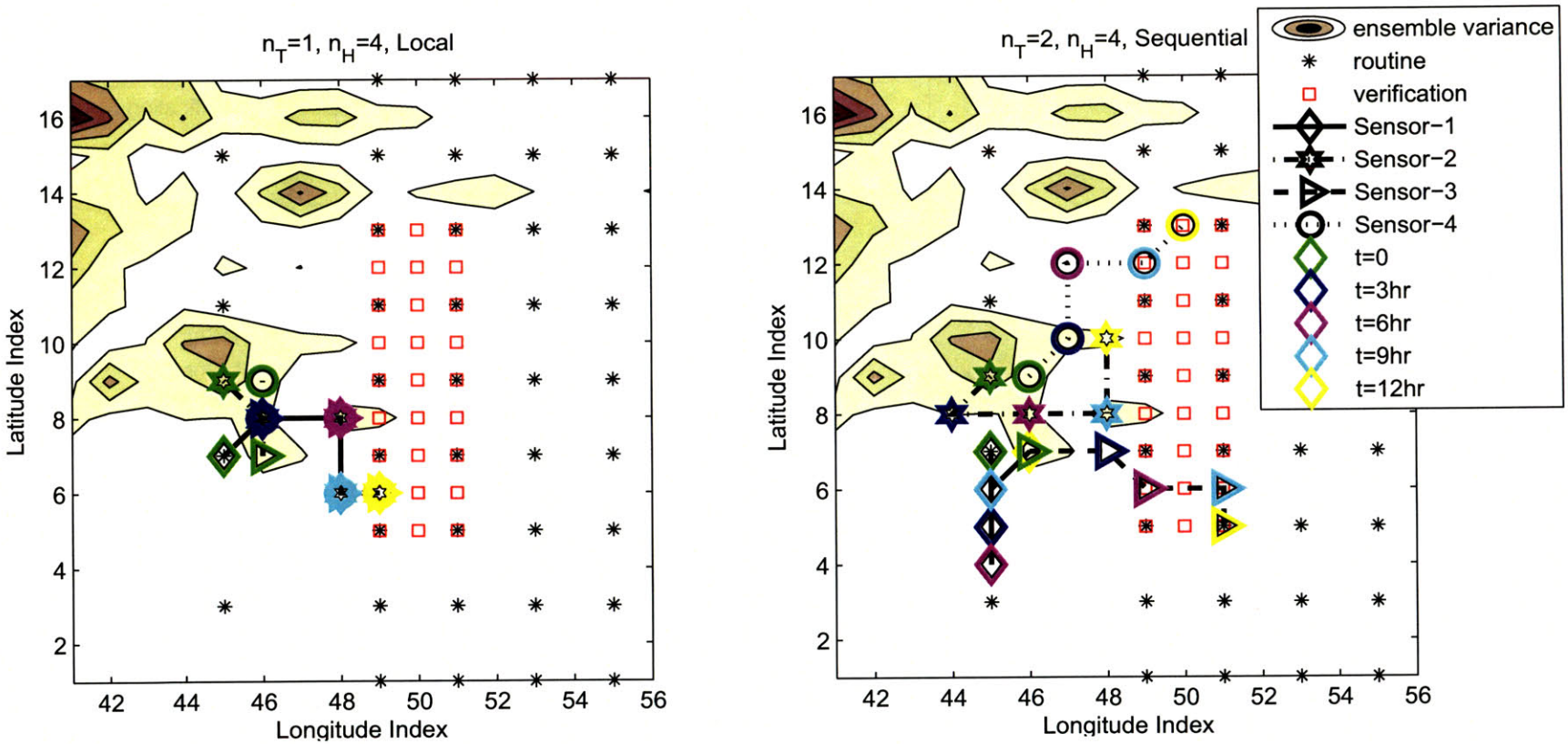

Figure 4-4: Targeting solutions for the sensors initially close to each other: Different markers represent different sensors and different colors represent the waypoints at different time instances. The starting locations are in green. Local decisions (left) lead to collapsed sensing paths providing poor performance, while sequential decisions (right) create paths for wide exploration.

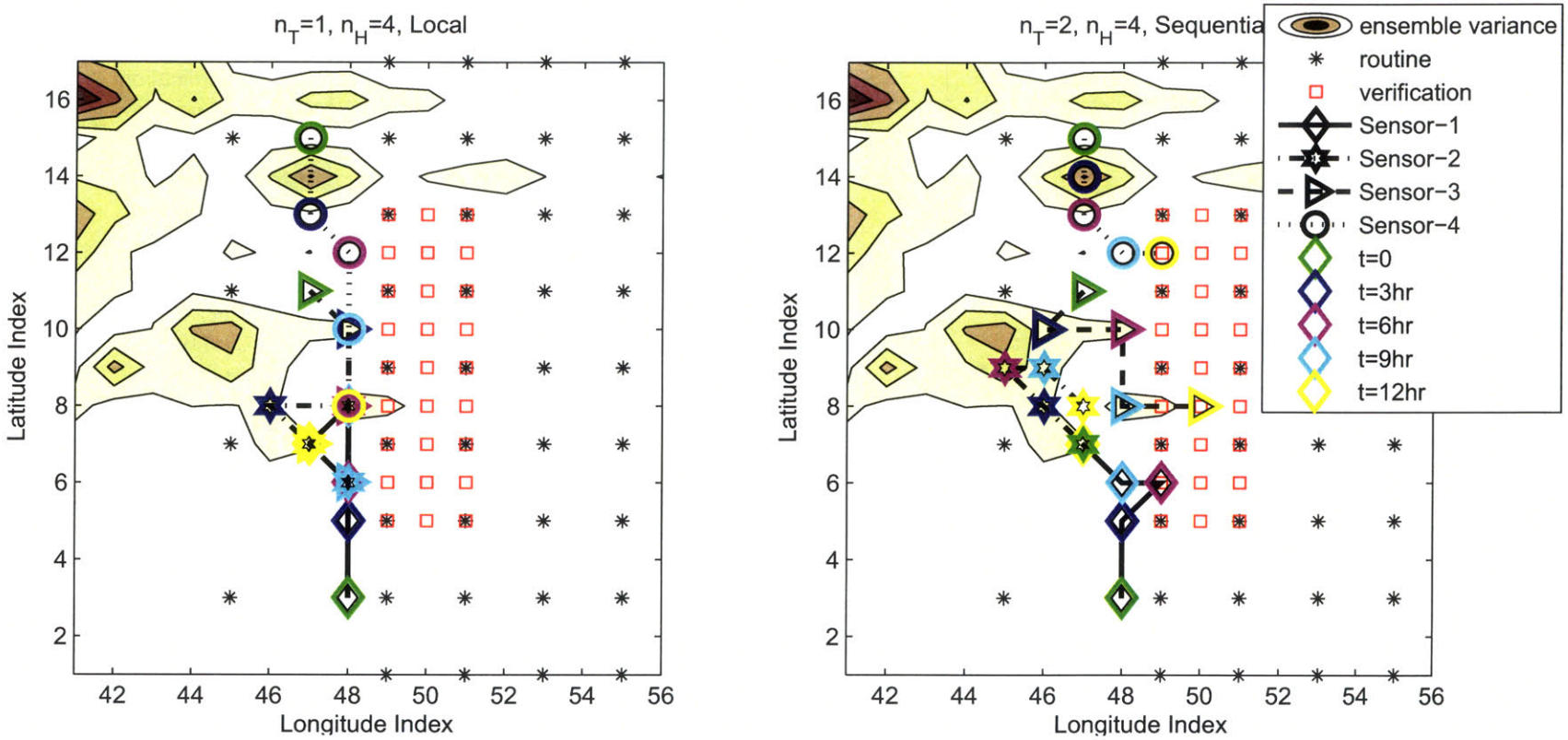

Figure 4-5: Targeting solutions for the sensors initially dispersed widely: Different markers represent different sensors and different colors represent the waypoints at different time instances. The starting locations are in green. Local decisions (left) still lead to multiple sensors taking measurements at the same location/time.

Table 4.4: Average computation time (seconds) for 4-sensor problems

	$n_H = 1$	$n_H = 2$	$n_H = K - 1 = 4$
$n_T = 1_L$	3.8	3.0	3.1
$n_T = 1_S$	5.9	4.4	4.1
$n_T = 2_L$	3.9	3.7	1088.0^{co}
$n_T = 2_S$	5.3	4.8	1075.0^{co}
$n_T = n_s = 4$	4.7	1001.9^{co}	N/A

Table 4.5: Average value of $\mathcal{I}(Z_{\mathbf{s}^*[t_1:t_K]}; V)$ for 4 sensors with dispersed configuration ($\mathcal{I}(Z_{\mathbf{s}^{rnd}[t_1:t_K]}; V) = 5.7066$)

	$n_H = 1$	$n_H = 2$	$n_H = K - 1 = 4$
$n_T = 1_L$	8.7646	8.5795	7.4195
$n_T = 1_S$	9.5129	9.7168	9.6681
$n_T = 2_L$	9.1316	9.2557	9.2071^{co}
$n_T = 2_S$	9.5757	9.8560	10.2609^{co}
$n_T = n_s = 4$	9.8511	9.9705^{co}	N/A

Table 4.6: Average value of $\mathcal{I}(Z_{\mathbf{s}^*[t_1:t_K]}; V)$ for 4-sensor case with smaller routine networks ($\mathcal{I}(Z_{\mathbf{s}^{rnd}[t_1:t_K]}; V) = 1.8037$)

	$n_H = 1$	$n_H = 2$	$n_H = K - 1 = 4$
$n_T = 1_L$	3.4065	3.2438	3.1887
$n_T = 1_S$	3.8024	4.0597	4.2118
$n_T = 2_L$	3.6649	3.6939	3.9021^{co}
$n_T = 2_S$	3.8177	4.0276	4.2499^{co}
$n_T = n_s = 4$	3.9028	4.1575^{co}	N/A

4.4.4 Effect of Routine Networks

In addition, this numerical study also investigates the influence of the static observation network. By deleting six routine observation sites in the oceanic region, the average and maximum squared analysis error are increased approximately eight times compared to the case in section 4.4.3. Table 4.6 represents the performance of each strategy. It is first noted that information rewards are generally smaller than those with a denser routine network; this implies that targeting is of less help when the prior information is inaccurate. Also, the performance gaps amongst strategies decrease as the overall performance levels decrease; but, the qualitative features as to the communication topology are still coherent.

4.5 Conclusions

This chapter presented a targeting algorithm for multiple sensor platforms with limited mobility, which featured the backward selection algorithm and the cut-off heuristics for enhancement of computational efficiency. Several decomposition schemes that break down a large-scale problem into small pieces of subproblems were proposed, with pointing out their interpretation in terms of sensor coordination. Numerical simulations using the Lorenz-2003 model verified the proposed targeting algorithm and quantitatively supported the importance of coordinated information sharing amongst sensor platforms.

In the later part of this thesis, planning of motions of sensor platforms in the abstraction of smaller time- and length-scales will be addressed (Chapter 5) with a continuous representation of vehicle paths and information rewards.

Chapter 5

Continuous Sensor Motion Planning

Chapters 3 and 4 addressed the discrete targeting problem to find information-rich landmarks, and these decisions were associated with relatively large time- and length-scale (e.g. examples in Chapter 4 considered a 12hr mission with waypoints distributed every 3 hrs). The motion planning problem in this chapter considers much shorter time- and length-scale in the sense that it concerns the information continuously extracted along the motion of sensors, and determines the steering commands of the sensor platforms.

In this chapter, the mutual information in the continuous-time domain is quantified for linear (time-varying) systems that represent the short-term behavior of the original nonlinear dynamics of the environmental variables. This chapter first presents the *filter form*, which is a straight-forward extension of the previous work [20–22], by treating the forecast problem as a filtering problem with a longer time window. However, three reasons are then given for why this form might not be suitable for motion planning for a long-term forecast: sensitivity to linearization error, computational cost, and the lack of on-the-fly knowledge of the accumulated information.

The first two issues arise because the filter form requires integration of matrix differential equation for the time window $[0, T]$ where T is the verification time in the *far* future. Integration for a long time interval accumulates the linearization error,

and requires more computation resource than integration for a short time interval. The computation burden can be problematic, especially, when the filter form is used as a subroutine for an optimization process that needs to evaluate many different options of measurement choices.

Alternatively, this work suggests the *smoother form*, which regards forecasting as fixed-interval smoothing. The equivalence of the smoother form to the filter form is proven based on the conditional independence of the measurement history and the future verification variables for a given present state value. Utilizing the covariance equations for a two-filter type Kalman smoothing in Wall et al. [44], this chapter derives an expression for the mutual information in the forecast problem involving matrix differential equations that are only integrated over the planning time window. This reduction of the duration of integration mitigates the effect of the linearization error, and offers better computational efficiency.

Moreover, the smoother form simplifies the process of quantifying the information accumulated along the path on-the-fly. In the smoother form, the accumulated information can be calculated in parallel with the process of computing the ultimate information reward. This concurrent knowledge of information accumulation allows for exploring various path planning algorithms as well as plays an important role in receding-horizon approximations to the optimal motion planning (discussed in Appendix B). Thus, the smoother form resolves all three problems in the filter form and provides an efficient and more informative way of calculating the information reward along the path.

This work presents a general methodology of continuous motion planning for forecast problems by combining the smoother form for quantifying the information associated with a temporally continuous measurement, and the spatial interpolation-based representation of spatially continuous sensing paths. In addition to an optimal path planning formulation that maximizes the mutual information, a real-time steering law based on the spatial gradient of the time derivative of information is proposed. A numerical example of a linearized Lorenz-2003 model validates the proposed framework.

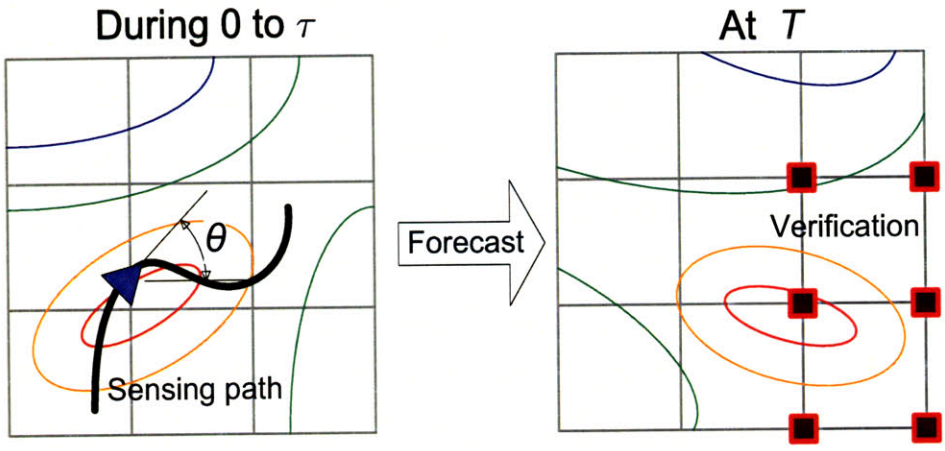

Figure 5-1: Continuous motion planning of a sensor for best information forecast: a mobile sensor senses some environmental variable shown in contour along the path designed to achieve best forecast for the verification region.

5.1 Problem Description

This chapter addresses continuous motion planning for mobile sensors to improve forecast over the specified region. In Figure 5-1, a mobile sensor *continuously* takes measurements of environmental field variables along a *continuous path* during the time interval $[0, \tau]$, while the quantity of interest is the forecast performance over the verification region at the verification time T. The objective of the motion planning is to design steering commands for the sensor that leads to the largest uncertainty reduction in the forecast, which is represented by the notion of *mutual information* in the continuous-time domain. In the context of adaptive sampling for weather prediction, this motion planning problem can be posed as decisions over some local search space with an appropriate length-scale (e.g., the 3×3 world in Figure 5-1 represents some local region of the entire globe). However, in the later part of this chapter, the localized view will not be emphasized, because main results can apply to other problems of smaller scales.

There are two main aspects in this problem: 1) quantification of the information reward associated with a continuous (in both the temporal and spatial sense) measurement, and 2) path planning techniques that provide optimal (or good suboptimal)

solutions. Effective quantification of the information reward in the continuous-time domain will be first presented; then, a finite dimensional representation of a spatially continuous path will be provided. This chapter will formulate an optimal motion planning problem and also present a real-time motion planning law based on the information potential field.

5.2 Information by Continuous Measurement

5.2.1 Linear System Model

Consider the environmental dynamics of state variables $X_t \in \mathbb{R}^{nx}$ described by the following linear time-varying system[1]:

$$\dot{X}_t = A(t)X_t + W_t \tag{5.1}$$

where $W_t \in \mathbb{R}^n$ is a zero-mean Gaussian noise with $\mathbb{E}[W_t W_s'] = \Sigma_W \delta(t-s)$, $\Sigma_W \succeq 0$, which is independent of X_t. The prime sign $'$ denotes the transpose of a matrix. The initial condition of the state, X_0 is normally distributed as $X_0 \sim \mathcal{N}(\mu_0, P_0)$, $P_0 \succ 0$. This work considers a linear measurement model for $Z_t \in \mathbb{R}^m$ with additive Gaussian noise:

$$Z_t = C(t)X_t + N_t \tag{5.2}$$

where $N_t \in \mathbb{R}^m$ is zero-mean Gaussian with $\mathbb{E}[N_t N_s'] = \Sigma_N \delta(t-s)$, $\Sigma_N \succ 0$, which is independent of X_t and W_s, $\forall s$. A linear sensing model can be a good representation of observation of environmental variables distributed in field, such as temperature and pressure.

With this environmental system model, this work is interested in determining the impact of a measurement history in the past on the uncertainty reduction of some

[1]For notational convenience, the subscript represents the time argument in contrast to the location index as does in other chapters. Also, this chapter will use different notations for the verification time, the state dimension, and the size of verification region.

verification variables in the future. A measurement history up to time t is defined as

$$\mathcal{Z}_t = \{Z_\sigma : \sigma \in [0, t]\}. \tag{5.3}$$

The verification variables are a subset of the state variables that can be expressed as

$$V_t = M_V X_t \in \mathbb{R}^{n_V} \tag{5.4}$$

where $M_V \in \{0,1\}^{n_V \times n_X}$, $n_V \leq n_X$ with every row-sum of M_V being unity. Although this work is specifically interested in the case entries of M_V are zero or one, the results can be easily extended to a general $M_V \in \mathbb{R}^{n_V \times n_X}$.

Employing entropy as a metric of uncertainty, the uncertainty reduction of a random quantity by another random quantity is expressed as the mutual information between them. The *information reward by a measurement path*, in this work, is defined as follows:

$$\mathcal{J}_V(T, \tau) \triangleq \mathcal{I}(V_T; \mathcal{Z}_\tau), \quad 0 \leq \tau < T, \tag{5.5}$$

which is the mutual information between V_T and $\mathcal{Z}_\tau$. This represents the entropy reduction of the verification variables in the future time T by the measurement history up to time τ, and also the entropy reduction of $\mathcal{Z}_\tau$ by knowledge of V_T.

5.2.2 Filter Form

For linear Gaussian systems, there are known expressions for the mutual information between the state variables at a given time and the measurement history up to that time [20–22]. Therefore, one way to compute the information reward is to consider a *filtering* problem that estimates X_T based on the measurement history up to time T denoted as

$$\mathcal{Z}_T \triangleq \mathcal{Z}_\tau \cup \emptyset_{(\tau, T]} \tag{5.6}$$

where $\emptyset_{(\tau,T]}$ means that no measurement is taken during $(\tau, T]$. Then,

$$\mathcal{I}(X_T; \mathcal{Z}_T) = \mathcal{I}(X_T; \mathcal{Z}_\tau), \tag{5.7}$$

because no information is gathered by null measurement. This procedure of obtaining $\mathcal{I}(X_T; \mathcal{Z}_\tau)$ can be extended when our interest is in $\mathcal{I}(V_T; \mathcal{Z}_\tau)$, as outlined in the following proposition:

Proposition 3. *(Filter Form)* For the linear system described in this chapter, the information reward can be computed as

$$\mathcal{J}_V^F(T, \tau) \triangleq \mathcal{I}(V_T; \mathcal{Z}_\tau, \emptyset_{(\tau,T]}) \tag{5.8}$$

$$= \frac{1}{2} \log \det(M_V P_X(T) M_V') - \frac{1}{2} \log \det(M_V Q_X(T) M_V') \tag{5.9}$$

where the superscript 'F' stands for the filter form, and $P_X(T)$ and $Q_X(T)$ are obtained by integrating the following matrix differential equations:

$$\dot{P}_X(t) = A(t)P_X(t) + P_X(t)A'(t) + \Sigma_W \tag{5.10}$$

$$\dot{Q}_X(t) = A(t)Q_X(t) + Q_X(t)A'(t) + \Sigma_W - \mathbb{I}_{[0,\tau]}(t)Q_X(t)C(t)'\Sigma_N^{-1}C(t)Q_X(t) \tag{5.11}$$

with initial conditions $P_X(0) = Q_X(0) = P_0 \succ 0$, and $\mathbb{I}_{[0,\tau]}(t) : \mathbb{R}_+ \mapsto \{0, 1\}$ is the indicator function that is unity for $t \in [0, \tau]$ and zero elsewhere. The above equations are well-defined for finite T with $P_0 \succ 0$.

Proof. Note that

$$\mathcal{I}(V_T; \mathcal{Z}_\tau) = \mathcal{I}(V_T; \widehat{V}_T) \tag{5.12}$$

where $\widehat{V}_T = \mathbb{E}[V_T | \mathcal{Z}_\tau]$, because $\widehat{V}_T$ is the sufficient statistic that captures all information contained in $\mathcal{Z}_\tau$ about V_T [22]. Since the remaining randomness in V_T for a given $\widehat{V_T}$, is the estimation error $\widetilde{V}_T \triangleq V_T - \widehat{V}_T$, the mutual information

$$\mathcal{I}(V_T; \mathcal{Z}_\tau) = \mathcal{H}(V_T) - \mathcal{H}(\widetilde{V}_T). \tag{5.13}$$

For a linear system with Gaussian noise, both V_T and $\widetilde{V}_T$ are normally distributed with covariance of:

$$\mathbf{P}(V_T) = M_V P_X(T) M_V', \ \mathbf{P}(\widetilde{V}_T) = M_V Q_X(T) M_V', \tag{5.14}$$

when $P_X(T)$ and $Q_X(T)$ are obtained as (5.10) and (5.11). From (2.3), the entropy of the Gaussian random vector V_T is computed as

$$\mathcal{H}(V_T) = \frac{1}{2} \left[\log \det \ \mathbf{P}(V_T) + n_V \log(2\pi e) \right], \tag{5.15}$$

and likewise, the entropy of $\widetilde{V}_T$ can be computed; this finally gives the expression for $\mathcal{J}_V^F(T, \tau)$ in (5.9). □

The filter form provides a simple expression for the information reward for a continuous history of past measurement, which is described in terms of the solution of a Lyapunov equation and a Riccati equation. However, this form of computation involves the following issues, particularly, in case the forecast horizon is much longer than the planning horizon, i.e. $T \gg \tau$:

Remark 2. *(Issues in filter form)*

- As done in this thesis, a linear model is often used to represent the short-term (and localized) dynamic behavior of a nonlinear system. Since the filter form of the information reward relies on integration of linear dynamics for the time horizon $[0, T]$, it can be sensitive to linearization error if T is large.

- The computational complexity of numerical integration is linear in the integration time interval if the differential equation is integrated using a fixed-time step integration scheme. Even in case a variable time-step scheme is used, it is typical that integration of a same type of differential equation for longer time consumes more computational resource than that for shorter time. Thus, the computation time for computing $\mathcal{J}_V^F(T, \tau)$ increases with respect to T. This is not a problem when one is interested in computation of the information reward

for few measurement histories. However, if the goal is to design an optimal measurement path, and the computation of information rewards for many different candidate measurement histories is needed, long integration can cause computational inefficiency.

- The concurrent knowledge about the information gathered by the measurement taken thus far, $\mathcal{I}(V_T; \mathcal{Z}_\sigma)$ for some $\sigma < \tau$, can be useful for real-time adaptive decision making. However, in the filter form expression, this information requires significant amount of further computation, as the expression (5.9) computed at σ does not represent the information rewarded thus far; in other words,

$$\mathcal{I}(V_T; \mathcal{Z}_\sigma) \neq \frac{1}{2}\left[\log\det(M_V P_X(\sigma) M_V') - \log\det(M_V Q_X(\sigma) M_V')\right] = \mathcal{I}(V_\sigma; \mathcal{Z}_\sigma). \tag{5.16}$$

Instead, calculation of $\mathcal{I}(V_T; \mathcal{Z}_\sigma)$ requires additional integration of $Q_X(t)$ over $[\sigma, T]$:

$$\dot{Q}_X(t) = A(t)Q_X(t) + Q_X(t)A'(t) + \Sigma_W(t), \tag{5.17}$$

starting with initial condition $Q_X(\sigma)$, in order to propagate the influence of $\mathcal{Z}_\sigma$ until the verification time T. Further discussion on the on-the-fly information will be given in section 5.2.4.

5.2.3 Smoother Form

This section proposes a *smoother form* for quantification of the information reward, which resolves all the issues in the filter form listed in Remark 2. It can be shown that the differential equations for the smoother form are integrated for the planning time horizon, and that the information attained by the measurement thus far can be immediately computed without significant computation. This section first suggests an alternative expression of the mutual information that applies to any nonlinear non-Gaussian case:

Proposition 4. *(Information Identity)* If state dynamics satisfy the Markov property, i.e. the future state is conditionally independent of the past state given the present

state, and the measurement noise is independent of future process noise, then

$$\mathcal{I}(V_T; \mathcal{Z}_\tau) = \mathcal{I}(X_\tau; \mathcal{Z}_\tau) - \mathcal{I}(X_\tau; \mathcal{Z}_\tau | V_T). \tag{5.18}$$

In other words, $\mathcal{I}(V_T; \mathcal{Z}_\tau)$ can be interpreted as the difference between the information about X_τ contained in $\mathcal{Z}_\tau$, before and after V_T is revealed.

Proof. Utilizing the relationship between the mutual information and the conditional entropy, and commutativity of mutual information, the right-hand side of (5.18 can be manipulated as

$$\begin{aligned}
\mathcal{I}(X_\tau; \mathcal{Z}_\tau) - \mathcal{I}(X_\tau; \mathcal{Z}_\tau | V_T) &= \mathcal{I}(\mathcal{Z}_\tau; X_\tau) - \mathcal{I}(\mathcal{Z}_\tau; X_\tau | V_T) \\
&= \mathcal{H}(\mathcal{Z}_\tau) - \mathcal{H}(\mathcal{Z}_\tau | X_\tau) - \mathcal{H}(\mathcal{Z}_\tau | V_T) + \mathcal{H}(\mathcal{Z}_\tau | X_\tau, V_T) \\
&= \mathcal{I}(\mathcal{Z}_\tau; V_T) - \mathcal{I}(\mathcal{Z}_\tau; V_T | X_\tau) \\
&= \mathcal{I}(V_T; \mathcal{Z}_\tau) - \mathcal{I}(V_T; \mathcal{Z}_\tau | X_\tau). \qquad (5.19)
\end{aligned}$$

Notice that

$$\mathcal{I}(V_T; \mathcal{Z}_\tau | X_\tau) = 0, \tag{5.20}$$

because $\mathcal{Z}_\tau - X_\tau - V_T$ forms a Markov chain; i.e., $\mathcal{Z}_\tau$ and V_T are conditionally independent of each other for given X_τ. Thus, finally it follows

$$\mathcal{I}(V_T; \mathcal{Z}_\tau) = \mathcal{I}(X_\tau; \mathcal{Z}_\tau) - \mathcal{I}(X_\tau; \mathcal{Z}_\tau | V_T). \tag{5.21}$$

□

For linear systems with Gaussian noise described by (5.1) and (5.2), the mutual information between X_τ and $\mathcal{Z}_\tau$ can be easily computed by using a known expression in the literature [20–22]. The conditional mutual information $\mathcal{I}(X_\tau; \mathcal{Z}_\tau | V_T)$ can be quantified by posing a fixed-interval smoothing problem that incorporates the continuous measurement history $\mathcal{Z}_\tau$ and the discrete noise-free measurement of the verification variables at time T:

Proposition 5. *(Smoother Form)* If $P_{0|V} \triangleq \mathbf{P}(X_0|V_T) \succ 0$ is available, the information reward $\mathcal{I}(V_T; \mathcal{Z}_\tau)$ can be computed as

$$\mathcal{J}_V^S(T,\tau) \triangleq \mathcal{I}(X_\tau; \mathcal{Z}_\tau) - \mathcal{I}(X_\tau; \mathcal{Z}_\tau|V_T) \tag{5.22}$$

$$= \mathcal{J}_0(\tau) - \frac{1}{2}\log\det(I + Q_X(\tau)\Delta_S(\tau)) \tag{5.23}$$

where

$$\mathcal{J}_0(\tau) \triangleq \frac{1}{2}\log\det S_{X|V}(\tau) - \frac{1}{2}\log\det S_X(\tau) \tag{5.24}$$

$$\Delta_S(\tau) \triangleq S_{X|V}(\tau) - S_X(\tau), \tag{5.25}$$

and $S_X(\tau)$, $S_{X|V}(\tau)$, and $Q_X(\tau)$ are determined by the following matrix differential equations:

$$\dot{S}_X(t) = -S_X(t)A(t) - A(t)'S_X(t) - S_X(t)\Sigma_W(t)S_X(t) \tag{5.26}$$

$$\dot{S}_{X|V}(t) = -S_{X|V}(t)(A(t) + \Sigma_W(t)S_X(t)) - (A(t) + \Sigma_W(t)S_X(t))'S_{X|V}(t) + S_{X|V}(t)\Sigma_W(t)S_{X|V}(t) \tag{5.27}$$

$$\dot{Q}_X(t) = A(t)Q_X(t) + Q_X(t)A(t)' + \Sigma_W(t) - Q_X(t)C(t)'\Sigma_N^{-1}(t)C(t)Q_X(t) \tag{5.28}$$

with initial conditions $S_X(0) = P_0^{-1}$, $S_{X|V}(0) = P_{0|V}^{-1}$, and $Q_X(0) = P_0$. The superscript 'S' denotes the smoother form.

Proof. First, the unconditioned mutual information $\mathcal{I}(X_\tau; \mathcal{Z}_\tau)$ can be expressed as

$$\mathcal{I}(X_\tau; \mathcal{Z}_\tau) = \frac{1}{2}\log\det P_X(\tau) - \frac{1}{2}\log\det Q_X(\tau) \tag{5.29}$$

where $P_X(\tau)$ is the solution to the Lyapunov equation:

$$\dot{P}_X(t) = A(t)P_X(t) + P_X(t)A(t)' + \Sigma_W(t), \tag{5.30}$$

with $P_X(0) = P_0$, and $Q_X(t)$ is the solution to the Riccati equation:

$$\dot{Q}_X(t) = A(t)Q_X(t) + Q_X(t)A(t)' + \Sigma_W(t) - Q_X(t)C(t)'\Sigma_N^{-1}(t)C(t)Q_X(t). \quad (5.31)$$

Regarding the conditional mutual information term, note that

$$\mathcal{I}(X_\tau; \mathcal{Z}_\tau | V_T) = \frac{1}{2}(\log\det P_1 - \log\det P_2) \quad (5.32)$$

where P_1 is the covariance of $\widetilde{X}_1 \triangleq X_\tau - \mathbb{E}[X_\tau|V_T]$ and P_2 is the covariance of $\widetilde{X}_2 \triangleq X_\tau - \mathbb{E}[X_\tau|V_T, \mathcal{Z}_\tau]$. Namely, P_2 is the error covariance of the fixed-interval smoothing with the past measurement $\mathcal{Z}_\tau$ and the future measurement V_T; Wall et al. [44] suggested the expression for the error covariance for the fixed-interval smoothing

$$Q_{X|V}^{-1}(t) \triangleq \mathbf{P}(X_t|V_T, \mathcal{Z}_t) = Q_X^{-1}(t) + P_{X|V}^{-1}(t) - P_X^{-1}(t) \quad (5.33)$$

where $P_{X|V}(t)$ is the estimation error covariance accounting for the future measurement plus a priori information. $P_{X|V}(t)$ is computed as a solution of a Riccati-like equation that is integrated backwards. Since, in this work's setting, there is no future measurement except a discrete measurement at T, $P_{X|V}(\tau)$ is same as P_1 and can be computed by the following Lyapunov-like equation integrated backwards:

$$\dot{P}_{X|V}(t) = (A(t) + \Sigma_W(t)P_X^{-1}(t))P_{X|V}(t) + P_{X|V}(t)(A(t) + \Sigma_W(t)P_X^{-1}(t))' - \Sigma_W(t) \quad (5.34)$$

with terminal condition $P_{X|V}(T) = \mathbf{P}(X_T|V_T)$ to time τ. Note that $\mathbf{P}(X_T|V_T)$ is all zero except the part corresponding to $\mathbf{P}(X_T \setminus V_T|V_T)$. It should be noticed that this equation need not to be integrated backwards, because every quantity on the right-hand side is available at time t by the past knowledge. It can be integrated forward with initial condition $P_{X|V}(0) = P_{0|V}$, which is assumed to be available. Thus, P_2 can be computed by the forward integration of three matrix differential equations: a Lyapunov equation for P_X, a Riccati equation for Q_X, and a Lyapunov-like equation for $P_{X|V}$, so can the information reward $\mathcal{J}_V^S(T, \tau)$.

In addition, equations for P_X and $P_{X|V}$ can preferably be written in terms of the information matrices, $S_X \equiv P_X^{-1}$ and $S_{X|V} \equiv P_{X|V}^{-1}$; this removes the need for performing matrix inversion in (5.34). Using

$$\frac{d}{dt}\left(M_1^{-1}\right) = -M_1^{-1}\left(\frac{d}{dt}M_1\right)M_1^{-1} \tag{5.35}$$

for a symmetric positive definite M_1, equations for S_X and $S_{X|V}$ are obtained as

$$\dot{S}_X(t) = -S_X(t)A(t) - A(t)'S_X(t) - S_X(t)\Sigma_W(t)S_X(t) \tag{5.36}$$

$$\begin{aligned}\dot{S}_{X|V}(t) = &-S_{X|V}(t)(A(t) + \Sigma_W(t)S_X(t)) - (A(t) + \Sigma_W(t)S_X(t))'S_{X|V}(t) \\ &+ S_{X|V}(t)\Sigma_W(t)S_{X|V}(t).\end{aligned} \tag{5.37}$$

Finally, using the properties of the determinant function: $\log\det M_1^{-1} = -\log\det M_1$ and $\det(M_1M_2) = \det M_1 \det M_2$ for square matrices M_1 and M_2,

$$\begin{aligned}\mathcal{J}_V^S(T,\tau) &= \mathcal{I}(X_\tau;\mathcal{Z}_\tau) - \mathcal{I}(X_\tau;\mathcal{Z}_\tau|V_T) \\ &= \frac{1}{2}\left[\log\det P_X(\tau) - \frac{1}{2}\log\det Q_X(\tau)\right] - \frac{1}{2}\left[\log\det P_{X|V}(\tau) - \log\det Q_{X|V}(\tau)\right] \\ &= \frac{1}{2}\Big[- \log\det S_X(\tau) - \log\det Q_X(\tau) + \log\det S_{X|V}(\tau) \\ &\qquad - \log\det\left(Q_X^{-1}(\tau) + S_{X|V}(t) - S_X(\tau)\right)\Big] \\ &= \frac{1}{2}\left[\log\det S_{X|V}(\tau) - \log\det S_X(\tau)\right] \\ &\qquad - \frac{1}{2}\left[\log\det\left(Q_X(\tau)[Q_X^{-1}(\tau) + S_{X|V}(t) - S_X(\tau)]\right)\right] \\ &= \mathcal{J}_0(\tau) - \frac{1}{2}\log\det(I + Q_X(\tau)\Delta_S(\tau)),\end{aligned} \tag{5.38}$$

with $\mathcal{J}_0(\tau) \triangleq \frac{1}{2}\log\det S_{X|V}(\tau) - \frac{1}{2}\log\det S_X(\tau)$ and $\Delta_S(\tau) \triangleq S_{X|V}(\tau) - S_X(\tau)$. □

Remark 3. *(Computation of Conditional Initial Covariance)* For the linear setting given defined by (5.1) and (5.2), $P_{0|V}$ can be computed by the covariance update formula:

$$P_{0|V} = P_0 - P_0\Phi'_{(T,0)}M'_V\left[M_VP_X(T)M'_V\right]^{-1}M_V\Phi_{(T,0)}P_0. \tag{5.39}$$

where $\Phi_{(T,0)}$ is the state transition matrix from time 0 to T. Since the final state X_T can be represented by the initial state and the process noise history as

$$X_T = \Phi_{(T,0)} X_0 + \int_0^T \Phi_{(T,\sigma)} W_\sigma d\sigma, \tag{5.40}$$

the covariance between the initial state and the final state becomes

$$P_{X_0,X_T} \triangleq \mathbf{P}(X_0, X_T) = P_0 \Phi'_{(T,0)}, \tag{5.41}$$

because W_t, $t \in [0,T]$ is independent of X_0.[2] Therefore, $\mathbf{P}(X_0, V_T)$ becomes $M_V \Phi_{(T,0)} P_0$, which leads to the conditional covariance expression in (5.39). If the system is time-invariant, the state transition matrix $\Phi_{(T,0)} = \exp(AT)$, which can be calculated without integrating any differential equations. The final covariance $P_X(T)$ is computed by integrating the Lyapunov equation

$$\dot{P}(t) = A(t)'P(t) + P(t)A(t) + \Sigma_W \tag{5.42}$$

starting from P_0. Note that the inverse on the right-hand side exists for finite T with $P_0 \succ 0$.

For a time-varying case, a fixed-point smoothing using state augmentation can be applied for finding $P_{0|V}$. By augmenting X_t with X_0, the dynamics of the augmented state is expressed as

$$\begin{bmatrix} \dot{X}_t \\ \dot{X}_0 \end{bmatrix} = \begin{bmatrix} A(t) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} X_t \\ X_0 \end{bmatrix} + \begin{bmatrix} W_t \\ 0 \end{bmatrix}. \tag{5.43}$$

The covariance propagation for this augmented state is represented as the following Lyapunov equation:

$$\begin{bmatrix} \dot{P}_X & \dot{P}_{X_t,X_0} \\ \dot{P}_{X_0,X_t} & \dot{P}_0 \end{bmatrix} = \begin{bmatrix} A(t) & 0 \\ 0 & 0 \end{bmatrix}' \begin{bmatrix} P_X & P_{X_t,X_0} \\ P_{X_0,X_t} & P_0 \end{bmatrix} + \begin{bmatrix} P_X & P_{X_t,X_0} \\ P_{X_0,X_t} & P_0 \end{bmatrix} \begin{bmatrix} A(t) & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} \Sigma_W & 0 \\ 0 & 0 \end{bmatrix}. \tag{5.44}$$

[2] As defined in Chapter 2, the notation $\mathbf{P}(Y_1, Y_2) \triangleq \mathbb{E}[(Y_1 - \mathbb{E}[Y_1])(Y_2 - \mathbb{E}[Y_2])']$ for two random vectors Y_1 and Y_2.

Integration of this equation with initial condition $P_0 \otimes I_2$ over $[0, T]$ provides $P_X(T)$ and P_{X_0,X_T}.

In case the linear system is used to approximate a short-term behavior of a nonlinear system whose long-term behavior is tracked by some nonlinear estimation scheme such as an ensemble forecast system, $P_{0|V}$ can be provided by this nonlinear estimator. The ensemble augmentation technique presented in section 3.1.1 can be used for this purpose. Given initial ensemble $\mathbf{X}_0 \in \mathbb{R}^{n_X \times L_E}$, the forecast ensemble at T is computed by

$$\mathbf{X}_T = \int_0^T \dot{\mathbf{X}}_t dt, \tag{5.45}$$

and a measurement update process (as in (2.17)) for the augmented ensemble

$$\mathbf{X}_{aug} = \begin{bmatrix} \mathbf{X}_0 \\ M_V \mathbf{X}_T \end{bmatrix} \in \mathbb{R}^{(n_X+n_V)\times L_E} \tag{5.46}$$

with noise-free measurement with observation matrix

$$\mathbf{H}_{aug} = \begin{bmatrix} 0_{n_V \times n_X} I_{n_V \times n_V} \end{bmatrix} \tag{5.47}$$

results in the conditional perturbation ensemble $\widetilde{\mathbf{X}}_{0|V}$. Then, the conditional initial covariance $P_{0|V}$ is calculated as

$$P_{0|V} \approx \frac{1}{L_E - 1} \widetilde{\mathbf{X}}_{0|V} \widetilde{\mathbf{X}}'_{0|V}. \tag{5.48}$$

□

Remark 4. *(Resolution of Issues in Filter Form)*

- Note that given $P_{0|V}$, every differential equation for the smoother form is integrated forward up to time τ. The only part for which a long integration is required is the calculation of the conditional initial covariance $P_{0|V}$, which is performed only once before incorporating the measurement history using techniques given in Remark 3.

- With regard to the issue of sensitivity to linearization error, if $P_{0|V}$ is provided from a nonlinear estimation scheme, calculation of $P_{0|V}$ is not affected by linearization error. Therefore, the linearization error propagates only for the short time interval $[0, \tau]$, which provides better robustness to linearization error than the filter form. Even when there is no such nonlinear estimator, the smoother form allows for pinpointing the possible effect of the limited model validity to the just the $P_{0|V}$ term, which enables a separate analysis of improving the quality of this term.

- Regarding computational complexity, only one quantity, $Q_X(\tau)$ is dependent on the measurement history; $\mathcal{J}_0(\tau)$ and $\Delta_S(\tau)$ can be pre-computed in advance of computing $Q_X(\tau)$. Thus, if one needs to evaluate the information rewards for different measurement histories, then one can simply integrate the Riccati equation in (5.28) over $[0, \tau]$ to get $Q_X(\tau)$ for various measurement options, while all other information has been computed without regard to the measurement choice. Therefore, compared to the filter form, the computation cost of evaluating rewards for different measurement options is reduced by the factor of T/τ, because the same matrix differential equation (i.e., Riccati equation) is integrated for a shorter time interval $[0, \tau]$.

- For the smoother form, the information gain accomplished by a partial measurement history $\mathcal{Z}_\sigma$, $\sigma < \tau$ is straightforwardly computed as $\mathcal{J}_V^S(T, \sigma)$, the same expression as (5.23) with the matrix values evaluated for time σ. Since all the matrix differential equations are integrated forward, $\mathcal{J}_V^S(T, \sigma)$ is available in real time without further computation. This on-the-fly information will be discussed in detail in section 5.2.4 with the notion of mutual information rate.

Remark 5. *(Interpretation as Weighting)* Conventional path planning problems [33] focus on reducing the uncertainty in the state estimate at the end of the planning window: i.e. $\min \log\det\, Q_X(\tau)$. Recalling that the forecast problem written in the smoother form considers $\min \log\det(I + Q_X(\tau)\Delta_S(\tau))$, the smoother form objective function can be regarded as some weighted version of the conventional objective

function. Depending on $\Delta_S(\tau)$, the solutions of the two problems can be very different; however, a planning algorithm for the conventional decision can be *easily extended* to the forecast problem, since the smoother enables projection of the decision space from a long forecast horizon $[0, T]$ on to a short planning horizon $[0, \tau]$. □

Remark 6. *(Information Form Riccati Equation)* The smoother form can also be computed using the information form Riccati equation instead of the covariance form Riccati equation in (5.28); in other words,

$$\mathcal{J}_V^S(T, \tau) = \mathcal{J}_0(\tau) - \frac{1}{2}\left[-\log\det J_X(\tau) + \log\det(J_X(\tau) + \Delta_S(\tau))\right] \tag{5.49}$$

where $J_X(\tau) \triangleq Q_X(\tau)^{-1}$ is determined from forward integration of

$$\dot{J}_X(t) = -A(t)J_X(t) - J_X(t)A(t)' - J_X(t)\Sigma_W J_X(t) + C'(t)\Sigma_N^{-1}C(t) \tag{5.50}$$

with initial condition $J_X(0) = P_0^{-1}$. The information form can be preferred in the cases where a) there is no process noise, i.e., $\Sigma_W = 0$, and b) the observation matrix C represents the measurements by multiple sensor platform. If $\Sigma_W = 0$, the Riccati equation in (5.50) becomes a Lyapunov equation that is easier to integrate. When C is expressed as

$$C = \begin{bmatrix} C_1' & C_2' & \dots & C_n' \end{bmatrix}' \tag{5.51}$$

with C_i representing the observation matrix for i-th sensor platform, the last term in the right-hand side of (5.50) becomes

$$C'\Sigma_N^{-1}C = \begin{bmatrix} C_1' & \dots & C_n' \end{bmatrix} \Sigma_N^{-1} \begin{bmatrix} C_1 \\ \vdots \\ C_n \end{bmatrix} = \sum_{i=1}^{n} C_i'\Sigma_N^{-1}C_i. \tag{5.52}$$

Compared to the covariance form Riccati equation in (5.28), the information form is preferable in that the contribution of individual sensor platform is easily quantifiable. In this work, main results will be given in the covariance form as it provides some important insights such as weighting matrix discussed in Remark 5. □

5.2.4 On-the-fly Information and Mutual Information Rate

This section discusses the on-the-fly information available in the process of computing the filter form and the smoother form mutual information, and identifies important features of the smoother form in terms of information supply and dissipation. Moreover, this analysis facilitates building an information potential field that can be used to visualize the spatial distribution of information quantity and is used to develop a real-time steering law for a mobile sensor in section 5.4.2.

Filter-Form On-the-fly Information (FOI)

Information Since the Lyapunov equation in (5.10) and the Riccati equation in (5.11) are integrated forward from time 0, the available matrix values at time $t < \tau$ are $P_X(t)$ and $Q_X(t)$. With these, the mutual information between the current state variables and the measurement thus far can be evaluated as

$$\mathcal{I}(X_t; \mathcal{Z}_t) = \frac{1}{2} \log\det P_X(t) - \frac{1}{2} \log\det Q_X(t). \tag{5.53}$$

Information Rate The expression of the time derivative of FOI was first presented in [21]; its interpretation as information supply and dissipation was provided in [22]. The rate of FOI can be derived as

$$\begin{aligned}
\frac{d}{dt}\mathcal{I}(X_t; \mathcal{Z}_t) &= \frac{d}{dt}\left[\frac{1}{2}\log\det\ P_X(t) - \frac{1}{2}\log\det\ Q_X(t)\right] \\
&= \frac{1}{2}\mathtt{trace}\left\{P_X^{-1}\dot{P}_X - Q_X^{-1}\dot{Q}_X\right\} \\
&= \underbrace{\frac{1}{2}\mathtt{trace}\left\{\Sigma_N^{-1} C Q_X C'\right\}}_{\text{Info Supply}} - \underbrace{\frac{1}{2}\mathtt{trace}\left\{\Sigma_W(Q_X^{-1} - P_X^{-1})\right\}}_{\text{Info Dissipation}}
\end{aligned} \tag{5.54}$$

where every matrix is evaluated at t. The first term depends on the measurement and represents the rate of information supply, while the second term depends on the process noise and represents the rate of information dissipation [22]. It can be proven

that the signs of the supply and the dissipation term are non-negative:

$$\mathtt{trace}\Big\{\Sigma_N^{-1}CQ_XC'\Big\} \geq 0, \qquad \mathtt{trace}\Big\{\Sigma_W(Q_X^{-1} - P_X^{-1})\Big\} \geq 0,$$

since $CQ_XC' \succeq 0$, $Q_X^{-1} - P_X^{-1} \succ 0$, and trace of the product of two symmetric positive definite matrices is non-negative [45]. Thus, measurement tends to increase FOI while the process noise tends to decrease it; FOI can be decreasing over time if the information dissipation dominates the information supply.

Note that previous work [33] derived a quantity identical to the information supply term in developing some information potential field. The author of [33] considered the entropy of the current state, $\mathcal{H}(X_t)$, which is $-\frac{1}{2}\log\det J_X(t) + \frac{n_X}{2}\log(2\pi e)$, where $J_X(t) \triangleq Q_X^{-1}(t)$ is the Fisher information matrix at time t. Then,

$$\begin{aligned}
-\frac{d}{dt}\mathcal{H}(X_t|\mathcal{Z}_t) &= \frac{1}{2}\frac{d}{dt}\log\det J_X(t) && (5.55)\\
&= \frac{1}{2}\mathtt{trace}\Big\{J_X^{-1}\Big(-J_XA - A'J_X - J_X\Sigma_WJ_X + C'\Sigma_N^{-1}C\Big)\Big\}\\
&= \frac{1}{2}\mathtt{trace}\Big\{J_X^{-1}C'\Sigma_N^{-1}C\Big\} - \frac{1}{2}\mathtt{trace}\Big\{A + J_X^{-1}A'J_X + \Sigma_WJ_X)\Big\}\\
&= \frac{1}{2}\mathtt{trace}\Big\{\Sigma_N^{-1}CQ_XC'\Big\} + \alpha(t) && (5.56)
\end{aligned}$$

where $\alpha(t)$ consists of terms that are not dependent on the observation matrix C. Note that $\alpha(t) \neq \frac{1}{2}\mathtt{trace}(Q_X^{-1} - P_X^{-1})$; thus, the procedure given in [33] does not provide the expression of the rate of FOI.

Note that the dissipation term in (5.54) is zero if $P_X(t) = Q_X(t)$, which corresponds to the case where no measurement is ever taken up to time t. Thus, the information supply term represents the rate of information accumulation assuming no measurement has been taken up to the current time. Consider the case no measurement is taken over $[0,\tau]$, i.e. $\mathcal{Z}_\tau = \emptyset$; then, at some $t < \tau$, the rate of FOI in (5.54) equals to zero regardless of the process noise. However, in this case, the remainder term $\alpha(t)$ in (5.56) derived from the information form Riccati equation is positive in case $\Sigma_W \succ 0$, representing the increase of the entropy due to the process noise.

Projected Filter-Form On-the-fly Information (PFOI)

Information Similar to FOI, the mutual information between the current verification variables and the measurement thus far can also be computed on the fly, while computing the filter form mutual information:

$$\mathcal{I}(V_t; \mathcal{Z}_t) = \frac{1}{2} \log\det P_V(t) - \frac{1}{2} \log\det Q_V(t). \tag{5.57}$$

where $P_V(\cdot) \triangleq M_V P_X(\cdot) M_V'$ and $Q_V(\cdot) \triangleq M_V Q_X(\cdot) M_V'$.

Information Rate The time derivation of PFOI can also be expressed in terms of $P_X(t)$ and $Q_X(t)$ as follows.

$$\begin{aligned} \frac{d}{dt}\mathcal{I}(V_t; \mathcal{Z}_t) &= \frac{1}{2}\mathtt{trace}\Big\{ P_V^{-1}\dot{P}_V - Q_V^{-1}\dot{Q}_V \Big\} \\ &= \underbrace{\frac{1}{2}\mathtt{trace}\Big\{ \Sigma_N^{-1} C Q_X M_V' Q_V^{-1} M_V Q_X C' \Big\}}_{\texttt{Direct Supply}} + \beta(t), \end{aligned} \tag{5.58}$$

where $\beta(t)$ represents all the remaining terms that do not depend on the observation matrix C. The first term, underbraced as "Direct Supply" represents the immediate influence the measurement on the current verification variables; the remaining term $\beta(t)$ captures all the correlated effect due to coupling in dynamics on the information supply/dissipation. The sign of $\beta(t)$ is *indefinite*, while the direct supply term is non-negative as $C Q_X M_V' Q_V^{-1} M_V Q_X C' \succeq 0$. Although $\beta(t)$ is indefinite in general, it is zero if $P_X(t) = Q_X(t)$ and no measurement is taken at t. Thus, in case no measurement is taken over $[0, \tau]$, the value of PFOI is always zero in the time interval $[0, \tau]$ regardless of the process noise.

5.2.5 Smoother-Form On-the-fly Information for Forecasting (SOIF)

Information The smoother form can quantify, on the fly, the information accumulated by the measurement thus far. In the smoother form framework, the mutual

information between the future verification variables V_T and the measurement up to the current time t can be calculated as

$$\begin{aligned}\mathcal{I}(V_T;\mathcal{Z}_t) &= \mathcal{I}(X_t;\mathcal{Z}_t) - \mathcal{I}(X_t;\mathcal{Z}_t|V_T) \\ &= \mathcal{J}_0(t) - \frac{1}{2}\log\det(I + Q(t)\Delta_S(t)).\end{aligned} \tag{5.59}$$

The values of matrices $\mathcal{J}_0(t)$, $Q(t)$, and $\Delta_S(t)$ are calculated in the process process of the forward integration (5.26)–(5.28).

Information Rate The temporal derivative of the smoother form mutual information can be written as follows.

Proposition 6. *(Smoother-form Information Rate)* The temporal derivative of the smoother form information reward at t, $\mathcal{J}_V^S(T,t) \triangleq \mathcal{J}_0(t) - \frac{1}{2}\log\det(I + Q_X(t)\Delta_S(t))$ is written as:

$$\frac{d}{dt}\mathcal{J}_V^S(T,t) = \frac{1}{2}\mathtt{trace}\left\{\Sigma_N^{-1}C(t)\Pi(t)C(t)'\right\} \tag{5.60}$$

where

$$\Pi(t) \triangleq Q_X(t)(S_{X|V}(t) - S_X(t))[I + Q_X(t)(S_{X|V}(t) - S_X(t))]^{-1}Q_X(t). \tag{5.61}$$

Proof. For a symmetric positive definite matrix, M_1, the time derivative of its log det value is expressed as $\frac{d}{dt}(\log\det M_1) = \mathtt{trace}(M_1^{-1}\frac{d}{dt}M_1)$. Using this expression, the rate of change of $\mathcal{J}_V^S(T,t)$ can be written as:

$$\begin{aligned}\frac{d}{dt}\mathcal{J}_V^S(T,t) &= \frac{d}{dt}\left[\mathcal{J}_0(t) - \frac{1}{2}\log\det(I + Q_X(t)\Delta_S(t))\right] \\ &= \frac{d}{dt}\left[\frac{1}{2}\left(\log\det S_{X|V}(t) - \log\det S_X(t)\right) - \frac{1}{2}\log\det(I + Q_X(t)\Delta_S(t))\right] \\ &= \frac{1}{2}\mathtt{trace}\left\{S_{X|V}^{-1}\dot{S}_{X|V} - S_X^{-1}\dot{S}_X\right\} \\ &\quad - \frac{1}{2}\mathtt{trace}\left\{[I + Q_X(S_{X|V} - S_X)]^{-1}\left[\dot{Q}_X(S_{X|V} - S_X) + Q_X(\dot{S}_{X|V} - \dot{S}_X)\right]\right\}\end{aligned}$$

Consider the first term:

$$
\begin{aligned}
&\texttt{trace}\Big\{S_{X|V}^{-1}\dot{S}_{X|V} - S_X^{-1}\dot{S}_X\Big\} \\
&\quad = \texttt{trace}\Big\{S_{X|V}^{-1}(-S_{X|V}(A+\Sigma_W S_X) - (A+\Sigma_W S_X)'S_{X|V} + S_{X|V}\Sigma_W S_{X|V}) \\
&\qquad\qquad - S_X^{-1}(-S_X A - A'S_X - S_X\Sigma_W S_X)\Big\} \\
&\quad = \texttt{trace}\Big\{ - A - \Sigma_W S_X - A'S_{X|V}S_{X|V}^{-1} - S_X\Sigma_W S_{X|V}S_{X|V}^{-1} + \Sigma_W S_{X|V} \\
&\qquad\qquad + A + A'S_X S_X^{-1} + \Sigma_W S_X\Big\} \\
&\quad = \texttt{trace}\Big\{\Sigma_W(S_{X|V} - S_X)\Big\},
\end{aligned}
\tag{5.62}
$$

where the transition from the second to third lines uses the property of trace function: $\texttt{trace}(ABC) = \texttt{trace}(BCA) = \texttt{trace}(CAB)$. The second term can be expanded as:

$$
\begin{aligned}
&\texttt{trace}\Big\{[I+Q_X\Delta_S]^{-1}\big[\dot{Q}_X\Delta_S + Q_X\dot{\Delta}_S\big]\Big\} \\
&= \texttt{trace}\Big\{(I+Q_X\Delta_S)^{-1}\times\big[(AQ_X + Q_XA' + \Sigma_W \\
&\qquad\qquad - Q_XC'\Sigma_N^{-1}CQ_X)\Delta_S + Q_X(-\Delta_S A - A'\Delta_S A + \Delta_S\Sigma_W\Delta_S)\big]\Big\} \\
&= \texttt{trace}\Big\{(I+Q_X\Delta_S)^{-1}AQ_X\Delta_S + (I+Q_X\Delta_S)^{-1}Q_XA'\Delta_S + (I+Q_X\Delta_S)^{-1}\Sigma_W\Delta_S \\
&\qquad\qquad - (I+Q_X\Delta_S)^{-1}Q_XC'\Sigma_N^{-1}CQ_X\Delta_S - (I+Q_X\Delta_S)^{-1}Q_X\Delta_S A \\
&\qquad\qquad - (I+Q_X\Delta_S)^{-1}Q_XA'\Delta_S + (I+Q_X\Delta_S)^{-1}\Delta_S\Sigma_W\Delta_S\Big\} \\
&= \texttt{trace}\Big\{(I+Q_X\Delta_S)^{-1}AQ_X\Delta_S - (I+Q_X\Delta_S)^{-1}Q_X\Delta_S A \\
&\qquad\qquad + (I+Q_X\Delta_S)^{-1}(I+Q_X\Delta_S)\Sigma_W\Delta_S - (I+Q_X\Delta_S)^{-1}Q_XC'\Sigma_N^{-1}CQ_X\Delta_S\Big\} \\
&= \texttt{trace}\Big\{Q_X\left[\Delta_S(Q_X^{-1}+\Delta_S)^{-1}Q_X^{-1} - Q_X^{-1}(Q_X^{-1}+\Delta_S)^{-1}\Delta_S\right]A + \Sigma_W\Delta_S \\
&\qquad\qquad - (I+Q_X\Delta_S)^{-1}Q_X\Delta_S AQ_XC'\Sigma_N^{-1}CQ_X\Delta_S\Big\} \\
&= \texttt{trace}\Big\{\Sigma_W\Delta_S - (I+Q_X\Delta_S)^{-1}Q_XC'\Sigma_N^{-1}CQ_X\Delta_S\Big\},
\end{aligned}
$$

where the expression of $\dot{\Delta}_S$ can be derived from $\dot{S}_{X|V} - \dot{S}_X$. Therefore, $\frac{d}{dt}\mathcal{J}_V^S(T,t)$

can be written as

$$\begin{aligned}\frac{d}{dt}\mathcal{J}_V^S(T,t) &= \frac{1}{2}\mathtt{trace}\left\{\Sigma_W\Delta_S - \Sigma_W\Delta_S + (I+Q_X\Delta_S)^{-1}Q_XC'\Sigma_N^{-1}CQ_X\Delta_S\right\}\\ &= \frac{1}{2}\mathtt{trace}\left\{\Sigma_N^{-1}CQ_X(S_{X|V}-S_X)(I+Q_X(S_{X|V}-S_X))^{-1}Q_XC'\right\}\\ &= \frac{1}{2}\mathtt{trace}\left\{\Pi C'\Sigma_N^{-1}C\right\} \qquad (5.63)\\ &= \frac{1}{2}\mathtt{trace}\left\{\Sigma_N^{-1}C(t)\Pi C'\right\} \qquad (5.64)\end{aligned}$$

with $\Pi = Q_X(S_{X|V}-S_X)(I+Q_X(S_{X|V}-S_X))^{-1}Q_X$. □

Proposition 7. *(Non-Negativity of Information Rate)* The rate of the smoother-form on-the-fly information for forecasting (SOIF) is non-negative:

$$\frac{d}{dt}\mathcal{I}(V_T;\mathcal{Z}_t) = \underbrace{\frac{1}{2}\mathtt{trace}\left\{\Sigma_N^{-1}C(t)\Pi(t)C(t)'\right\}}_{\texttt{Info Supply}} \geq 0 \qquad (5.65)$$

Proof. One way to prove this is to utilize some properties of symmetric positive definite matrices. By the matrix inversion lemma [46], the matrix $\Pi \triangleq Q_X\Delta_S[I+Q_X\Delta_S]^{-1}Q_X$ is symmetric:

$$\Pi = Q_X\Delta_SQ_X - Q_X\Delta_S(Q_X^{-1}+\Delta_S)^{-1}\Delta_SQ_X = \Pi'$$

The Wigner's theorem [47] states that a product of multiple symmetric positive definite matrices is positive definite if the product is symmetric; thus, $\Pi \succeq 0$. This leads to $C\Pi C' \succeq 0$, and finally $\mathtt{trace}\{\Sigma_N^{-1}C\Pi C'\} \geq 0$, because the trace of two positive definite matrices is non-negative [45].

Another way to prove (5.65) is to use the principle of "information never hurts [35]." By definition of the time derivative,

$$\frac{d}{dt}\mathcal{I}(V_T;\mathcal{Z}_t) = \lim_{\epsilon\to 0}\left(\mathcal{I}(V_T;\mathcal{Z}_{t+\epsilon}) - \mathcal{I}(V_T;\mathcal{Z}_{t+\epsilon})\right)/\epsilon \qquad (5.66)$$

For a positive ϵ, $\mathcal{Z}_{t+\epsilon} \supset \mathcal{Z}_t$; therefore, $\mathcal{I}(V_T;\mathcal{Z}_{t+\epsilon}) \geq \mathcal{I}(V_T;\mathcal{Z}_t)$, because the additional information by measurement over $(t,t+\epsilon]$ does not increase the entropy of V_T. Like-

wise, $\mathcal{I}(V_T; \mathcal{Z}_{t+\epsilon}) \leq \mathcal{I}(V_T; \mathcal{Z}_t)$ for a negative ϵ. For both cases, the right-hand side of (5.66) is non-negative. This second way of proof can apply to nonlinear non-Gaussian cases as well. □

Since the influence of the future process noise has already been captured in $S_{X|V}$, the mutual information rate for the smoother form is non-negative regardless of the process noise as proven in Proposition 7. If one stops taking measurement at time t, the information reward stays constant. Thus, the mutual information rate for the smoother form can extract out the *pure impact of sensing* on the entropy reduction of the verification variables, while the rates for the filter forms depend on the process noise.

Remark 7. *(Information Rate for Multiple Sensors)* Consider the case where there are multiple sensor platforms, and the observation matrix of i-th sensor is C_i, constituting the overall observation matrix of

$$C = \begin{bmatrix} C_1' & \dots & C_n' \end{bmatrix}'. \tag{5.67}$$

Then, the smoother-form mutual information rate in Proposition 6 can be preferably written as

$$\begin{aligned}
\frac{d}{dt}\mathcal{J}_V^S(T,t) &= \frac{1}{2}\mathtt{trace}\left\{\Sigma_N^{-1}C(\mathbf{x},\mathbf{y})\Pi C(\mathbf{x},\mathbf{y})'\right\} \\
&= \frac{1}{2}\mathtt{trace}\left\{\Pi \times C(\mathbf{x},\mathbf{y})'\Sigma_N^{-1}C(\mathbf{x},\mathbf{y})\right\} \\
&= \frac{1}{2}\mathtt{trace}\left\{\Pi \times \sum_{i=1}^{n} C_i(x_i,y_i)'\Sigma_{N_i}^{-1}C_i(x_i,y_i)\right\} \\
&= \frac{1}{2}\mathtt{trace}\left\{\Pi\sum_{i=1}^{n} C_i(x_i,y_i)'\Sigma_{N_i}^{-1}C_i(x_i,y_i)\right\} \\
&= \frac{1}{2}\mathtt{trace}\left\{\sum_{i=1}^{n}\Pi C_i(x_i,y_i)'\Sigma_{N_i}^{-1}C_i(x_i,y_i)\right\} \\
&= \sum_{i=1}^{n}\frac{1}{2}\mathtt{trace}\left\{\Sigma_{N_i}^{-1}C_i(x_i,y_i)\Pi(t)C_i(x_i,y_i)'\right\}
\end{aligned} \tag{5.68}$$

where $(\mathbf{x},\mathbf{y})$ represents the positions of all the sensor platforms, and Σ_{N_i} is the (i,i)-th entry of Σ_N. In other words, the total rate of change of mutual information is the

sum of rate of change of mutual information of individual sensor platforms. □

Remark 8. It is noted that when $Q_X(t) = S_X^{-1}(t)$,

$$\Pi(t) = S_X^{-1}(t) - S_{X|V}^{-1}(t). \tag{5.69}$$

This specifically means, at initial time, $\Pi(0) = P_0 - P_{0|V}$, which leads to

$$\frac{d}{dt}\mathcal{I}(V_T; \mathcal{Z}_t)\Big|_{t=0} = \frac{1}{2}\mathtt{trace}\Big\{\Sigma_N^{-1}C(P_0 - P_{0|V})C'\Big\}. \tag{5.70}$$

This should be contrasted to the filter-form information rate, which becomes

$$\frac{d}{dt}\mathcal{I}(X_t; \mathcal{Z}_t)\Big|_{t=0} = \frac{1}{2}\mathtt{trace}\{\Sigma_N^{-1}CP_0C'\} \tag{5.71}$$

at the initial time. □

Example 1. Figure 5-2 compares the time histories of three on-the-fly quantities: the smoother-form on-the-fly information $\mathcal{J}_V^S(T,t) = \mathcal{I}(V_T; \mathcal{Z}_t)$, the filter-form on-the-fly information $\mathcal{I}(X_t; \mathcal{Z}_t)$, and the projected filter-form on-the-fly information $\mathcal{I}(V_t; \mathcal{Z}_t)$. In this example, the following system matrices are used with $\tau = 2$ and $T = 5$:

$$A = \begin{bmatrix} 0.1 & 1 \\ -1 & -0.5 \end{bmatrix}, \qquad \Sigma_W = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}, \qquad P_0 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$
$$C = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}, \qquad \Sigma_N = 0.01, \qquad M_V = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

First, notice that $\mathcal{J}_V^S(T,t)$ at $t = \tau$ is the same as $\mathcal{I}(V_t; \mathcal{Z}_t)$ at $t = T$, as $\mathcal{I}(V_T; \mathcal{Z}_\tau) = \mathcal{I}(V_T; \mathcal{Z}_T)$ with null measurement during $(\tau, T]$; this agreement in the plots numerically confirms the equivalence of the filter form and the smoother form.

For $\mathcal{J}_V^S(T,t)$, it is found that information increases in the presence of measurement (before τ) and stays constant in the absence of measurement (after τ). In the history of $\mathcal{I}(X_t; \mathcal{Z}_t)$, the information supply over $[0, \tau]$ increases the information while the information dissipation over $(\tau, T]$ decreases the information. The history of $\mathcal{I}(V_t; \mathcal{Z}_t)$ is fluctuating; it can decrease with measurement (around $t = 0.5$) and can increase

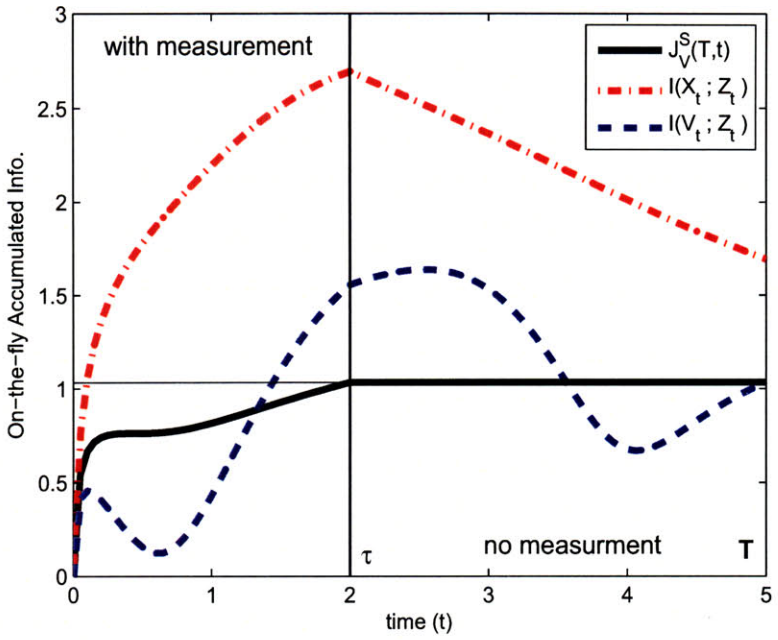

Figure 5-2: On-the-fly information by a partial measurement path $\mathcal{Z}_t$

without measurement (at $t = 2$), because information can be supplied/dissipated from/to the other state variables $X_t \setminus V_t$ via dynamics. Thus, this simple example clearly demonstrates the information supply/dissipation analysis in this section.

As a consequence, the filter form quantities, $\mathcal{I}(X_t; \mathcal{Z}_t)$ and $\mathcal{I}(V_t; \mathcal{Z}_t)$, are not good indicators of the accumulated information $\mathcal{I}(V_T; \mathcal{Z}_t)$; only the smoother form quantity $\mathcal{J}_V^S(T,t)$ accurately represents the accumulated information. □

5.3 Path Representation

The previous section suggests a formula to quantify the information reward for a continuous measurement history in a finite-dimensional linear system framework. This section shows how to relate the motion of a sensor in continuous space to a measurement history in the time domain, starting with the spatial interpolation method. This method describes the continuous field of the environmental variables in terms of a finite number of variables associated with the specified grid points.

This work assumes that the environmental variables at location $\mathbf{r}$ can be repre-

sented as a linear combination of those at a finite number of grid points $\mathbf{r}_i$'s:

$$\phi_t(\mathbf{r}) = \sum_{i=1}^{n_G} \lambda_i(\mathbf{r}, \mathbf{r}_i)\phi_t(\mathbf{r}_i) \tag{5.72}$$

where n_G is the number of grid points, $\phi_t(\cdot) \in \mathbb{R}^{n_E}$ represents the environmental variables associated with a given location at time t with n_E denoting the number of environmental variables associated with a single grid point, and the coefficient λ_i is determined as a function of the location vectors $\mathbf{r}$ and $\mathbf{r}_i$. This thesis considers time-invariant λ_i, although the results can be easily extended to time-varying cases. Many spatial interpolation techniques such as Kriging [28] and Gaussian processes regression (GPR) [29] provide expressions of the form in (5.72). In determining the coefficients for the linear combination, this thesis specifically considers the zero-mean GPR method, which is equivalent to Simple Kriging, with squared exponential covariance function. The coefficients $\lambda_i(\mathbf{r}, \mathbf{r}_i)$ are expressed as

$$\lambda_i(\mathbf{r}, \mathbf{r}_i) = \sum_{j=1}^{n_G} \alpha_{ij}\rho(\mathbf{r}, \mathbf{r}_j) \tag{5.73}$$

where $\rho(\mathbf{r}, \mathbf{r}_j)$ is defined as

$$\rho(\mathbf{r}, \mathbf{r}_j) \triangleq \exp\left[-\frac{1}{2l_x^2}(x - x_j)^2 - \frac{1}{2l_y^2}(y - y_j)^2\right] \tag{5.74}$$

in the two-dimensional space, and α_{ij} is the (i, j)-th element of the matrix $[\rho(\mathbf{r}_i, \mathbf{r}_j)]^{-1}$. The parameters l_x and l_y represent the correlation length scales in each direction. For an isotropic environment, $l_x = l_y$ can be assumed, but for the atmospheric application in this thesis, it is more reasonable to consider different length scales in each direction because the zonal and meridional variations of the weather are very different.

Under the assumption in (5.72), the environmental dynamics over the whole continuous space can be fully described by the dynamics of the finite number of variables at grid points. The state vector $X_t \in \mathbb{R}^{n_X}$, $n_X = n_G \times n_E$, is defined as

$$X_t = [\phi_t(\mathbf{r}_i)' \cdots \phi_t(\mathbf{r}_{n_G})']', \tag{5.75}$$

and this work considers linear dynamics for X_t as in (5.1).

Consider a sensor located at $\mathbf{r}$ at time t that receives measurement of $\phi_t(\mathbf{r})$. Since $\phi_t(\mathbf{r})$ is a linear combination of $\phi_t(\mathbf{r}_i)$'s, the observation equation for this sensor can be expressed as $Z_t = C(t)X_t + N_t$ where $C(t) \in \mathbb{R}^{n_E \times n_X}$ is

$$C(t) = [\lambda_1(\mathbf{r}, \mathbf{r}_1) I_{n_E} \quad \cdots \quad \lambda_{n_G}(\mathbf{r}, \mathbf{r}_{n_G}) I_{n_E}] \,. \tag{5.76}$$

For notational convenience, a single-sensor case will be considered in the later part of this chapter on; extension to multiple sensors is straightforward.

If a sensor is continuously moving, its motion is fully described by the time history of the location vector $\mathbf{r}(t)$. Thus, the effect of the sensor's motion on the estimation system is through the evolution of the observation matrix $C(t)$ due to changes in $\lambda_i(\mathbf{r}, \mathbf{r}_i)$'s in time. Consider a sensor moving along a specified path $\mathbf{p}_\tau = \{\mathbf{r}(t) : t \in [0, \tau]\}$ where $\mathbf{r}(t)$ is known for all $t \in [0, \tau]$. Then, the evolution of observation matrix $\mathcal{C}_\tau = \{C(t) : t \in [0, \tau]\}$ can be derived by relating $C(t)$ and $\mathbf{r}(t)$. Then, the information reward associated with this path, denoted as $\mathcal{J}_V(T, \tau; \mathbf{p}_\tau)$, can be computed by evaluating $Q_X(\tau; \mathcal{C}_\tau)$, which is the final value of the Riccati equation corresponding to observation matrix history $\mathcal{C}_\tau$, while $\mathcal{J}_0(\tau)$ and $\Delta_S(\tau)$ have been computed in advance independently of $\mathbf{p}_\tau$.

To account for the limited mobility of the sensor, the path is, in general, represented as a set of equations of the location vector and its time derivatives:

$$g_{dyn}(\mathbf{r}(t), \dot{\mathbf{r}}(t), \ddot{\mathbf{r}}(t), \mathbf{u}(t)) = \mathbf{0} \tag{5.77}$$

where $\mathbf{u}$ is the control input for the sensor motion. For instance, a two-dimensional holonomic motion of a UAV sensor platform with constant speed v can be written as

$$\dot{x}(t) = v \cos\theta(t), \qquad \dot{y}(t) = v \sin\theta(t) \tag{5.78}$$

where $\theta(t)$ is the flight path angle, which is treated as a control input in this model.

5.4 Path Planning Formulations

5.4.1 Optimal Path Planning

The optimal motion planning determines the path $\mathbf{p}_\tau$, or equivalently the time history of the control input, that maximizes the smoother form information reward $\mathcal{J}_V^S(T,\tau) = \mathcal{J}_0(\tau) - \frac{1}{2}\log\det(I + Q_X(\tau;\mathcal{C}_\tau)\Delta_S(\tau))$. The prior and posterior initial covariance P_0 and $P_{0|V}$, respectively, are computed first; $\mathcal{J}_0(\tau)$ and $S_0(\tau)$ are computed using these information. Then, the optimization problem only involving the computation of $Q_X(\tau;\mathcal{C}_\tau)$ is posed. This optimization problem is indeed a nonlinear *optimal control* problem (OCP) with a terminal cost functional. The control variables for this OCP are the controls for the sensor motion, e.g. $\theta(t)$ for two-dimensional holonomic motion, while there are two types of state variables: the vehicle position variables, x and y, and the entries of the $Q_X(t)$ matrix. The optimal path planning problem for a two-dimensional holonomic mobile sensor is stated as

$$\theta^\star(t) \in \arg\min_{\theta(t)} \log\det(I + Q_X(\tau;\mathcal{C}_\tau)\Delta_S(\tau)) \tag{5.79}$$

subject to

$$\begin{aligned} \dot{Q}_X &= AQ_X + Q_XA' + \Sigma_W - Q_XC(x,y)'\Sigma_N^{-1}C(x,y)Q_X \\ \dot{x} &= v\cos\theta,\ \dot{y} = v\sin\theta \\ Q_X(0) &= P_0,\ x(0) = x_0,\ y(0) = y_0 \end{aligned} \tag{5.80}$$

where $C(x,y)$ is expressed as a function of x and y to emphasize that dependency in time is only through the evolution of x and y. Regarding the size of this OCP, there is one control variable and the number of state variables is $n_X(n_X+1)/2 + 2$. Constraints in the sensor's motion such as endpoint restriction, waypoint requirement, nonholonomic aspect of motion can be easily incorporated by modifying the vehicle's dynamics and by imposing additional constraints. Also, multiple sensor problems can be dealt with by adding associated dynamic/kinematic constraints and by modifying

the expression of the observation matrix.

5.4.2 Information Field and Real-Time Steering

Optimal path planning gives a motion plan for maximum information reward, but, solving a nonlinear optimal control problem can incur substantial amount of computation cost, especially, when n_X is large. Thus, it is beneficial in practice to devise a computationally cheap feedback guidance law. One way to derive a real-time steering mechanism is to build some potential field and to move along the gradient of that field. The mutual information rate discussed in section 5.2.4 can be utilized to construct an *information potential field.* This type of information potential field extends a similar notion presented in [33], which derived the expression of $\frac{d}{dt}(\log\det Q_X^{-1}(t))$ and neglected terms unrelated to the observation matrix to build a potential field. This section builds a potential field with the smoother form information rate in (5.60), which consists of a single term explicitly dependent on the observation matrix.

For the two-dimensional holonomic sensor motion in (5.78), the guidance law is presented as

$$\theta_G(t) = \texttt{atan2}\left\{\frac{\partial}{\partial y}\left(\frac{d}{dt}\mathcal{J}_V^S(T,t)\right), \frac{\partial}{\partial x}\left(\frac{d}{dt}\mathcal{J}_V^S(T,t)\right)\right\} \tag{5.81}$$

where $\frac{d}{dt}\mathcal{J}_V^S(T,t)$ is the smoother form mutual information rate, and `atan2` denotes the four-quadrant arctangent.

Since the relationship between $C(x,y)$ and (x,y) is known, the mutual information rate can be particularly written as a function of spatial coordinates:

$$\frac{d}{dt}\mathcal{J}_V^S(T,t) = \frac{1}{2}\texttt{trace}\left\{R^{-1}C(x(t),y(t))\Pi(t)C(x(t),y(t))'\right\}, \tag{5.82}$$

and the gradient of the above expression can be evaluated accordingly. In case $C(x(t),y(t)) \in \mathbb{R}^{1\times n_X}$, namely, there is only one environmental variable of interest,

the spatial derivative can be written as

$$\frac{\partial}{\partial x}\left(\frac{d}{dt}\mathcal{J}_V^S(T,t)\right) = \Sigma_N^{-1}C(x(t),y(t))\Pi(t)\frac{\partial C(x(t),y(t))}{\partial x} \tag{5.83}$$

$$= \Sigma_N^{-1}C(x(t),y(t))\Pi(t)\mathbf{d}_C(x) \tag{5.84}$$

$$\frac{\partial}{\partial y}\left(\frac{d}{dt}\mathcal{J}_V^S(T,t)\right) = \Sigma_N^{-1}C(x(t),y(t))\Pi(t)\frac{\partial C(x(t),y(t))}{\partial y} \tag{5.85}$$

$$= \Sigma_N^{-1}C(x(t),y(t))\Pi(t)\mathbf{d}_C(y) \tag{5.86}$$

where $\mathbf{d}_C(x)$ and $\mathbf{d}_C(y)$ are n_X-dimensional column vectors whose i-th elements are

$$\mathbf{d}(x)_i = -l_x^{-2}\sum_j \alpha_{ij}\rho(\mathbf{r},\mathbf{r}_j)(x-x_j), \tag{5.87}$$

$$\mathbf{d}(y)_i = -l_y^{-2}\sum_j \alpha_{ij}\rho(\mathbf{r},\mathbf{r}_j)(y-y_j). \tag{5.88}$$

In case C is not a row vector, the relation in (5.68) suggests that the mutual information rate and its gradient are nothing more than the sum of those with individual row of the observation matrix.

5.5 Numerical Simulations

This section deals with numerical implementation of the proposed path planning formulations, to confirm that the linear estimation theoretic formula together with the path representation technique realizes the continuous measurement path for mobile sensors.

5.5.1 Scenarios

The path planning problem is posed for the linearized weather dynamics for some local region defined by the indices $(i,j) \in [L_{on}^{\min}, L_{on}^{\max}] \times [L_{at}^{\min}, L_{on}^{\max}]$. A linear invariant model is obtained by deriving the Jacobian matrix of the Lorenz-2003 dynamics in (2.18) (with $\rho_x = \rho_y = 2$) around the ensemble mean for ϕ_{ij}'s at the grid points in the local region. Thus, the state vector $X_t \in \mathbb{R}^{nx}$ represents the perturbation of

Table 5.1: Scenarios for continuous motion planning

Scenario	$[L_{on}^{\min}, L_{on}^{\max}]$	$[L_{at}^{\min}, L_{at}^{\max}]$	$n_G = n_X$	l_x	l_y	(x_0, y_0)	v (grid/hr)
1	[46,49]	[5,7]	12	1.5	1.0	(47.5, 6)	1/3
2	[47,50]	[12,14]	12	1.0	0.7	(49, 13)	1/2

the ϕ_{ij}'s from the ensemble mean. In this linear model, the dependence of the local dynamics on the evolution of the outside dynamics is ignored in deriving the Jacobian matrix (or A matrix). Instead, this effect is incorporated in the process noise term, i.e. the states on the boundary of the local region, which may be affected by outside dynamics more substantially, are assumed to be subject to larger process noise. The goal is to design a 6-hr flight path ($\tau = 6$ hrs) for a single sensor platform to improve the forecast over the right-most grid points in the local region in 72 hrs ($T = 72$ hrs). The motion of the sensor is described as 2-D holonomic motion and it flies at constant speed v grid/hr ($= 347v$ km/hr). The prior and posterior initial covariance matrices, P_0 and $P_{0|V}$ are provided by the EnSRF data assimilation scheme, where $P_{0|V}$ is computed by the ensemble augmentation method presented in Remark 3. Two scenarios with different configurations of local region, correlation length scale parameters, vehicle speed, and vehicle initial location, are considered. The sensing noise intensity $\Sigma_N = 0.0025$ is used, and other parameter values for each scenario are given in Table 5.1.

5.5.2 Results

Two proposed path planning methods, optimal path planning and gradient-based real-time steering, are compared with the myopic versions of them. Myopic path planning takes into account $\mathcal{I}(X_\tau; \mathcal{Z}_\tau)$ instead of $\mathcal{I}(V_T; \mathcal{Z}_\tau)$, the underlying hypothesis being that uncertainty reduction in the current estimate would lead to uncertainty reduction in the future forecast. More specifically, the optimal myopic solution minimizes $\log\det Q_X(\tau)$, and myopic real-time steering utilizes the filter form information

Table 5.2: Information rewards for different strategies

Scenario	Optimal	Real-Time	Myo. Opt	Myo. RT	Best SL	Worst SL
1	1.04	0.85	0.86	0.79	0.93	0.29
2	0.69	0.62	0.20	0.14	0.43	0.14

rate in (5.54) to construct an information potential field. Since the information dissipation term of the filter form information rate, $\frac{1}{2}\texttt{trace}(P_X^{-1} - Q_X^{-1})$ is not an explicit function of (x, y), the myopic real-time law results in a formula that has Q_X instead of Π in the gradient expression. Especially, at initial time, $\Pi(0) = P_0 - P_{0|V}$, while $Q_X(0) = P_0$.

Each of the two optimal control problem is formulated as a nonlinear program (NLP) by parameterizing the control history as a piecewise linear functions with 12 segments. TOMLAB/SNOPT v6.0 [48] is used to solve NLPs; real-time steering solutions, and various straight-line solutions are used as initial guess for the optimization. Both optimization solutions are obtained within two minutes (per initial guess) and satisfy first-order optimality criteria. Also, as references, the best and the worst straight-line paths are also considered. The best straight line solves an NLP to find a constant θ_0 that maximizes the smoother form information reward, assuming vehicle dynamics of $\dot{x} = v\cos\theta_0$, $\dot{y} = v\sin\theta_0$.

Table 5.2 represents the information rewards $\mathcal{J}_V^S(T, \tau)$ for all the methods considered. It is first found that the real-time steering provides relatively good performance for both scenarios, while the two myopic strategies provide very poor performance in Scenario 2. It is also noticed that the best straight-line solutions offer a better performance than the real-time steering for Scenario 1.

Figures 5-3 and 5-4 depict the sensor trajectories in the first scenario overlaid with the initial information potential fields – the smoother form information field in Figure 5-3, and the filter form information field in Figure 5-4. Similarly, Figures 5-5 and 5-6 show the trajectories for the second scenario. Note that in scenario 2 the shape of the smoother form information field substantially differs from the filter

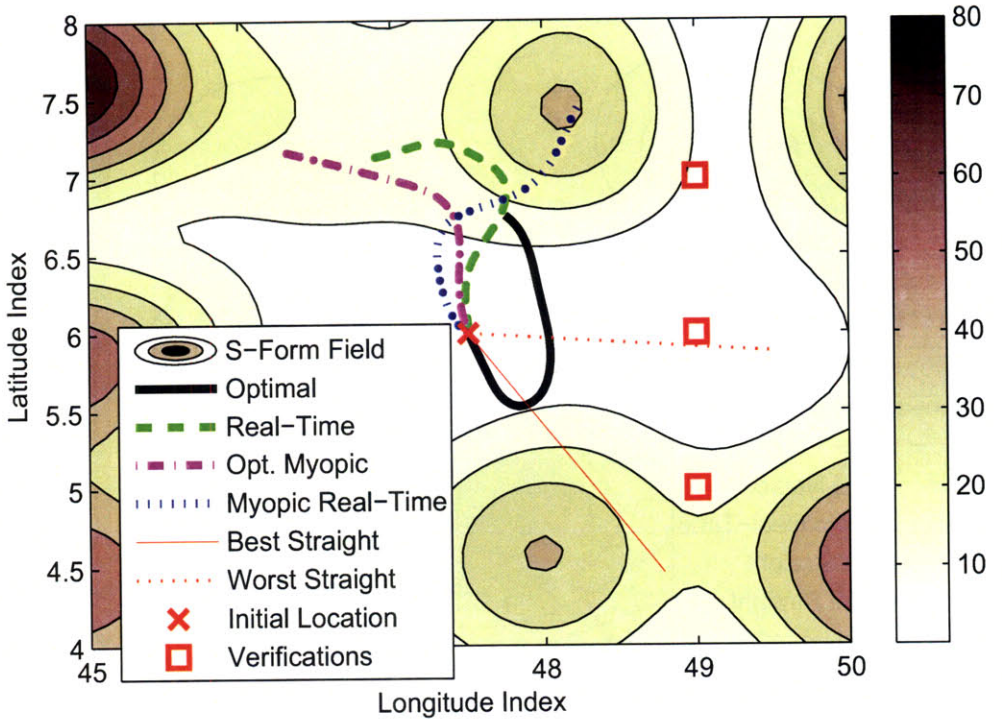

Figure 5-3: Sensor trajectories for Scenario 1 overlaid with the smoother form information potential field

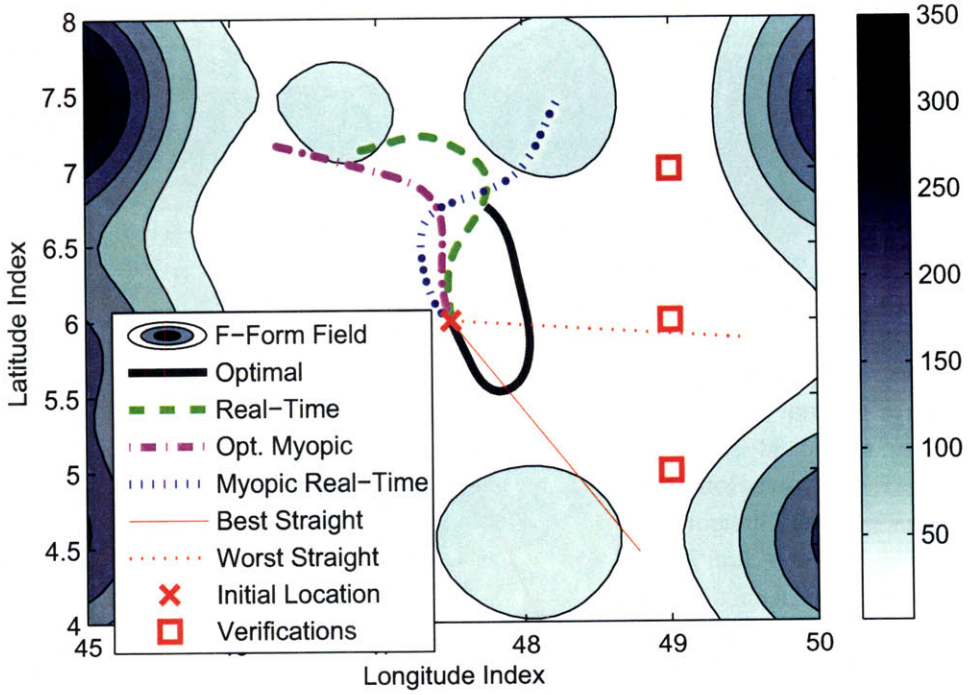

Figure 5-4: Sensor trajectories for Scenario 1 overlaid with the filter form information potential field

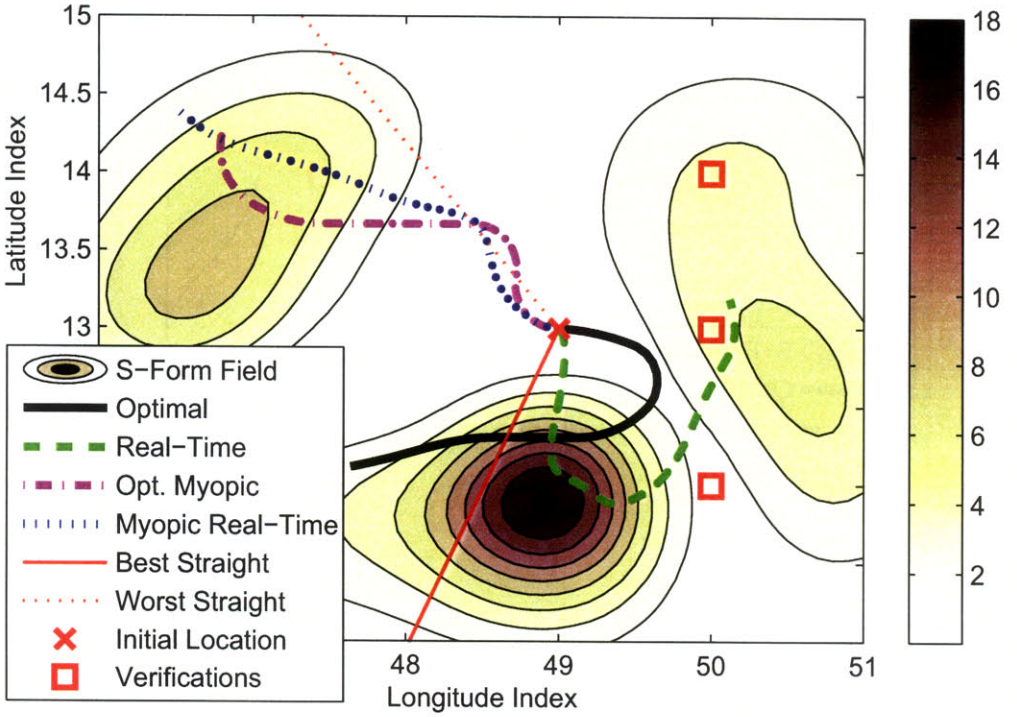

Figure 5-5: Sensor trajectories for Scenario 2 overlaid with the smoother form information potential field

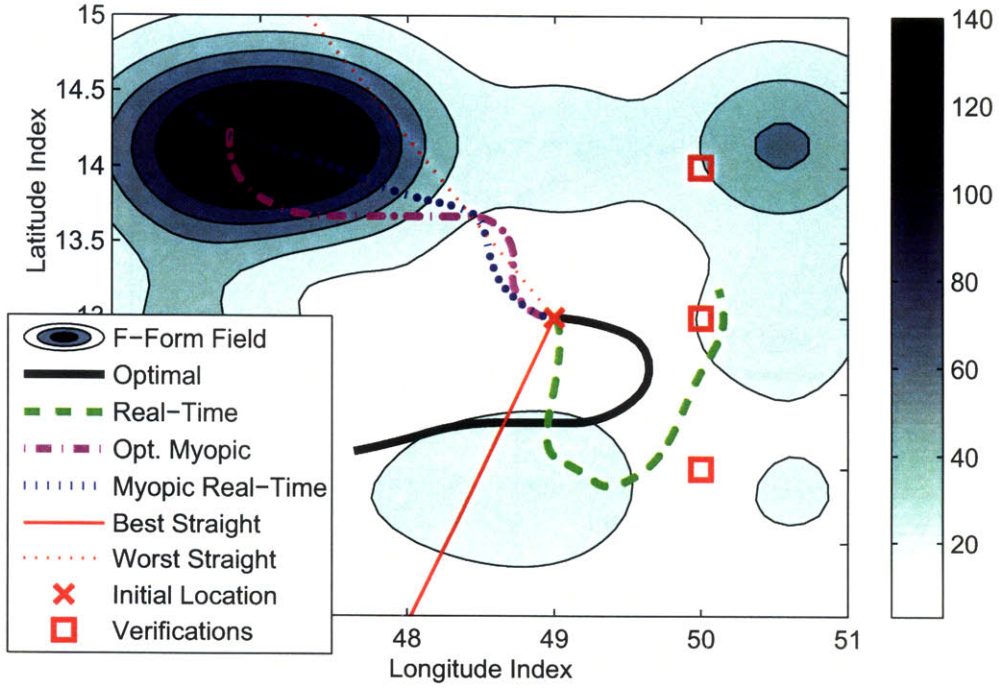

Figure 5-6: Sensor trajectories for Scenario 2 overlaid with the filter form information potential field

form one, while they are similar in scenario 1. Thus, in Scenario 2, moving toward the information-rich region in the north-western corner from the filter point of view is never a good strategy from the smoother point of view, because most of the smoother-form information is concentrated in the southern part of the region. This example confirms that reducing the uncertainty in the current state variables may not be a particularly good way for reducing the uncertainty in the future verification variables.

Figures 5-7, 5-8, and 5-9 give snapshots of the vehicle trajectories in scenario 1 for the optimal, real-time steering, and best straight-line strategies, respectively, overlaid with the information field at the corresponding time instances. Note that the shape of the information field changes a lot as measurements are taken along the path. Since the smoother form information rate has only one term that is directly dependent on the observation matrix, the information field stays the same if no measurement is taken. This is not true for the filter form field, although in this particular example that background variation of field happens not to be significant because of slow dynamics and small process noise.

In Figure 5-8, the real-time steering leads the sensor to climb up to the information-rich peak in the north; at $t = 3$hr the sensor has taken most of information around the north peak, which is not a peak any more. The sensor then steers to the west, but time is limited to extract much information. However, the optimal solution in Figure 5-7 first heads toward to a south peak; at $t = 2$hr the sensor turns back to the north in order to gather the information around the north peak. Notice that the initial behavior of the best straight-line solution in Figure 5-9 is very similar to the optimal solution, which leads to good performance. Thus, in terms of the performance, the real-time steering may not be better than an optimized simple strategy, because it cannot take into account a global shape of the information field. But, since the real-time steering consumes a trivial amount of computation resource than solving an optimization problem, it can be a useful suboptimal strategy in practice. (See Appendix D for more results.)

Figures 5-10 and 5-11 illustrate the time histories of the accumulated information $\mathcal{I}(V_T; \mathcal{Z}_t)$ for $t \in [0, \tau]$ for different strategies in Scenario 1 and 2, respectively. For

the first scenario, it can be seen that the optimal solution gains a large amount of information during two distinctive time intervals, which correspond to visiting two information-rich regions in the south and north. In scenario 2, a large amount information is obtained only in the interval $t = 2$–4 hr because information is concentrated in one region in the scenario as shown in Figure 5-5.

5.6 Additional Discussion

5.6.1 Backward Selection and Smoother Form

Since the information identity in Proposition 4 holds for more general nonlinear, non-Gaussian, discrete (or hybrid) systems, it is also possible to apply the smoother form in this chapter to the targeting problem presented in Chapter 3 and 4. This section compares the key characteristics of the backward selection and the smoother form from the perspective of Bayesian inference, as it provides an important insight on the reasonable methodology for different levels of abstraction of the problem space.

Figure 5-12 illustrates the key conceptual differences amongst all the methodologies considered in this thesis – forward and backward selections, and filter and smoother forms – using a graphical structure that represents conditional independence. In this graph, if two nodes are separated by some subgraph, the entities in these two nodes are conditionally independent each other conditioned on the entities in the subgraph. For instance, $\mathcal{Z}_\tau = \{Z_0, \cdots, Z_\tau\}$ and X_T are conditionally independent each other for a given X_τ, as they come to belong to different subgraphs if the node X_τ is deleted.

The goal of informative selection is to find the best measurement sequences $\mathcal{Z}_\tau \triangleq \{Z_0, \cdots, Z_\tau\}$, which can be either finite-dimensional or continuous, to maximize the mutual information between the future verification variable V_T and itself. To do this, the backward selection first performs a *single backward inference* to build the *joint* distribution of the measurement choices over $[0, \tau]$; then, the remaining decision is a *static combinatorial selection* problem in the domain of $\mathcal{Z}_\tau$. On the other hand,

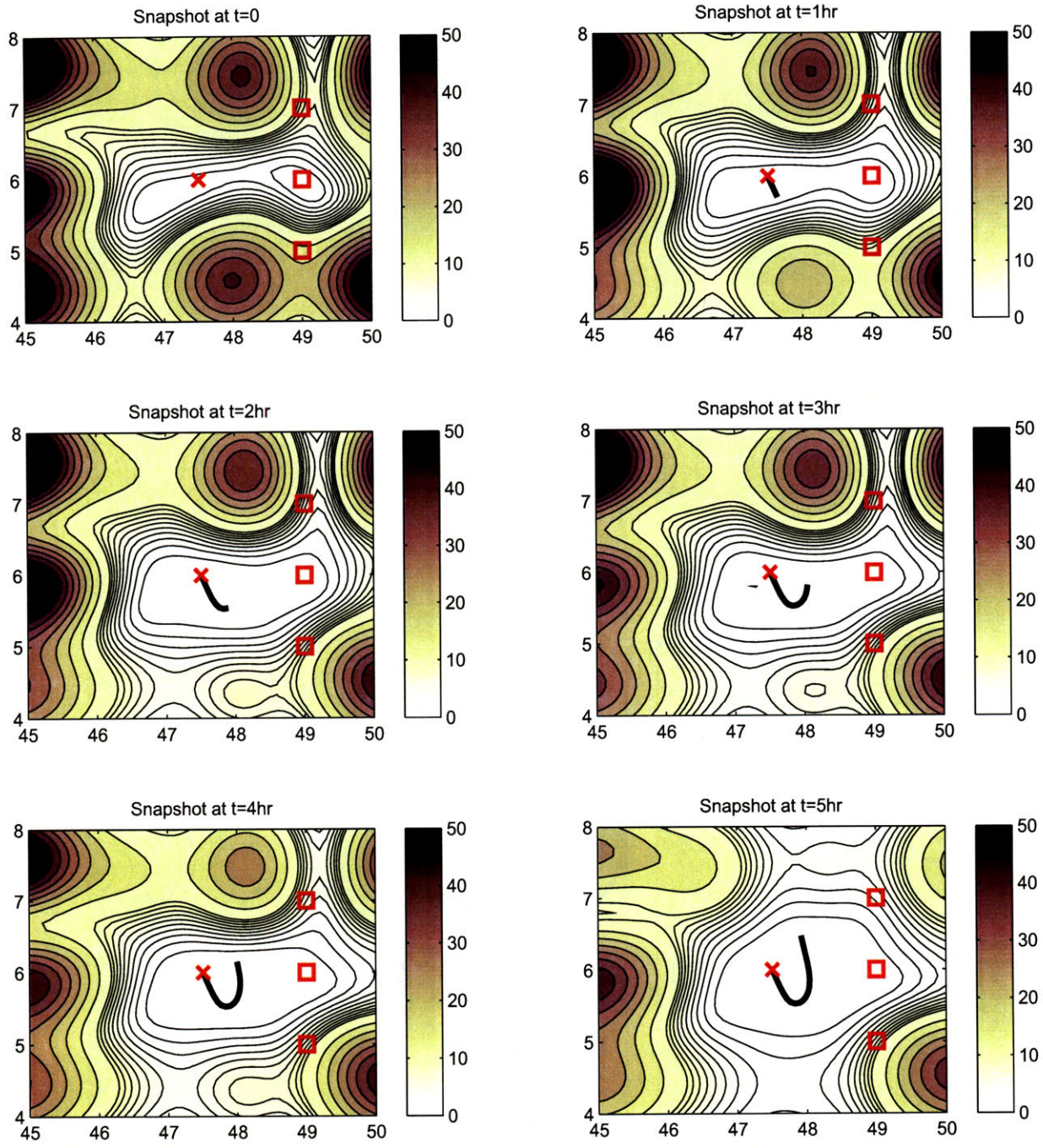

Figure 5-7: Snapshots of the optimal trajectory with evolution of information field (Scenario 1)

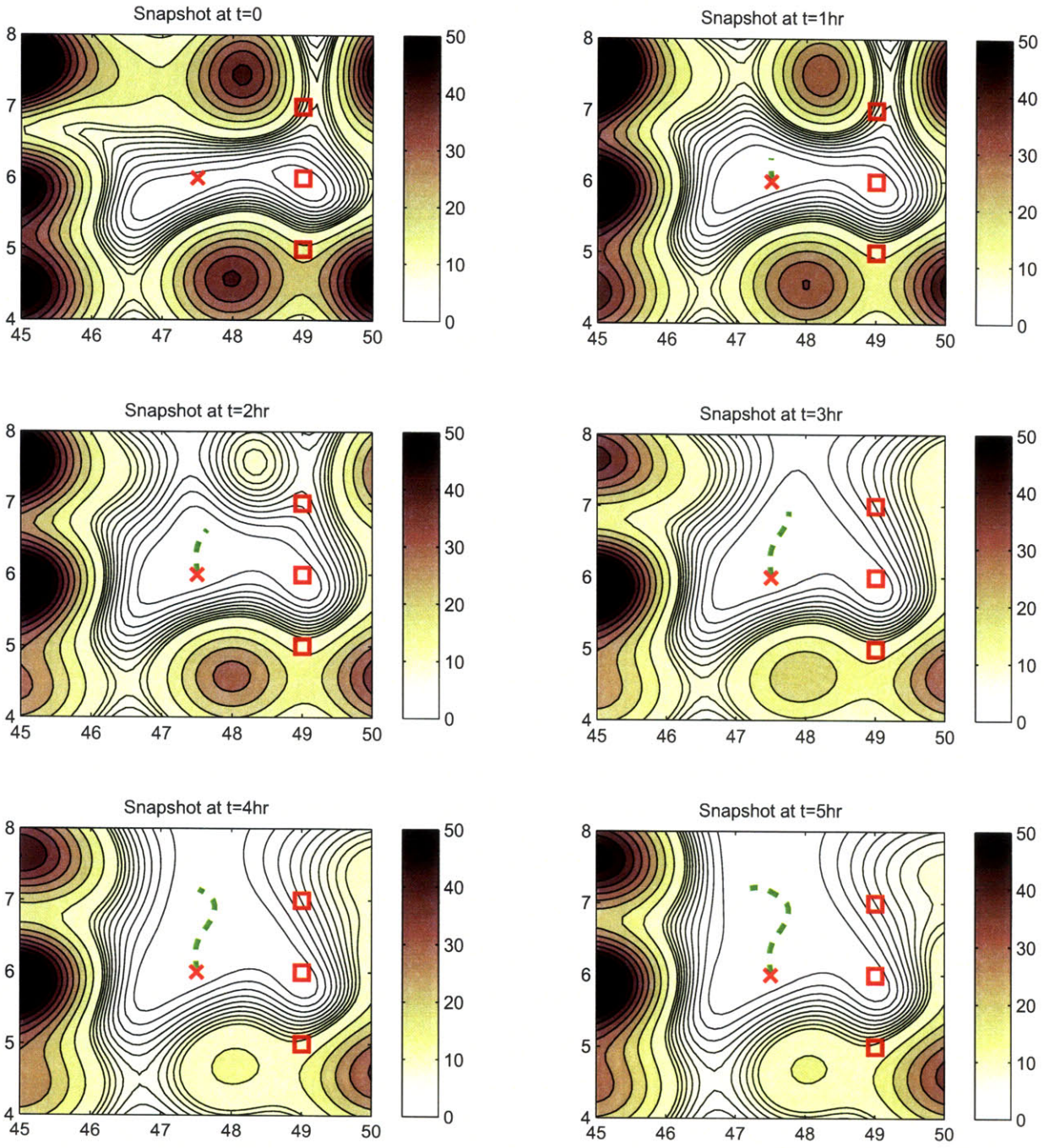

Figure 5-8: Snapshots of the trajectory of real-time steering with evolution of information field (Scenario 1)

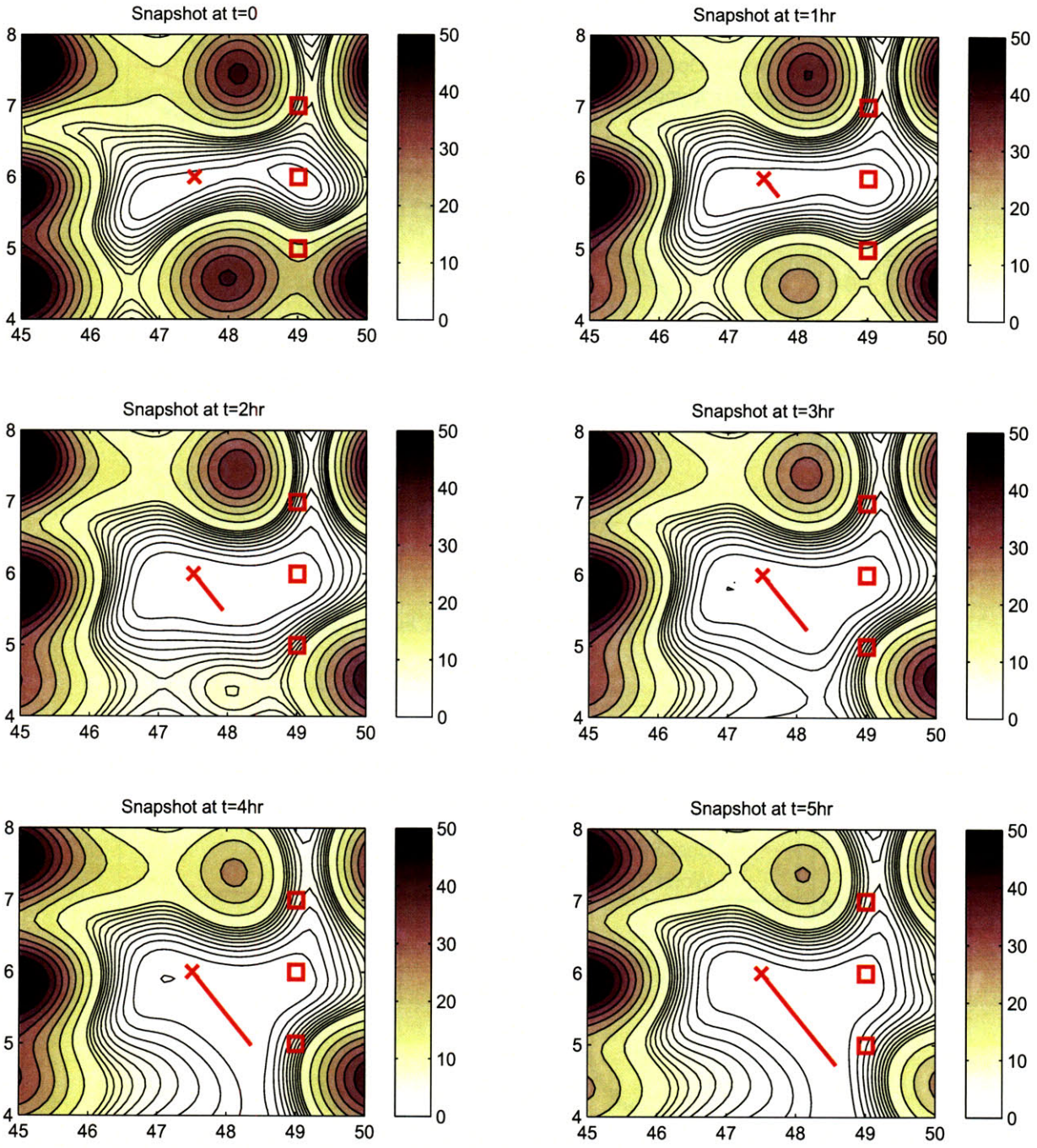

Figure 5-9: Snapshots of the best straight-line trajectory with evolution of information field (Scenario 1)

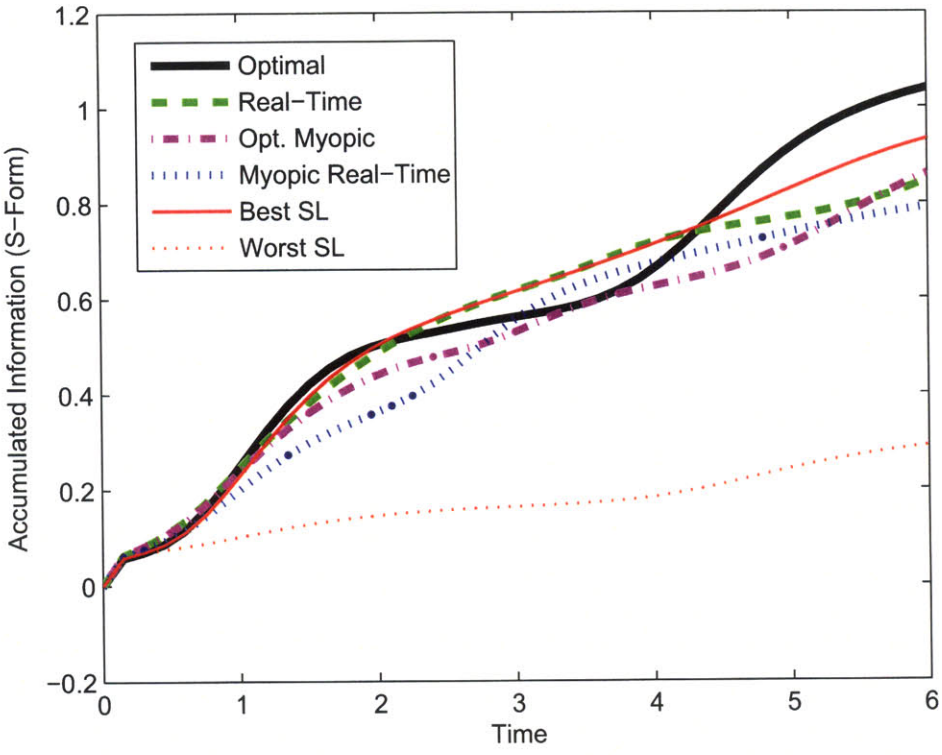

Figure 5-10: Time histories of information accumulation in Scenario 1

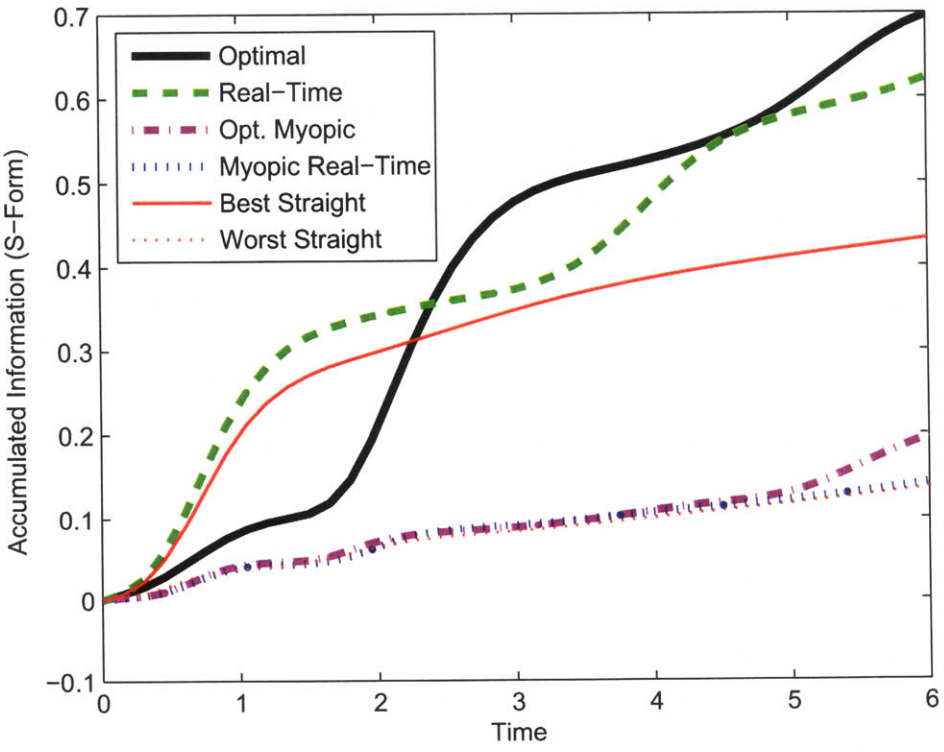

Figure 5-11: Time histories of information accumulation in Scenario 2

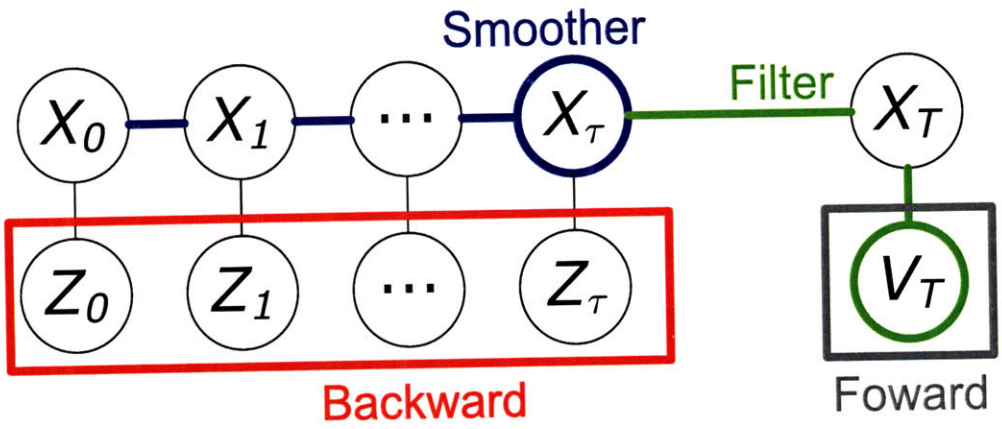

Figure 5-12: Graphical structure of the presented approaches

the smoother form performs a *single backward inference* to figure out the *marginal* distribution of the state at the end of the planning horizon, X_τ; then conducts a *combinatorial* number of *forward inferences* that incorporates the effects of the measurement choices.

Therefore, the main source of the computational expense of the backward selection is quantification of the conditional (on V_T) entropy of $\mathcal{Z}_\tau$, while that of the smoother form is forward inference over $[0, \tau]$ and quantification of conditional (on $\mathcal{Z}_\tau$) entropy of X_τ. Hence, in case the forward inference is very expensive (e.g. nonlinear integration of large-scale differential equations) and/or the state dimension is large incurring significant cost in the computation of the entropy, the backward selection can provide better computational efficiency than the smoother form. In contrast, in case the forward inference process is expedited by some approximations (e.g. linearization or learning), and/or the state dimension is smaller than the measurement dimension, the smoother can be more effective than the backward selection, even for the discrete decision problem.

Recall that this thesis addresses problems at two levels of abstraction: the targeting problem for a large-scale decision with nonlinear dynamics, and the motion planning problem for a smaller-scale decision with linear dynamics as an approximation of nonlinear dynamics. It should be noted that for this two-level setting, the presented combination of strategies – backward selection for the targeting and the smoother form for the motion planning, renders the best computational efficiency,

even in case the smaller-scale decision is posed in the discrete domain.

5.7 Conclusions

A methodology for continuous motion planning of sensors for information forecast was presented. The smoother form formula of the information reward, which exhibits better computational efficiency than a more conventional filter form formula, enabled calculation of accumulated information and the rate of information accumulation on-the-fly. An optimal path planning formulation was presented, as was a real-time steering law based on information potential field. Numerical examples using a linearized Lorenz-2003 model validated the proposed motion planning methodology.

Chapter 6

Sensitivity Analysis for Ensemble-Based Targeting

This chapter analyzes the sensitivity of the targeting solutions in Chapter 3 to the ensemble size. Monte-Carlo experiments with the Lorenz-95 targeting example characterizes three important aspects of sensitivity: discrepancy of the predicted and actual information reward value, performance degradation of the targeting solution, and inconsistency in the targeting solution with respect to the choice of ensemble. In addition, based on the analysis of the statistics of entropy estimation, this chapter proposes new predictors of the degree of impact that limitation of ensemble size might have on the solution optimality: the range-to-noise ratio (RNR) and the probability of correct decision (PCD). Numerical investigation of the Lorenz-95 targeting example validates the effectiveness of these predictors.

6.1 Limited Ensemble Size

The quality of the state representation by statistical ensembles tends to be enhanced as the ensemble size increases, while theds to be degraded as the state dimension increases. Thus, the ratio of the ensemble size to the state dimension, L_E/L_X can be used as the simplest indicator of the quality of a set of ensembles for representing the true state, although how to select each ensemble member to build the ensemble set

of a given size also affects the quality of resulting ensembles.

Consider this ratio L_E/L_X for the numerical examples in the previous chapters, the Lorenz-2003 model with $\rho_x = \rho_y = 1,\ 2$ have 324 and 1224 state variables, respectively; $L_E = 1024,\ 1224$ ensembles were used for data assimilation of the corresponding system. These sets of values of L_X and L_E lead to $L_E/L_X = 3.2$ and 1, which is $\sim \mathcal{O}(1)$. It should be noted that for a realistic weather model, L_E/L_X is typically much smaller than $\mathcal{O}(1)$; the typical size of operational ensemble is maximum $\mathcal{O}(100)$, while the state dimension is $\mathcal{O}(10^6)$, which results in L_E/L_X being less than $\mathcal{O}(10^{-4})$. Thus, the accuracy of the ensemble representation of the realistic weather model can be very low compared to that of the Lorenz-2003 models used in this work.

This limitation in the ensemble size is caused by the computational expense of integrating forward a large-scale nonlinear system, and storing the large ensemble data sets; there have been studies that either tried to speed up the ensemble forecasting process or to figure out and mitigate the effect of small L_E/L_X on the quality of forecast and data assimilation [30, 31]. The work in [30] showed that the impact of ensemble size on the forecast performance is highly dependent on the choice of metric. The ensemble mean is less sensitive to the ensemble size, while the spread-skill relation and the outlier statistic are more sensitive to the ensemble size. Also, [31] suggested a distance-dependent ensemble Kalman Filter (EnKF) that mitigates the effect of erroneous estimation of correlation coefficients due to limited ensemble size on the quality of data assimilation.

In spite of the limitation of ensemble diversity, the ensemble forecast system has been an attractive concept for numerical weather prediction because it can incorporate the state-dependent uncertainty information into data assimilation in a relatively efficient way compared to the extended Kalman filter. However, the success of the ensemble-based approaches in data assimilation and forecasting (using $\mathcal{O}(10^2)$ ensemble members) does not necessarily mean that the currently available ensemble size is sufficiently large enough to create reliable targeting decisions. This is because the quantity of primary interest in data assimilation is the accuracy of tracking the mean behavior of the weather dynamics, whereas in the targeting problem the quantity

of primary interest is the accuracy of the uncertainty (or covariance) representation. Given this distinguishing aspect of the targeting problem from data assimilation, it is important to conduct a sensitivity analysis of the targeting algorithms developed in Chapters 3 and 4. This chapter focuses on the analysis of the sensing point targeting algorithm in Chapter 3, because its extension to the multi-sensor platform targeting algorithm is straightforward.

6.2 Effects of Limited Ensemble Size on Targeting

Recall that the forward and the backward ensemble-based targeting algorithms are written as:

$$\mathbf{s}^{\star}_{F,En} = \arg\min_{\mathbf{s}\in\mathbb{S}_n} \frac{1}{2}\log\det\left(\frac{1}{L_E-1}\widetilde{\mathbf{X}}_{V|Z_{\mathbf{s}}}\widetilde{\mathbf{X}}'_{V|Z_{\mathbf{s}}}\right), \tag{6.1}$$

$$\begin{aligned}\mathbf{s}^{\star}_{B,En} = \arg\max_{\mathbf{s}\in\mathbb{S}_n} \frac{1}{2}\log\det\left(\frac{1}{L_E-1}\widetilde{\mathbf{X}}_{X_{\mathbf{s}}}\widetilde{\mathbf{X}}'_{X_{\mathbf{s}}} + R_{\mathbf{s}}\right)\\ - \frac{1}{2}\log\det\left(\frac{1}{L_E-1}\widetilde{\mathbf{X}}_{X_{\mathbf{s}}|V}\widetilde{\mathbf{X}}'_{X_{\mathbf{s}}|V} + R_{\mathbf{s}}\right).\end{aligned} \tag{6.2}$$

Note that for each measurement candidate $Z_{\mathbf{s}}$ with $|\mathbf{s}| = n$, the forward ensemble targeting computes the covariance of a $M(= |V|)$-dimensional random vector using L_E ensembles, while the backward one computes the covariances of a n-dimensional random vector using the same number of ensembles. Accuracy of the covariance estimate using samples is dependent on both the number of samples and the size of the random vector (see section 6.3.1 for further details.) Thus, the forward and the backward algorithms represent different characteristics regarding the sensitivity to the ensemble size. The first step is an investigation of well-posedness of the forward and the backward algorithms.

6.2.1 Well-Posedness

The first concern regarding the limited ensemble size is *well-posedness* of the algorithm, since the log det function is not defined for a rank-deficient matrix. Note that

for a (perturbation ensemble) matrix $\widetilde{\mathbf{X}} \in \mathbb{R}^{k \times L_E}$

$$\mathtt{rank}(\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}') = \mathtt{rank}(\widetilde{\mathbf{X}}) \leq \min\{k, L_E\}. \tag{6.3}$$

A *necessary* condition for $\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}'$ to be full-rank is $k \leq L_E$; thus, k needs to be larger enough than L_E to have positive definite $\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}'$. On the other hand,

$$\mathtt{rank}(\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}' + \mathbf{R}) = k, \tag{6.4}$$

for a positive definite matrix $\mathbf{R}$, whether or not $\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}'$ is full-rank. From these facts, note that:

1. The forward algorithm might suffer from rank-deficiency unless the ensemble size is much larger than the size of the verification region (i.e., $M \ll L_E$).

2. The backward algorithm is not subject to rank-deficiency for nontrivial measurement noise with $R_s > 0$.

3. In case of perfect measurement (i.e., $R_s \equiv 0$), the backward algorithm can also be subject to rank-deficiency unless the ensemble size is sufficiently larger than the number of sensing points (i.e., $n \ll L_E$).

The above three aspects indicate that the backward algorithm is preferred to the forward algorithm in terms of the well-posedness of the algorithm and the accuracy of the covariance estimate. The rationale is that: a) most sensors have noise that cannot be ignored in the targeting decision, b) M tends to be larger than n, because the verification region is determined by the mission objective while the number of sensing points[1] can be tuned by the designer. As will be shown in section 6.3.1, the sample estimate of a smaller-size covariance matrix is more accurate than that of a larger-size covariance matrix. Since the backward selection offers better computational efficiency (as discussed in Chapter 3) and less possibility of ill-posedness as discussed in this

[1]In a multi-sensor platform scenario, n in this well-posedness discussion is not the total number of sensing points but the size of the covariance matrix whose determinant is calculated, i.e. $n_T n_H$ using terminology in section 4.3.

section, the analysis in the later part of this chapter will focus on the backward algorithm.

6.2.2 Numerical Studies

Well-posedness of the backward algorithm does not mean that it provides *the* optimal solution independent of the ensemble size. Even in case $n \ll L_E$, inaccuracy in the covariance estimate might lead to an incorrect targeting decision. In this section, Monte-Carlo experiments are performed to quantify the degree of performance degradation caused by the limited ensemble size. The same ensemble targeting scenario as in section 3.4.1 is considered: an EnSRF with 1024 ensemble members is used for the data assimilation of the Lorenz-95 model ($L_X = 324$) with 93 routine observations; the sizes of the verification region and the search space are $M = 10$ and $N = 108$, respectively. The number of sensing point is fixed to $n = 2$.

The (backward) ensemble targeting is conducted with various ensemble sizes. The *full* ensemble set in the targeting problem is constructed by the ensemble augmentation process (section 3.1.1) using the whole ensemble members used for data assimilation. Denote the full ensemble set as $\widetilde{\mathbf{X}}^{L_E^0}_{X_S \cup V}$, and the corresponding ensemble size as L_E^0; i.e., $L_E^0 = 1024$ in this experiment. This work assumes that the covariance approximation using the full ensemble set:

$$\mathbf{P}(X_{\mathbf{s}}|\cdot) \approx \frac{1}{L_E^0 - 1} \widetilde{\mathbf{X}}^{L_E^0}_{X_{\mathbf{s}}|\cdot} \left(\widetilde{\mathbf{X}}^{L_E^0}_{X_{\mathbf{s}}|\cdot} \right)' \tag{6.5}$$

is sufficiently close to the true value, and regards the targeting solution using the full set as the true optimal solution.

For another ensemble size L_E, 50 different ensemble combinations are randomly chosen from the full ensemble set as

$$\widetilde{\mathbf{X}}^{L_E}_{X_S \cup V} = \widetilde{\mathbf{X}}^{L_E^0}_{X_S \cup V} \times U_{L_E} \tag{6.6}$$

with some $U_{L_E} \in \{0,1\}^{L_E^0 \times L_E}$ that satisfies

$$U'_{L_E}\mathbf{1}_{L_E^0} = \mathbf{1}_{L_E}, \text{ and } U_{L_E}\mathbf{1}_{L_E} \leq \mathbf{1}_{L_E^0}. \tag{6.7}$$

The ensemble matrix in (6.6) is used to evaluate both the prior and the posterior covariances in the targeting decision. This specifically means that the posterior ensemble $\widetilde{\mathbf{X}}^{L_E}_{X_\mathbf{s}|V}$ is not taken from $\widetilde{\mathbf{X}}^{L_E^0}_{X_\mathbf{s}|V}$ but is computed using the ensemble update formula (2.17) with the reduced ensemble set in (6.6).

Two quantities are defined to represent the performance of the targeting solution with a limited number ensemble members: the *predicted* optimal reward and the *actual* optimal reward. The predicted optimal reward is the perceived optimal objective value in the targeting with a limited ensemble size:

$$\begin{aligned}\widehat{\mathcal{I}}^\star_{L_E} \triangleq \max_{\mathbf{s}\in\mathbb{S}_n} \; & \frac{1}{2}\log\det\left(\frac{1}{L_E-1}\widetilde{\mathbf{X}}^{L_E}_{X_\mathbf{s}}(\widetilde{\mathbf{X}}^{L_E}_{X_\mathbf{s}})' + R_\mathbf{s}\right) \\ & - \frac{1}{2}\log\det\left(\frac{1}{L_E-1}\widetilde{\mathbf{X}}^{L_E}_{X_\mathbf{s}|V}(\widetilde{\mathbf{X}}^{L_E}_{X_\mathbf{s}|V})' + R_\mathbf{s}\right),\end{aligned} \tag{6.8}$$

while the actual optimal reward is the value that the optimal solution of (6.8) will exhibit with "true" covariance (equivalently, with full ensemble set):

$$\begin{aligned}\bar{\mathcal{I}}^\star_{L_E} \triangleq & \frac{1}{2}\log\det\left(\frac{1}{L_E^0-1}\widetilde{\mathbf{X}}^{L_E^0}_{X_{\mathbf{s}^\star_{L_E}}}(\widetilde{\mathbf{X}}^{L_E^0}_{X_{\mathbf{s}^\star_{L_E}}})' + R_\mathbf{s}\right) \\ & - \frac{1}{2}\log\det\left(\frac{1}{L_E^0-1}\widetilde{\mathbf{X}}^{L_E^0}_{X_{\mathbf{s}^\star_{L_E}}|V}(\widetilde{\mathbf{X}}^{L_E^0}_{X_{\mathbf{s}^\star_{L_E}}|V})' + R_\mathbf{s}\right),\end{aligned} \tag{6.9}$$

where $\mathbf{s}^\star_{L_E}$ is the solution of the optimization in (6.8).

Figure 6-1 compares the predicted optimal reward and the actual optimal reward for various L_E. The average is taken over the different ensemble combinations, and the errorbar represents the 5th and the 95th percentiles. As a reference, the average predicted and actual reward for the random selection strategy is also plotted, while the red horizontal line crossing 0.84 in the y-axis represents the mutual information value for the optimal solution computed with the full ensemble set.

Note that there is *discrepancy* between the predicted and actual information re-

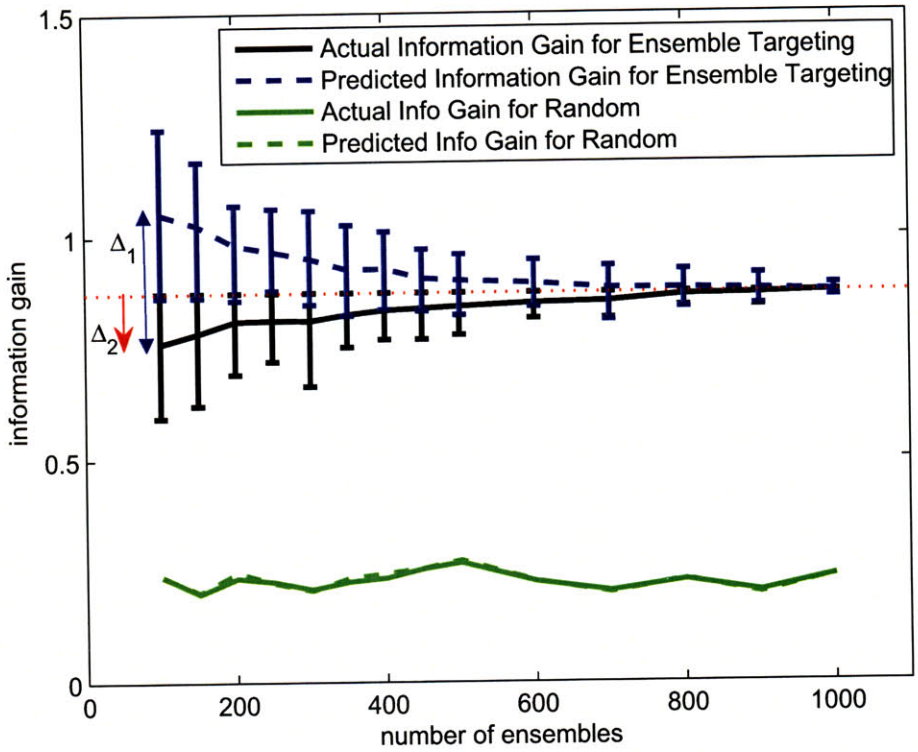

Figure 6-1: Errorbar plots of predicted/actual optimal reward values: Small ensemble size increases the gap between the predicted and the actual objective values Δ_1, decreases average performance Δ_2, and leads to large variation with respect to ensemble selection Δ_3.

ward values (Δ_1 in the figure), which increases as the ensemble size is reduced. This discrepancy indicates the magnitude of the estimation error by the limitation of ensemble size. It is found that the average magnitude of discrepancy increases to 0.29 for $L_E = 100$. In addition to the discrepancy, a small ensemble size causes *performance degradation* (Δ_2 in the figure). The average value of performance degradation grows to 13% (= 0.11 in the raw objective value) of the true optimal solution value as L_E decreases to 100. Another aspect that should be noted is *inconsistency* among the targeting solutions with a small number of ensemble members, which is represented by the size of the error bar. For $L_E = 100$, the size of the error bar increases to 32%(= 0.27 in raw objective value) of the true optimal solution value. It can be also found that the predicted optimal reward tends to be larger than the true optimal solution. As will be discussed in section 6.3.1, the variance of the mutual information

estimate does not vary too much for different measurement selections for a given ensemble size; but, it is highly dependent on the ensemble size. Thus, as the ensemble size decreases, the likelihoods of the ensemble approximation overestimating or underestimating the true covariance both increase. As the targeting process searches for the largest value, it tends to select the overestimated mutual information; this leads to the predicted optimal reward being larger than the true optimal reward.

To ensure that the sensitivity aspects found in Figure 6-1 is caused by the limitation in the ensemble size rather than the choice of the information metric or the ensemble selection mechanism, additional experiments are conducted: First, instead of using the ensembles from the EnSRF data assimilation, the ensemble matrices are created by random sampling from a known true covariance. In this simulation, the results turn out to represent similar sensitivity features (although they are not plotted in this thesis). Second, instead of the mutual information, a conventional trace measure that was employed in [4] is used for targeting. In this case, the uncertainty reduction of the verification region is represented by the difference between the trace of the prior and the posterior covariance matrix of the verification variables. Since the commutativity does not hold for this trace metric, the forward method is used to find the optimal solution. The predicted and the actual optimal trace metric are defined as

$$\widehat{\mathcal{T}}^{\star}_{L_E} \triangleq \max_{\mathbf{s}\in\mathbb{S}_n} \frac{1}{L_E-1}\mathtt{trace}\left\{\widetilde{\mathbf{X}}^{L_E}_{V}(\widetilde{\mathbf{X}}^{L_E}_{V})' - \widetilde{\mathbf{X}}^{L_E}_{V|Z_{\mathbf{s}}}(\widetilde{\mathbf{X}}^{L_E}_{VZ_{\mathbf{s}}})'\right\}, \tag{6.10}$$

and

$$\bar{\mathcal{T}}^{\star} \triangleq \max_{\mathbf{s}\in\mathbb{S}_n} \frac{1}{L_E-1}\mathtt{trace}\left\{\widetilde{\mathbf{X}}^{L_E}_{V}(\widetilde{\mathbf{X}}^{L_E}_{V})' - \widetilde{\mathbf{X}}^{L_E}_{V|Z_{\mathbf{s}^{T\star}_{L_E}}}(\widetilde{\mathbf{X}}^{L_E}_{VZ_{\mathbf{s}^{T\star}_{L_E}}})'\right\}, \tag{6.11}$$

where $\mathbf{s}^{T\star}_{L_E}$ is the optimal solution of (6.10). Figure 6-2 shows similar sensitivity features identified for the mutual information metric – discrepancy, performance degradation, and inconsistency. Moreover, in this example, the size of verification region is larger than the number of sensing points; the impact of the limited ensemble size becomes more significant in the targeting with the trace measure that evaluates covariances of a larger vector using ensembles.

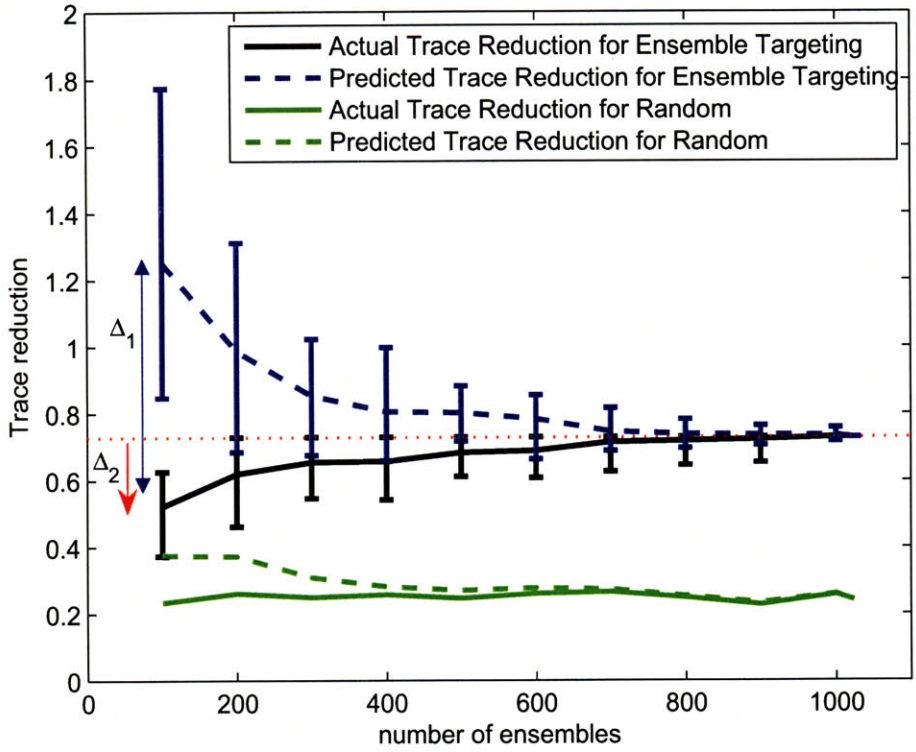

Figure 6-2: Error bar plots of predicted/actual optimal objective values when using trace measure: Similar sensitivity characteristics to the ensemble size are found with the conventional trace metric.

6.3 Analysis of Effects of Limited Ensemble Size

Given the experimental findings of the optimality degradation and the solution inconsistency due to the limited ensemble size, this section investigates how the ensemble size is related to the solution optimality of the targeting algorithm and discusses why nontrivial amount of inconsistencies can occur, even for the cases where L_E is two orders of magnitude larger than n.

6.3.1 Sample Entropy Estimation

As the simplest analysis, this section starts with the estimation of population entropy $h \equiv \log p$ (the factor 2 and the additive term are neglected for notational simplicity; but they can be incorporated without any difficulty) and the population variance p for comparison for a scalar Gaussian random variable X. Estimation is assumed to

be based on $L_E = m+1$ randomly drawn samples (ensembles)[2]. *Unbiased* estimators for these two quantities exist [49]:

$$\widehat{p} = \frac{1}{m}\sum_{i=1}^{m+1}(x_i - \bar{x})^2 \tag{6.12}$$

$$\widehat{h} = \log\frac{m\widehat{p}}{2} - \psi\left(\frac{m}{2}\right) \tag{6.13}$$

where x_i's and $\bar{x}$ denote each sample realization and the sample mean, and $\psi(\cdot)$ is a digamma function defined as $\psi(x) = \frac{d}{dx}(\log\Gamma(x)) = \frac{\Gamma'(x)}{\Gamma(x)}$.

Since $\frac{m\widehat{p}}{p} \sim \chi^2_m$, the pdfs of the estimation errors $\widetilde{p} \equiv \widehat{p} - p$ and $\widetilde{h} \equiv \widehat{h} - h$ are expressed as

$$f_{\widetilde{p}}(x) = \frac{m}{p} f_{\chi^2_m}\left(\frac{m}{p}(x+p)\right), \tag{6.14}$$

$$f_{\widetilde{h}}(x) = 2e^{x+\psi(m/2)} f_{\chi^2_m}\left(2e^{x+\psi(m/2)}\right) \tag{6.15}$$

where $f_{\chi^2_m}(x) = \frac{1}{2}\left(\frac{x}{2}\right)^{m/2-1} e^{-x/2}$, $x \geq 0$. Also, the second and third central moments are computed as:

$$\mathbb{E}\left[(\widehat{p}-p)^2\right] = \frac{2p^2}{m}, \quad \mathbb{E}\left[(\widehat{p}-p)^3\right] = \frac{8p^3}{m^2}, \tag{6.16}$$

and

$$\mathbb{E}\left[(\widehat{h}-h)^k\right] = \psi^{(k-1)}\left(\frac{m}{2}\right), \quad k = 2,3 \tag{6.17}$$

where $\psi^{(k-1)}$ is the $(k-1)$-th derivative of the digamma function. It is noted that the pdf of $\widetilde{h}$ does not depend on the true value h, while that of $\widetilde{p}$ depends on p. Figure 6-3 depicts the pdfs for both estimation errors for the case $m = 99$ and $p = 1$; it can be seen that for this value of p the second moment of $\widetilde{p}$ is almost same as that of $\widetilde{h}$ for any m (Figure 6-4). Since the variance of $\widetilde{p}$ is proportional to p, the shape of a pdf with a larger p will be more dispersed. In Figure 6-3 the shapes of Gaussian

[2]The variable $m = L_E - 1$ rather than L_E will be used in this chapter's analysis for notational convenience.

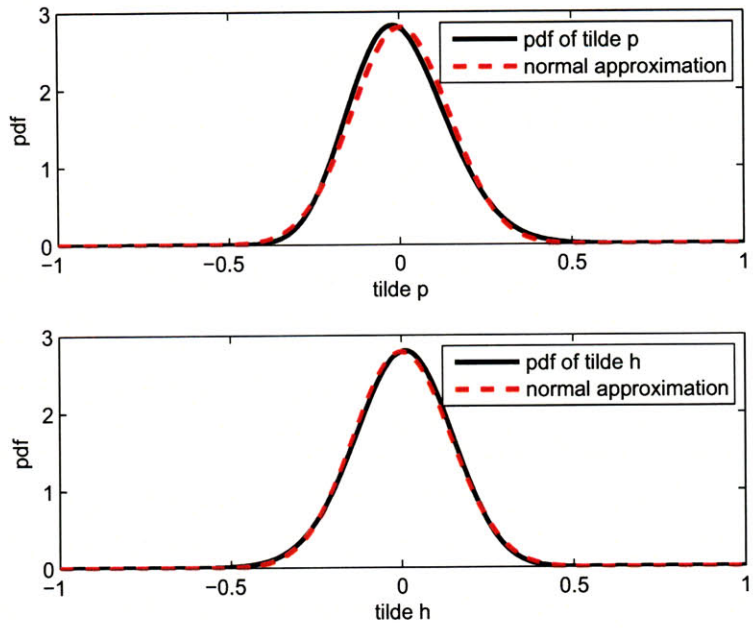

Figure 6-3: Probability density of estimation error ($m = 99$)

pdfs with same variances, which are $\mathbb{E}[\tilde{p}^2]_{p=1} = 0.1421^2$ and $\mathbb{E}[\tilde{h}^2] = 0.1429^2$, are also plotted for comparison. Also, it is observed that pdf of $\tilde{p}$ is positively skewed (right long tail) while that of $\tilde{h}$ is slightly negatively skewed. Since the skewness is relatively small for $\tilde{h}$ and it is reduced further for large m, the pdf of $\tilde{h}$ is well approximated as a Gaussian.

As Figure 6-4 shows the standard deviation of $\tilde{h}$ with respect to m that almost coincides that of $\tilde{p}$ in case $p = 1$, it is conceived that the estimation error of entropy estimation decreases on the order of $1/\sqrt{m}$ as m increases. For instance, in order to estimate p with estimation error standard deviation being less than 10% of the true value, more than 200 ensembles are needed. When regarding p itself as a *signal* not the variance of another signal, this can be interpreted as more than 200 samples are needed to have bigger than 20 dB SNR.

This analysis of the statistics of estimation error can be extended to a multivariate case. The same sort of unbiased estimator of $\log\det P$ for $P \succ 0 \in \mathbb{R}^{n\times n}$ is written as

$$\widehat{\mathcal{H}} = \log\det\left(\frac{m}{2^n}\widehat{P}\right) - \prod_{i=0}^{n-1}\psi\left(\frac{m-i}{2}\right) \tag{6.18}$$

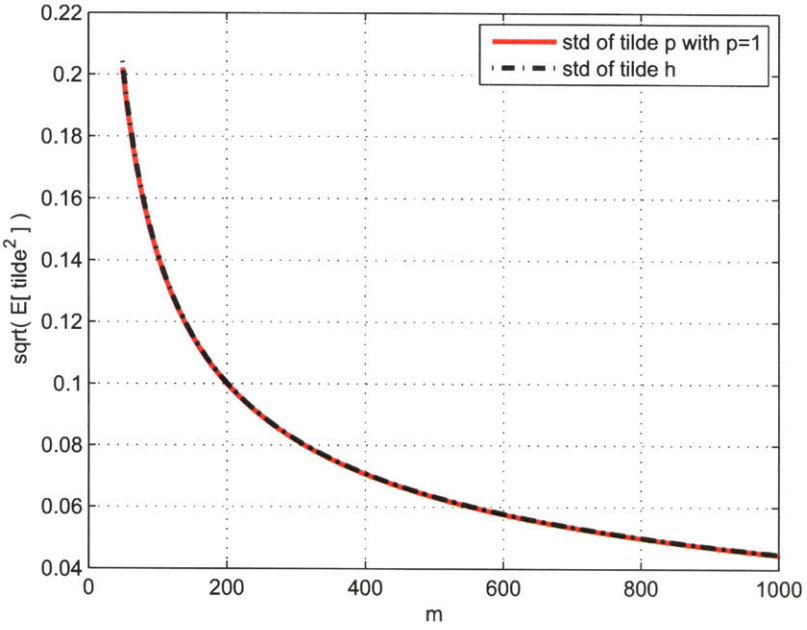

Figure 6-4: Standard deviation of $\widetilde{p}$ and $\widetilde{h}$

where $\widehat{P} \equiv \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}'/m$ is the minimum-variance unbiased estimator of P [49]. Then, the estimation error variance becomes

$$\mathbb{E}\left[\left(\widehat{\mathcal{H}} - \mathcal{H}\right)^2\right] = \sum_{i=0}^{n-1} \psi^{(1)}\left(\frac{m-i}{2}\right), \tag{6.19}$$

which depends on the dimension of the random vector n, the sample size $m+1$, but not the true value of $\mathcal{H}$. Figure 6-5 depicts the error standard deviation for various values of n and m. The plots shows that large n and small m lead to large estimation error (on the order of $\sqrt{n/m}$). This dependency of error standard deviation on the order of $\sqrt{n/m}$ will be utilized to figure out the impact of limited ensemble size on the performance of the ensemble-based targeting, in the following section.

6.3.2 Range-to-Noise Ratio

The ensemble-based targeting problem must distinguish the best measurement candidate from other suboptimal measurement candidates. An important predictor of the degree of impact that limitation of sample size might have on the solution optimality

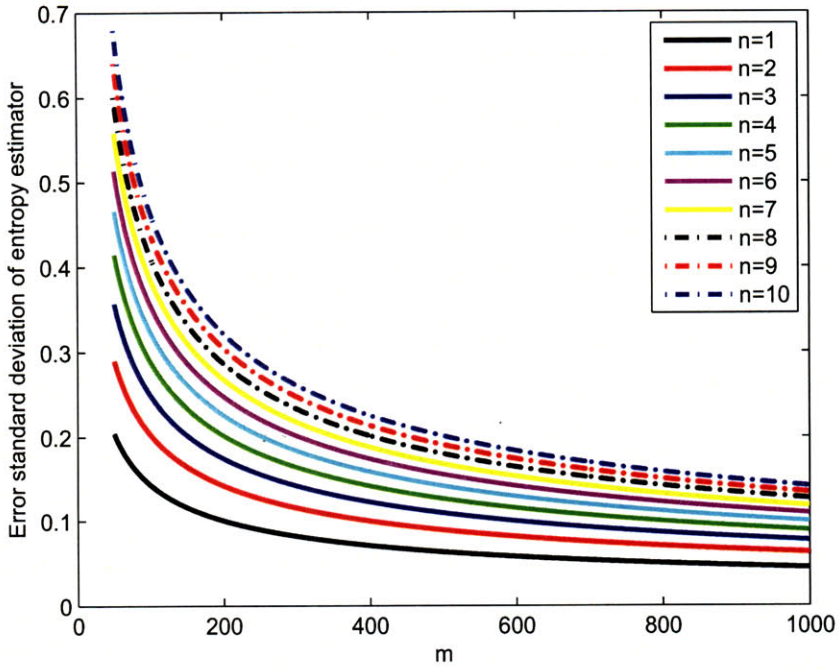

Figure 6-5: Standard deviation of $\widehat{\mathcal{H}} - \mathcal{H}$

is the *range-to-noise ratio* of the mutual information values:

$$\mathrm{RNR} = \frac{\sup_{\mathbf{s}} \mathcal{I}(\mathbf{s}) - \inf_{\mathbf{s}} \mathcal{I}(\mathbf{s})}{\sup_{\mathbf{s}} \sqrt{\mathbb{E}\left[\left(\widehat{\mathcal{I}}(\mathbf{s}) - \mathcal{I}(\mathbf{s})\right)^2\right]}} \tag{6.20}$$

where $\widehat{\mathcal{I}}(\mathbf{s})$ and $\mathcal{I}(\mathbf{s})$ denote the predicted and the actual mutual information value for $Z_{\mathbf{s}}$. Utilizing the statistics of the entropy estimation described in the previous section, the sup value in the denominator can be obtained without regard to the true values of $\mathcal{I}(\mathbf{s})$. Estimation of mutual information can be treated as estimation of the two – prior and posterior – entropies and substraction of those two. Since the bias term $\prod_{i=0}^{n-1} \psi\left(\frac{m-i}{2}\right)$ is the same for the prior and posterior entropy estimation, the

estimation error of the mutual information can be expressed as

$$\begin{aligned}
&\mathbb{E}\left\{\left[\frac{1}{2}(\widehat{\mathcal{H}}^- - \widehat{\mathcal{H}}^+) - \frac{1}{2}(\mathcal{H}^- - \mathcal{H}^+)\right]^2\right\} \\
&= \frac{1}{4}\mathbb{E}\left[(\widehat{\mathcal{H}}^- - \mathcal{H}^-)^2\right] + \frac{1}{4}\mathbb{E}\left[(\widehat{\mathcal{H}}^+ - \mathcal{H}^+)^2\right] \\
&\quad - \frac{1}{2}\mathbb{E}\left[(\widehat{\mathcal{H}}^- - \mathcal{H}^-)(\widehat{\mathcal{H}}^+ - \mathcal{H}^+)\right] \\
&= \frac{1}{2}\sum_{i=0}^{n-1}\psi^{(1)}\left(\frac{m-i}{2}\right) - \frac{1}{2}\mathbb{E}\left[(\widehat{\mathcal{H}}^- - \mathcal{H}^-)(\widehat{\mathcal{H}}^+ - \mathcal{H}^+)\right].
\end{aligned} \tag{6.21}$$

where superscripts '-' and '+' denote the prior and posterior. It can be shown that the cross correlation term in the final expression is always non-negative; so the estimation error of the mutual information is upper-bounded by

$$\mathbb{E}\left[\left(\widehat{\mathcal{I}}(\mathbf{s}) - \mathcal{I}(\mathbf{s})\right)^2\right] \leq \frac{1}{2}\sum_{i=0}^{n-1}\psi^{(1)}\left(\frac{m-i}{2}\right) \triangleq \sigma_{m,n}^2 \tag{6.22}$$

where equality holds if the prior and posterior entropy estimators are uncorrelated, which corresponds to infinite mutual information. With this upper bound, the RNR can be approximated as

$$\mathrm{RNR} \approx \frac{\sup_{\mathbf{s}}\mathcal{I}(\mathbf{s}) - \inf_{\mathbf{s}}\mathcal{I}(\mathbf{s})}{\sigma_{m,n}}. \tag{6.23}$$

In contrast to the denominator, the numerator of (6.23) is problem-dependent. Moreover, the sup and inf values cannot be known unless the true covariances (and equivalently true entropies) are known. Regarding the inf value, note that $\mathcal{I}(\mathbf{s})$ is lower-bounded by zero; therefore, it is reasonable to say that $\inf_{\mathbf{s}}\mathcal{I}(\mathbf{s})$ is a very small positive quantity. This suggests that we can approximate the inf value in the numerator of (6.23) as zero. With regard to the sup value, since the 95%-confident interval estimator of $\mathcal{I}(\mathbf{s})$ is $\widehat{\mathcal{I}}(\mathbf{s}) \pm 2\sigma_{m,n}$, the interval estimate for RNR is

$$\widehat{\mathrm{RNR}} = \left[\max\left\{0, \frac{\sup_{\mathbf{s}}\widehat{\mathcal{I}}(\mathbf{s})}{\sigma_{m,n}} - 2\right\}, \frac{\sup_{\mathbf{s}}\widehat{\mathcal{I}}(\mathbf{s})}{\sigma_{m,n}} + 2\right] \tag{6.24}$$

with confidence level 95%. The max function that RNR is positive. If the objective

of computing RNR is to predict whether or not a *small* RNR would cause significant performance degradation of the targeting, the following one-sided interval estimator can also be used:

$$\overline{\text{RNR}} = \left[0, \frac{\sup_{\mathbf{s}} \widehat{\mathcal{I}}(\mathbf{s})}{\sigma_{m,n}} + 1.7\right] \tag{6.25}$$

with 95% confidence level.

6.3.3 Probability of Correct Decision

This section considers the probability that the ensemble-based targeting provides the true optimal (or $(1-\epsilon)$-optimal) solution, which is referred to as *probability of correct decision* (PCD) hereafter, for given values of RNR, m, n, and the total number of candidates q. To do this, the following are assumed:

1. There are a total of q measurement candidates denoted as $\mathbf{s}_1, \ldots, \mathbf{s}_q$. Without loss of generality, $\mathbf{s}_i$ corresponds to the i-th best targeting solution.

2. The true mutual information values are uniformly distributed over the corresponding range $\mathcal{I}(\mathbf{s}_1) - \mathcal{I}(\mathbf{s}_q)$. In other words,

$$\mathcal{I}(\mathbf{s}_i) = \mathcal{I}(\mathbf{s}_1) - (i-1)\delta \tag{6.26}$$

   where

$$\delta = \frac{\mathcal{I}(\mathbf{s}_1) - \mathcal{I}(\mathbf{s}_q)}{q-1} = \text{RNR} \cdot \frac{\sigma_{m,n}}{q-1}. \tag{6.27}$$

3. The estimation error of each mutual information value is distributed with $\mathcal{N}(0, \sigma_{m,n}^2)$.

4. The estimation errors of the mutual information for each measurement candidate are uncorrelated each other. In other words,

$$\mathbb{E}\left[(\widehat{\mathcal{I}}(\mathbf{s}_i) - \mathcal{I}(\mathbf{s}_i))(\widehat{\mathcal{I}}(\mathbf{s}_j) - \mathcal{I}(\mathbf{s}_j))\right] = 0, \ \forall i \neq j. \tag{6.28}$$

Under these assumptions, it can be shown that for $i \leq q-1$

$$D_i \triangleq \widehat{\mathcal{I}}(\mathbf{s}_1) - \widehat{\mathcal{I}}(\mathbf{s}_i) \sim \mathcal{N}\left((i-1)\delta, 2\sigma_{m,n}^2\right) \tag{6.29}$$

and

$$\mathbf{P}(D_i, D_j) = \mathbb{E}[(D_i - (i-1)\delta)(D_j - (j-1)\delta)] = \sigma_{m,n}^2, \quad \forall i \neq j. \tag{6.30}$$

Given that the PCD can de interpreted as the probability that the ensemble-based targeting declares $\mathbf{s}_1$ to be the best candidate, the PCD can be written in terms of D_i's as

$$\text{PCD} = \texttt{Prob}\left[D_i > 0, \ \forall i\right]. \tag{6.31}$$

Using (6.29) and (6.30), the PCD can be computed as

$$\begin{aligned} \text{PCD} = \int_{-\infty}^{\text{RNR}} \cdots \int_{-\infty}^{\frac{i}{q-1}\text{RNR}} \cdots \int_{-\infty}^{\frac{\text{RNR}}{q-1}} \\ f_{\mathcal{N}(0,\Sigma)}(x_1, \cdots, x_{q-1}) dx_1 \cdots dx_{q-1} \end{aligned} \tag{6.32}$$

where $f_{\mathcal{N}(0,\Sigma)}$ is the pdf of the zero-mean multivariate Gaussian distribution with the covariance matrix of

$$\Sigma = I_{q-1} + \mathbf{1}_{q-1} \otimes \mathbf{1}'_{q-1}. \tag{6.33}$$

where I_{q-1} denotes the $(q-1) \times (q-1)$ identity matrix, $\mathbf{1}_{q-1}$ is the $(q-1)$-dimension column vector with every element being unity, and $\otimes$ denotes the Kronecker product. That is, all the diagonal elements of Σ_q are 2, while all the off-diagonal elements are one.

Note that the PCD is expressed as a cdf of a $(q-1)$-dimensional normal distribution. In the special case of $q = 2$,

$$\text{PCD}_{q=2} = \Phi\left(\frac{\text{RNR}}{\sqrt{2}}\right) \tag{6.34}$$

where $\Phi(\cdot)$ is the cdf of the standard normal distribution. For the case with $q > 2$, eigenvalue decomposition of the inverse of Σ in (6.33) leads to PCD being a product of univariate normal cdfs, and there exists an efficient numerical algorithm based on Cholesky factorization [50].

Figure 6-6 shows how PCD changes with q and RNR. the plots shows that PCD is monotonically increasing with respect to RNR, while it is monotonically decreasing with respect to q. The dependency of PCD on q is crucial, since q is a very large number in practice – recall that $q = \binom{N}{n}$. Thus, PCD can be meaninglessly small for a large-scale selection problem. In addition, to calculate PCD for such large q is computationally very expensive, because it requires a cdf evaluation of a large-dimensional normal distribution.

For this reason, for a large q case, this work suggests to utilize the probability of ϵ-correct decision (ϵ-PCD) defined as

$$\epsilon\text{-PCD} = \texttt{Prob}\left[\bigcup_{i=1}^{\lfloor \epsilon q \rfloor} \left(\widehat{\mathcal{I}}(\mathbf{s}_i) > \widehat{\mathcal{I}}(\mathbf{s}_j),\ \forall j \neq i\right)\right], \tag{6.35}$$

since it can still be used as an indicator of the impact of limited sample size on the degree of optimality, and also it can be computed can be computed tractably.

By the symmetry of the distribution of the true mutual information values, the lower bound of this ϵ-PCD can be computed by

$$\text{PCD}_{\lfloor 1/\epsilon \rfloor} \leq \epsilon\text{-PCD}, \tag{6.36}$$

where equality holds if $\lfloor \epsilon q \rfloor$ and $\lfloor 1/\epsilon \rfloor$ are integers. In other words, if dividing q candidates into $1/\epsilon$ groups such that the i-th group consists of $\mathbf{s}_{(i-1)\epsilon q+1}$ through $\mathbf{s}_{i\epsilon q}$, then the decision of declaring one of the candidates in the first group to be the optimal solution is equivalent to the decision of distinguishing the best candidate out of $1/\epsilon$ candidates.

With this, Figure 6-6 can be used to interpret the relation between RNR and $\lceil q \rceil$-PCD. For instance, the graph of PCD for $q = 10$ represents the relation between

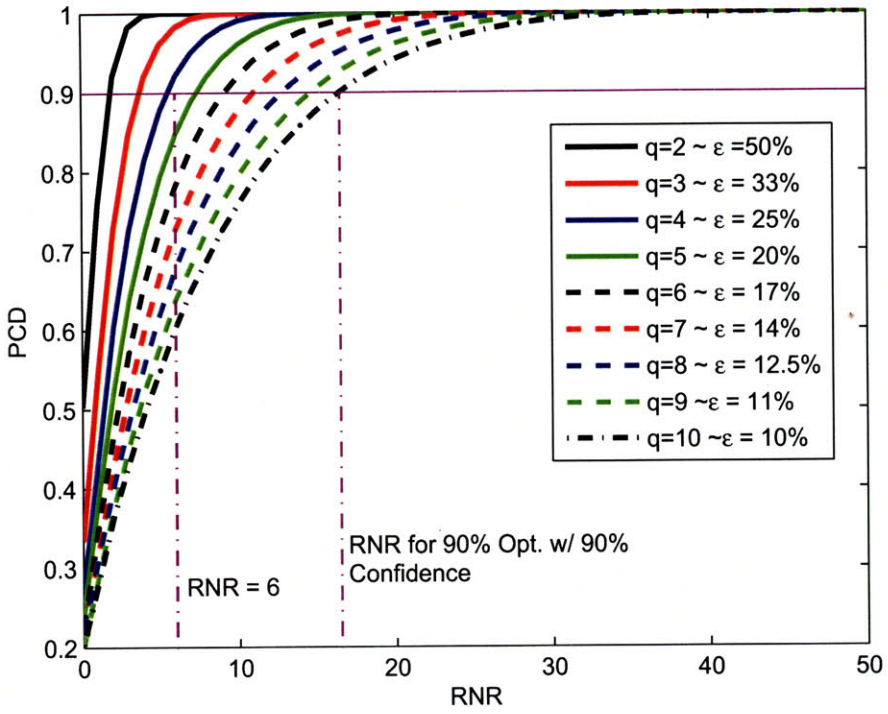

Figure 6-6: Probability of Correct Decision for targeting with q candidates

RNR and 10%-PCD for any size of targeting problem. In the picture, the dotted line indicates the RNR value above which 10%-PCD is greater than 90%, which is 16.5. This is interpreted as *in order to have a 90%-optimal targeting solution with the confidence level of 90%, RNR should be greater than 16.5.* In terms of the one-sided interval estimator of RNR in (6.25), this implies that

$$\sup_{\mathbf{s}} \widehat{\mathcal{I}}(\mathbf{s}) > 18.2\sigma_{m,n} \approx 18.2\sqrt{n/m} \tag{6.37}$$

for the same qualification with 95% confidence level. The last approximate expression comes from $\sigma_{m,n} \approx \sqrt{n/m}$ for a small n. In addition, the relation in Figure 6-6 can be utilized to figure out the level of performance for a given RNR. For instance, as the curve for $q = 4$ crosses the horizontal line of PCD=0.9 when the corresponding RNR is smaller than 6, it can be inferred that RNR=6 guarantees 75% optimality with 90% confidence.

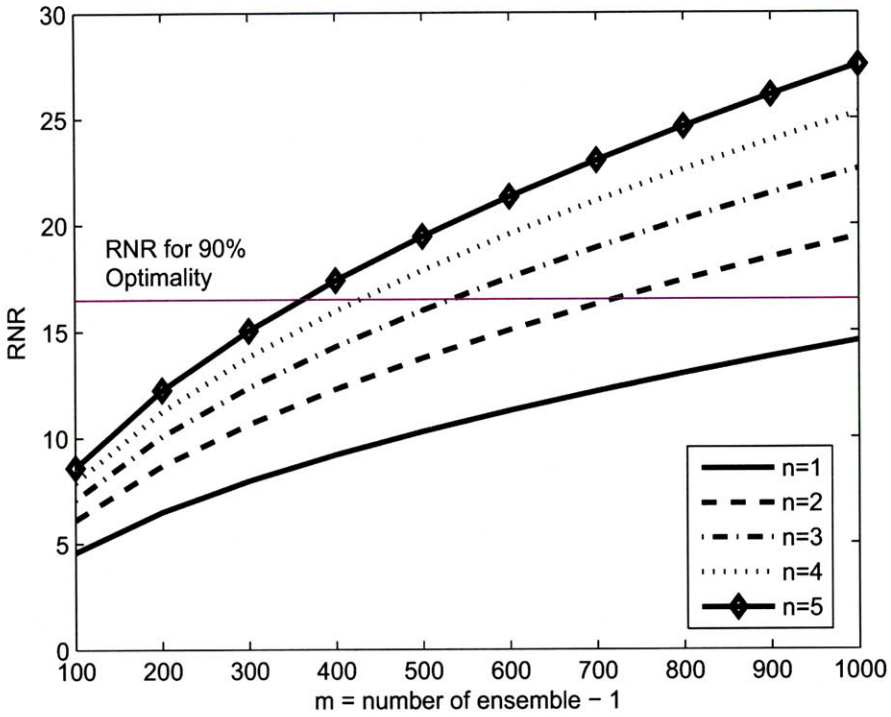

Figure 6-7: Range-to-noise ratio for Lorenz-95 example for various m and n

6.3.4 Confirmation of Analysis with Lorenz-95 Targeting Example

The results of Figure 6-1 showed a nontrivial performance degradation for the targeting example using the Lorenz-95 model. This section will verify that that performance degradation can be predicted in terms of RNR and ϵ-PCD. The RNR expression in (6.23) is considered under the assumption that $L_E^0 = 1024$ ensembles are sufficiently large to correctly estimate $\sup_{\mathbf{s}} \mathcal{I}(\mathbf{s})$ and $\inf_{\mathbf{s}} \mathcal{I}(\mathbf{s})$.

Figure 6-7 depicts the RNR values for $m \in [100, 1000]$ and for $n \in [1, 5]$. Note that for a given n, RNR decreases as m increases, while for a given m, it increases as n increases. For $n = 1$, the requirement of RNR>16.5 that achieves 90%-optimality with 90% confidence is not satisfied even with $m = 1000$, while $m = 400$ meets the same requirement for $n = 5$. Dependency of RNR on m is simply reflecting the fact that $\sigma_{m,n}$ is an increasing function of m for fixed n. The increasing tendency of RNR with respect to n is caused by the fact that the optimal mutual information value

grows faster than $\mathcal{O}(\sqrt{n})$. Since

$$\text{RNR} \approx \frac{\sup_{\mathbf{s}} \mathcal{I}(\mathbf{s})}{\sqrt{n/m}} \tag{6.38}$$

for a small n, RNR becomes an increasing function of n if the sup value in the numerator grows faster than $\mathcal{O}(\sqrt{n})$, which is the case for the Lorenz-95 example in this work. Also, seeing that the marginal increment of RNR diminishes as n increases, it is conceivable that there exists a threshold $\bar{n}$ over which increasing n no more improves RNR.

The Monte-Carlo simulation for Figure 6-1 considered $n = 2$; it can be seen that the size of error bar becomes smaller than 10% of the optimal solution value when the ensemble size is larger than 700. This result is consistent with Figure 6-7 which indicates that more than 750 ensembles are needed for 90%-optimality. Also, if $m = 100$ and $n = 2$, Figure 6-7 shows that RNR=6.1; in Figure 6-6, it can be found that the graph that crosses the dotted line of PCD=0.9 at RNR=6 is the one for $q = 4$. Since $\text{PCD}_{q=4} = 25\%$-PCD, it is conceived that the ensemble targeting with $m = 100$ will provide 75%-optimal solutions with 90% confidence, which is consistent with the result in Figure 6-1.

6.4 Conclusions

This chapter quantified the impact of the limited ensemble size on the performance variation of the targeting solutions in both experimental and theoretical ways. The concepts of range-to-noise ratio and probability of correct decision were proposed to derive mathematical relations between the ensemble size and the solution sensitivity, which were verified to be coherent with the numerical results.

Chapter 7

Conclusions

7.1 Contributions

This thesis developed information-theoretic frameworks for efficient planning of mobile sensor networks in the context of adaptive sampling for numerical weather prediction. While addressing key challenges of this decision making, this thesis has also made important technological contributions to the sensor networks and robotics communities, as outlined below.

In the area of sensor networks, this thesis has:

- Clearly showed the potential improvements in computational efficiency provided by the backward formulation of the mutual information that is achieved by reducing the number of covariance updates in the combinatorial sensor selection problem.
  - Presented computation time analysis to prove that the backward approach works significantly faster than the forward approach for the ensemble-based representation, and that it is never slower than the forward one, even for the conventional covariance representation.
  - Identified the types of systems for which the backward method provides a substantial computational improvement over the forward by contrasting the adaptive sampling problem to a smaller-scale sensor management

problem.

- Demonstrated the importance of coordinated update of the covariance information to achieve good overall targeting performance over the sensor networks.

- Presented a novel smoother form of the mutual information in the continuous-time domain that projects the decision space from the long forecast horizon to the short planning horizon.

  - Identified the advantages of the smoother form over a filter form in terms of computational efficiency, robustness to modeling error, easiness in extension of existing path planning techniques, and capability of concurrent tracking of accumulated information.

  - Clearly demonstrated that the smoother form quantifies the pure impact of measurement on the uncertainty reduction of the verification variables, by proving that the rate of smoother-form mutual information is non-negative regardless of the process noise.

In the area of robotics, this thesis has:

- Developed algorithms for multi-sensor platform targeting problems that efficiently incorporate the vehicle mobility constraints into the backward selection framework.

- Presented a framework of continuous motion planning of sensor platforms that combines the quantification tool of the continuous-time information reward and the representation technique of continuous paths using spatial interpolation methods.

  - Presented the information potential field for the smoother form expression to be used in developing a real-time steering law and in visualizing the information-rich regions.

In the area of numerical weather prediction, this thesis has:

- Introduced a information-theoretic methodology to adaptive sampling problems that can better account for the correlation amongst verification variables and exploit the properties of commutativity and conditional independence to improve computational efficiency.

- Characterized important sensitivity aspects of the targeting solution to the limited ensemble diversity, with specifically indicating similar features in traditional targeting schemes.

  - Proposed theoretical tools to relate the ensemble size and potential performance variation in the targeting solutions.

7.2 Future Work

- Future research will implement the proposed planning methods to adaptive sampling problems for a realistic weather model. Specifically, integration of the presented technology into the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS) [51] is on-going.

- Given inherent limitation in the ensemble diversity for a large-scale systems, a mechanism that provides robustness to the ensemble variation will an be important topic for future research.

- Decentralized and distributed implementation of the presented methods is another venue of interest. This will include incorporation of more details of the communication models.

Appendix A

Analysis of Submodularity

A.1 Conditions for Submodularity

For the discrete selection problem with a *submodular* reward function, the sequential greedy strategy guarantees that

$$\frac{\mathbf{SGRD}}{\mathbf{OPT}} \geq 1 - (1 - 1/n)^n > 1 - 1/e \tag{A.1}$$

where **SGRD** and **OPT** denote the objective value for the sequential greedy solution and the optimal solution. It was shown in [6] that the entropy is a submodular reward function, while the mutual information is not in general. Thus, FSGS (and equivalently, BSGS) in section 3.3 does not guarantee $(1 - 1/e) \cdot \mathbf{OPT}$ performance in the sensor targeting problem with objective function of $\mathcal{I}(V; Z_{\mathbf{s}})$ in general:

$$\mathcal{I}(V; Z_{\mathbf{s}^{FG\star}}) \lesseqgtr (1 - 1/e) \times \mathcal{I}(V; Z_{\mathbf{s}_F^\star}). \tag{A.2}$$

where $\lesseqgtr$ means either can happen depending on the problem. Nevertheless, this section will analyze the conditions where the information reward for the targeting problem becomes submodular, as this analysis provides good insights on the relationship amongst the mutual information, correlation coefficients, and sensing noise variance.

Submodularity means the diminishing marginal gain of a set function. With this work's information reward, submodularity is equivalent to the following statement:

$$\mathcal{I}(V; Z_s|Z_{\mathbf{s}_0}) - \mathcal{I}(V; Z_s|Z_q, Z_{\mathbf{s}_0}) \geq 0, \quad \forall s, q \in \mathcal{S}, \ \forall \mathbf{s}_0 \subset \mathcal{S}. \tag{A.3}$$

In other words, submodularity means for measurement selection Z_s, its contribution to the entropy reduction of V diminishes by another measurement selection Z_q, conditioned on any previous selection $Z_{\mathbf{s}_0}$. From this point on, for convenience, the conditioning on previous selection $Z_{\mathbf{s}_0}$ will be presumed; thus, "prior" or "unconditioned" means already being conditioned on $Z_{\mathbf{s}_0}$.

The left-hand side of (A.3) can be manipulated as

$$\begin{aligned}
&\mathcal{I}(V; Z_s|Z_{\mathbf{s}_0}) - \mathcal{I}(V; Z_s|Z_q, Z_{\mathbf{s}_0}) \\
&= \mathcal{I}(Z_s; V|Z_{\mathbf{s}_0}) - \mathcal{I}(Z_s; V|Z_q, Z_{\mathbf{s}_0}) \\
&= \mathcal{H}(Z_s|Z_{\mathbf{s}_0}) - \mathcal{H}(Z_s|V, Z_{\mathbf{s}_0}) - \mathcal{H}(Z_s|Z_q, Z_{\mathbf{s}_0}) + \mathcal{H}(Z_s|Z_q, V, Z_{\mathbf{s}_0}) \\
&= [\mathcal{H}(Z_s|Z_{\mathbf{s}_0}) - \mathcal{H}(Z_s|Z_q, Z_{\mathbf{s}_0})] - [\mathcal{H}(Z_s|V, Z_{\mathbf{s}_0}) - \mathcal{H}(Z_s|Z_q, V, Z_{\mathbf{s}_0})] \\
&= \mathcal{I}(Z_s; Z_q|Z_{\mathbf{s}_0}) - \mathcal{I}(Z_s; Z_q|V, Z_{\mathbf{s}_0}).
\end{aligned} \tag{A.4}$$

Namely, the difference between the contribution of Z_s to entropy reduction of V without and with another measurement Z_q is the same as the difference between the unconditioned and the conditioned mutual information between Z_s and Z_q where conditioning is on V. Thus, the information reward is submodular if the prior mutual information between any two measurement selections is larger than the posterior mutual information between them; therefore, submodularity is equivalent to:

$$\mathcal{I}(Z_s; Z_q|Z_{\mathbf{s}_0}) - \mathcal{I}(Z_s; Z_q|V, Z_{\mathbf{s}_0}) \geq 0, \quad \forall s, q \in \mathcal{S}, \ \ \forall \mathbf{s}_0 \subset \mathcal{S}. \tag{A.5}$$

One representative case that satisfies (A.5) is when $V = X_{\mathcal{S}}$, as then

$$\begin{aligned}\mathcal{I}(Z_s; Z_q | X_{\mathcal{S}}, Z_{\mathbf{s}_0}) &= \mathcal{H}(Z_s | X_{\mathcal{S}}, Z_{\mathbf{s}_0}) - \mathcal{H}(Z_s | X_{\mathcal{S}}, Z_q, Z_{\mathbf{s}_0}) \\ &= \mathcal{H}(Z_s | X_{\mathcal{S}}, Z_{\mathbf{s}_0}) - \mathcal{H}(Z_s | X_{\mathcal{S}}, Z_{\mathbf{s}_0}) \\ &= 0,\end{aligned} \tag{A.6}$$

because Z_s and Z_q are conditionally independent of each other given $X_{\mathcal{S}}$. The prior mutual information $\mathcal{I}(Z_s; Z_q | Z_{\mathbf{s}_0}) \geq 0$, because any mutual information value is non-negative [35]. Thus, if $V = X_{\mathcal{S}}$ mutual information is a submodular metric. Note that the sensor management problem in section 3.1.2 belongs to this case; therefore, the information reward is submodular for the sensor management decision and the greedy strategy guarantees $(1 - 1/e)$ optimality.

If $V \neq X_{\mathcal{S}}$, the posterior mutual information $\mathcal{I}(Z_s; Z_q | V, Z_{\mathbf{s}_0})$ can be greater than the prior one $\mathcal{I}(Z_s; Z_q, Z_{\mathbf{s}_0})$; there is no general theory that guarantees the minimum performance of the sequential greedy strategy. However, further analysis can relate the submodularity to the correlation coefficient amongst the measurement choices:

Suppose that the prior covariance of $X_s \cup X_q \cup V$ is expressed as

$$\begin{aligned}\mathbf{P}(X_s \cup X_q \cup V | Z_{\mathbf{s}_0}) &= \begin{bmatrix} \mathbf{P}(X_s | Z_{\mathbf{s}_0}) & \mathbf{P}(X_s, X_q | Z_{\mathbf{s}_0}) & \mathbf{P}(X_s, V | Z_{\mathbf{s}_0}) \\ \mathbf{P}(X_q, X_s | Z_{\mathbf{s}_0}) & \mathbf{P}(X_q | Z_{\mathbf{s}_0}) & \mathbf{P}(X_q, V | Z_{\mathbf{s}_0}) \\ \mathbf{P}(V, X_s | Z_{\mathbf{s}_0}) & \mathbf{P}(V, X_q | Z_{\mathbf{s}_0}) & \mathbf{P}(V | Z_{\mathbf{s}_0}) \end{bmatrix} \\ &\triangleq \begin{bmatrix} P_s & P_{s,q} & P_{s,V} \\ P_{q,s} & P_q & P_{q,V} \\ P_{V,s} & P_{V,q} & P_V \end{bmatrix}.\end{aligned} \tag{A.7}$$

Since the measurement noise at location s and location q are independent each other, the covariance of $X_s \cup X_q \cup V$ becomes

$$\mathbf{P}(Z_s \cup Z_q \cup V | Z_{\mathbf{s}_0}) = \begin{bmatrix} P_s + R_s & P_{s,q} & P_{s,V} \\ P_{q,s} & P_q + R_q & P_{q,V} \\ P_{V,s} & P_{V,q} & P_V \end{bmatrix}. \tag{A.8}$$

The unconditioned mutual information $\mathcal{I}(Z_s; Z_q|Z_{\mathbf{s}_0})$ can be expressed in terms of the elements in (A.8) as

$$\begin{aligned}
\mathcal{I}(Z_s; Z_q|Z_{\mathbf{s}_0}) &= \mathcal{H}(Z_s|Z_{\mathbf{s}_0}) - \mathcal{H}(Z_s|Z_q, Z_{\mathbf{s}_0}) \\
&= \mathcal{H}(Z_s|Z_{\mathbf{s}_0}) + \mathcal{H}(Z_q|Z_{\mathbf{s}_0}) - \mathcal{H}(Z_s, Z_q|Z_{\mathbf{s}_0}) \\
&= \log\det\begin{bmatrix} P_s + R_s & 0 \\ 0 & P_q + R_q \end{bmatrix} - \log\det\begin{bmatrix} P_s + R_s & P_{s,q} \\ P_{q,s} & P_q + R_q \end{bmatrix} \\
&= -\log\det\left(\begin{bmatrix} P_s + R_s & P_{s,q} \\ P_{q,s} & P_q + R_q \end{bmatrix}\begin{bmatrix} (P_s + R_s)^{-1} & 0 \\ 0 & (P_q + R_q)^{-1} \end{bmatrix}\right) \\
&= -\log\det\begin{bmatrix} I & P_{s,q}(P_q + R_q)^{-1} \\ P_{q,s}(P_s + R_s)^{-1} & (P_q + R_q)^{-1} \end{bmatrix} \\
&= -\log\det\left(I - P_{s,q}(P_q + R_q)^{-1}P_{q,s}(P_s + R_s)^{-1}\right).
\end{aligned} \tag{A.9}$$

As $|s| = |q| = 1$ (although the derivation in (A.9) holds for other cases as well), the mutual information can be expressed as

$$\mathcal{I}(Z_s; Z_q|Z_{\mathbf{s}_0}) = -\log\left[1 - \rho(Z_s, Z_q|Z_{\mathbf{s}_0})^2\right] \tag{A.10}$$

$$= -\log\left[1 - \rho(X_s, X_q|Z_{\mathbf{s}_0})^2 \times (1 + R_s/P_s)^{-1}(1 + R_q/P_q)^{-1}\right] \tag{A.11}$$

where $\rho(A_1, A_2|A_3)$ denotes the conditional correlation coefficient between two random variables A_1 and A_2 conditioned on A_3. Since $-\log(1 - \rho^2)$ is a monotonically increasing function of ρ, the mutual information between Z_s and Z_q equivalently means the degree of correlation between them. In terms of the state variables X_s and Z_q, the mutual information $\mathcal{I}(Z_s; Z_q|Z_{\mathbf{s}_0})$ represents the degree of correlation with some scaling factor that takes into account the sensing noise level. In addition, it is noted that $\mathcal{I}(Z_s; Z_q|Z_{\mathbf{s}_0})$ is a decreasing function of measurement noise R_s and R_q,

since

$$\frac{\partial \mathcal{I}(Z_s; Z_q|Z_{\mathbf{s}_0})}{\partial R_s} = -\frac{\rho(Z_s, Z_q|Z_{\mathbf{s}_0})^2}{(P_s + R_s)(1 - \rho(Z_s, Z_q|Z_{\mathbf{s}_0})^2)} = -\frac{e^{\mathcal{I}(Z_s;Z_q|Z_{\mathbf{s}_0})} - 1}{P_s + R_s} \leq 0 \quad \text{(A.12)}$$

$$\frac{\partial \mathcal{I}(Z_s; Z_q|Z_{\mathbf{s}_0})}{\partial R_q} = -\frac{\rho(Z_s, Z_q|Z_{\mathbf{s}_0})^2}{(P_q + R_q)(1 - \rho(Z_s, Z_q)^2)} = -\frac{e^{\mathcal{I}(Z_s;Z_q|Z_{\mathbf{s}_0})} - 1}{P_q + R_q} \leq 0. \quad \text{(A.13)}$$

Specifically, if $R_S = R_q = 0$, $\mathcal{I}(Z_s; Z_q|Z_{\mathbf{s}_0})$ is equivalent to the correlation of X_s and X_q, while if $R_s = R_q = \infty$, the mutual information becomes zero regardless of $\rho(X_s, X_q|Z_{\mathbf{s}_0})$

In a similar manner, the conditioned mutual information $\mathcal{I}(Z_s; Z_q; V)$ can be expressed as

$$\mathcal{I}(Z_{\mathbf{s}_1}; Z_{\mathbf{s}_2}|V, Z_{\mathbf{s}_0}) = -\log\det\left(I - P_{s,q|V}P_{q|V}^{-1}P_{q,s|V}P_s^{-1}\right) \quad \text{(A.14)}$$

$$= -\log\left[1 - \rho(Z_s, Z_q|V, Z_{\mathbf{s}_0})^2\right] \quad \text{(A.15)}$$

$$= -\log\left[1 - \rho(X_s, X_q|V)^2 \times (1 + R_s/P_{s|V})^{-1}(1 + R_q/P_{q|V})^{-1}\right] \quad \text{(A.16)}$$

where the conditional covariance elements are related to the unconditioned ones as

$$P_{s|V} = P_s - P_{s,V}P_V^{-1}P_{V,s} \quad \text{(A.17)}$$

$$P_{q|V} = P_q - P_{q,V}P_V^{-1}P_{V,q} \quad \text{(A.18)}$$

$$P_{s,q|V} = P_{s,q} - P_{s,V}P_V^{-1}P_{V,q} \quad \text{(A.19)}$$

$$P_{q,s|V} = P_{q,s} - P_{q,V}P_V^{-1}P_{V,s}. \quad \text{(A.20)}$$

Thus, for submodularity, the following should be satisfied:

$$\mathcal{I}(Z_s; Z_q|Z_{\mathbf{s}_0}) - \mathcal{I}(Z_s; Z_q|V, Z_{\mathbf{s}_0})$$
$$= -\log\left[1 - \rho(Z_s, Z_q|Z_{\mathbf{s}_0})^2\right] + \log\left[1 - \rho(Z_s, Z_q|V, Z_{\mathbf{s}_0})^2\right] \geq 0, \ \forall s, q \in \mathcal{S}, \quad \text{(A.21)}$$

which is equivalent to:

Proposition 8. The information reward $\mathcal{I}(V; Z_{\mathbf{s}})$ is a submodular set function of the measurement choice $\mathbf{s}$, if

$$\rho(Z_s, Z_q|Z_{\mathbf{s}_0})^2 \geq \rho(Z_s, Z_q|V, Z_{\mathbf{s}_0})^2, \ \forall s, q \in \mathcal{S}, \ \forall \mathbf{s}_0 \subset \mathcal{S}. \tag{A.22}$$

In other words, if knowledge of V always reduces the correlation amongst the observation in the search space $\mathcal{S}$, then the entropy reduction of V by measurement selection $\mathbf{s}$ is submodular. □

Think of the variability of the difference between the unconditioned and the conditioned mutual information, or $\mathcal{I}(Z_s; Z_q|Z_{\mathbf{s}_0}) - \mathcal{I}(Z_s; Z_q; V, Z_{\mathbf{s}_0})$ in terms of the sensing noise variance. For simplicity, assume that $R_s = R_q = R_0$. Then,

$$\begin{aligned}
&\frac{\partial}{\partial R_0}\Big(\mathcal{I}(Z_s; Z_q|Z_{\mathbf{s}_0}) - \mathcal{I}(Z_s; Z_q|V, Z_{\mathbf{s}_0})\Big) \\
&= -[e^{\mathcal{I}(Z_s;Z_q|Z_{\mathbf{s}_0})} - 1]\left[(P_s + R_0)^{-1} + (P_q + R_0)^{-1}\right] \\
&\quad + [e^{\mathcal{I}(Z_s;Z_q|V,Z_{\mathbf{s}_0})} - 1]\left[(P_{s|V} + R_0)^{-1} + (P_{q|V} + R_0)^{-1}\right].
\end{aligned} \tag{A.23}$$

The mutual information and covariance elements are related as

$$\mathcal{I}(V; Z_s|Z_{\mathbf{s}_0}) = \frac{1}{2}\log\frac{P_s + R_0}{P_{s|V} + R_0}, \quad \mathcal{I}(V; Z_q|Z_{\mathbf{s}_0}) = \frac{1}{2}\log\frac{P_q + R_0}{P_{q|V} + R_0}, \tag{A.24}$$

which equivalently means

$$(P_{s|V} + R_0)^{-1} = (P_s + R_0)^{-1}e^{2\mathcal{I}(V;Z_s|Z_{\mathbf{s}_0})}, \quad (P_{q|V} + R_0)^{-1} = (P_q + R_0)^{-1}e^{2\mathcal{I}(V;Z_q|Z_{\mathbf{s}_0})}. \tag{A.25}$$

Thus, the derivative in (A.23) can be written as

$$\begin{aligned}
&\frac{\partial}{\partial R_0}\Big(\mathcal{I}(Z_s;Z_q|Z_{\mathbf{s}_0})-\mathcal{I}(Z_s;Z_q|V,Z_{\mathbf{s}_0})\Big)\\
&=(P_s+R_0)^{-1}\left\{e^{2\mathcal{I}(V;Z_s)}\left(e^{\mathcal{I}(Z_s;Z_q|V)}-1\right)-\left(e^{\mathcal{I}(Z_s;Z_q)}-1\right)\right\}\\
&\quad+(P_q+R_0)^{-1}\left\{e^{2\mathcal{I}(V;Z_q)}\left(e^{\mathcal{I}(Z_s;Z_q|V)}-1\right)-\left(e^{\mathcal{I}(Z_s;Z_q)}-1\right)\right\}\\
&=(P_s+R_0)^{-1}\left\{\left(e^{2\mathcal{I}(V;Z_s|Z_{\mathbf{s}_0})}-1\right)\left(e^{\mathcal{I}(Z_s;Z_q|V)}-1\right)-\left(e^{\mathcal{I}(Z_s;Z_q|Z_{\mathbf{s}_0})}-e^{\mathcal{I}(Z_s;Z_q|V)}\right)\right\}\\
&\quad+(P_q+R_0)^{-1}\left\{\left(e^{2\mathcal{I}(V;Z_q|Z_{\mathbf{s}_0})}-1\right)\left(e^{\mathcal{I}(Z_s;Z_q|V,Z_{\mathbf{s}_0})}-1\right)-\left(e^{\mathcal{I}(Z_s;Z_q|Z_{\mathbf{s}_0})}-e^{\mathcal{I}(Z_s;Z_q|V,Z_{\mathbf{s}_0})}\right)\right\}
\end{aligned}\tag{A.26}$$

Since every mutual information term is non-negative and diminishes as R_0 goes to infinity, the following observations can be made regarding the expression in (A.26):

1. $\frac{\partial}{\partial R_0}\Big(\mathcal{I}(Z_s;Z_q|Z_{\mathbf{s}_0})-\mathcal{I}(Z_s;Z_q|V,Z_{\mathbf{s}_0})\Big)\geq 0$, if $\mathcal{I}(Z_s;Z_q|Z_{\mathbf{s}_0})\leq\mathcal{I}(Z_s;Z_q|V,Z_{\mathbf{s}_0})$
2. $\frac{\partial}{\partial R_0}\Big(\mathcal{I}(Z_s;Z_q|Z_{\mathbf{s}_0})-\mathcal{I}(Z_s;Z_q|V,Z_{\mathbf{s}_0})\Big)\in[\alpha_1,\alpha_2]$, if $\mathcal{I}(Z_s;Z_q|Z_{\mathbf{s}_0})\geq\mathcal{I}(Z_s;Z_q|V,Z_{\mathbf{s}_0})$
3. $\lim_{R_0\to\infty}\frac{\partial}{\partial R_0}\Big(\mathcal{I}(Z_s;Z_q|Z_{\mathbf{s}_0})-\mathcal{I}(Z_s;Z_q|V,Z_{\mathbf{s}_0})\Big)=0$,

where $\alpha_1\leq 0$ and $\alpha_2\geq 0$ corresponds to the right-hand side of (A.26) obtained by replacing cases where $\mathcal{I}(V;Z_s|Z_{\mathbf{s}_0})$ and $\mathcal{I}(V;Z_q|Z_{\mathbf{s}_0})$ as zero, and where $\mathcal{I}(Z_s;Z_q|Z_{\mathbf{s}_0})$ by $\mathcal{I}(Z_s;Z_q|V,Z_{\mathbf{s}_0})$, respectively.

Because in the limit of $R_0\to\infty$, both the unconditioned and the conditioned mutual information become zero, the difference between them also becomes zero. Therefore, from the relations above, the following statements can be made:

1. If $\mathcal{I}(Z_s;Z_q|Z_{\mathbf{s}_0})\geq\mathcal{I}(Z_s;Z_q|V,Z_{\mathbf{s}_0})$ for $R_0=0$, then the former will remain greater than the latter for all non-negative R_0.
2. If $-\infty<\mathcal{I}(Z_s;Z_q|Z_{\mathbf{s}_0})-\mathcal{I}(Z_s;Z_q|V,Z_{\mathbf{s}_0})\leq 0$ for $R_0=0$, there exists $\bar{R}_0$ such that $\mathcal{I}(Z_s;Z_q)\geq\mathcal{I}(Z_s;Z_q|V,Z_{\mathbf{s}_0})$ for every $R_0>\bar{R}_0$.
3. $\mathcal{I}(Z_s;Z_q|Z_{\mathbf{s}_0})-\mathcal{I}(Z_s;Z_q|V,Z_{\mathbf{s}_0})$ will converge *down* to zero as R_0 increases from a positive value.

4. (Generalization of statement 1) If the information gain $\mathcal{I}(V; Z_{\mathbf{s}})$ is submodular for $R_0 = 0$, then it will be submodular for all $R_0 \geq 0$.

5. (Generalization of statement 2) Even if the information gain $\mathcal{I}(V; Z_{\mathbf{s}})$ is not submodular, it will become submodular as R_0 increases.

To sum up, submodularity of the mutual information, $\mathcal{I}(V; Z_{\mathbf{s}})$, can be interpreted as diminishing correlation coefficients over the search space by knowledge of the verification variables V. Also, since the information reward tends to be submodular as the sensing noise increases, the non-submodular aspect of the sensor targeting problem can be of more concern in design of networks with high quality sensing devices. However, in terms of the performance of the sequential greedy strategy for the sensing point targeting problem in this chapter, numerical results in the later section will empirically demonstrate that it exhibits good performance for both the non-submodular and submodular information rewards.

A.2 Numerical Results with Sensing-Point Targeting

Regarding submodularity of the information reward, the maximum and the minimum values of the difference between the prior and the posterior mutual information between two points in the search space are evaluated:

$$\max_{s,q \in \mathcal{S}} \mathcal{I}(Z_s; Z_q) - \mathcal{I}(Z_s; Z_q|V), \quad \text{and} \quad \min_{s,q \in \mathcal{S}} \mathcal{I}(Z_s; Z_q) - \mathcal{I}(Z_s; Z_q|V). \tag{A.27}$$

The non-negativity of the above min value is a necessary condition for submodularity of mutual information $\mathcal{I}(V; Z_{\mathbf{s}})$, because the if and only if condition of submodularity is $\mathcal{I}(Z_s; Z_q|Z_{\mathbf{s}_0}) \geq \mathcal{I}(Z_s; Z_q|Z_{\mathbf{s}_0})$, $\forall \mathbf{s}_0 \in \mathbb{S}_n$. Figure A-1 shows the max and the min values in (A.27) with respect to the sensing noise variance R. Note that for $R = 0.0004$, the min value of the gap is less than -0.2, which means that Table 3.3 indeed compares the performance of the strategies for a non-submodular information

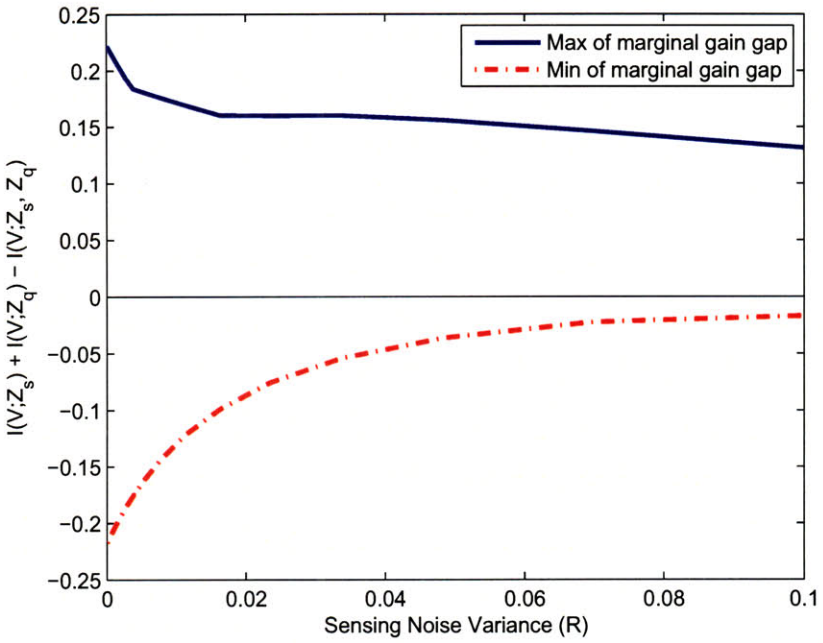

Figure A-1: The maximum and minimum value of gap between the prior and the posterior mutual information: negativity of the min value indicates non-submodularity of the information reward.

reward while the sequential greedy strategy provides good suboptimal solutions. In addition, as predicted in section A.1, the mutual information tends to be submodular as the sensing noise variance increases.

Appendix B

Extension of Smoother Form to Tracking Problems

This appendix extends the smoother form developed in Chapter 5 to another setting of information-theoretic path planning – the optimal *tracking* problem, in which the objective is to reduce the uncertainty in the quantity at the end of the sensing rather than in the far future, by analyzing the mutual information rate for the corresponding information reward function and proposing an effective receding-horizon formulation.

The optimal tracking problem can then be written as

$$\max_{\mathcal{Z}_\tau} \mathcal{I}(V_\tau; \mathcal{Z}_\tau). \qquad \textbf{(OTP)}$$

In other words, (**OTP**) finds the best (continuous) measurement history over $[0, \tau]$ that represents the largest mutual information achieved between the verification variables at τ.

B.1 Smoother-Form On-the-fly Information for Tracking (SOIT)

B.1.1 Information

In the process of computing the tracking mutual information $\mathcal{I}(X_\tau; \mathcal{Z}_\tau)$ by integrating forward the Lyapunov and Riccati equations in (5.10) and (5.11), the only two available matrix quantities are $P_X(t)$ and $Q_X(t)$. Using these, the mutual information between the current state and the measurement thus far, $\mathcal{I}(X_t; \mathcal{Z}_t$ can be calculated as discussed in section 5.2.4. However, this does not represent the information gathered by $\mathcal{Z}_t$ *for the final verification variables* V_τ, which is $\mathcal{I}(V_\tau; \mathcal{Z}_\tau)$.

This accumulated information for V_τ can be quantified by considering a forecasting problem with replacing T by τ and τ by t; then,

$$\mathcal{I}(V_\tau; \mathcal{Z}_t) = \mathcal{I}(X_t; \mathcal{Z}_t) - \mathcal{I}(X_t; \mathcal{Z}_t | V_\tau) \tag{B.1}$$

$$= \tfrac{1}{2}\left[\log\det S_{X|V_\tau}(t) - \log\det S_X(t)\right] - \tfrac{1}{2}\log\det\Big(I + Q_X(t)(S_{X|V_\tau}(t) - S_X(t))\Big) \tag{B.2}$$

where $S_{X|V_\tau}(t) \triangleq \mathbf{P}(X_t|V_\tau)^{-1}$, which is computed by integrating forward the following differential equation

$$\dot{S}_{X|V_\tau} = -S_{X|V_\tau}(A + \Sigma_W S_X) - (A + \Sigma_W S_X)' S_{X|V_\tau} + S_{X|V_\tau}\Sigma_W S_{X|V_\tau}. \tag{B.3}$$

The initial condition is given $S_{X|V_\tau}(0) = P_{0|V_\tau}^{-1}$, which can be expressed as

$$P_{0|V_\tau} = P_0 - P_0\Phi'_{(\tau,0)}M'_V P_V(\tau)^{-1} M_V \Phi_{(\tau,0)} P_0$$

where $\Phi_{(\tau,0)}$ is the state transition matrix from time 0 to τ.

Note that the expression in (B.2) is not well-defined at $t = \tau$, because $S_{X|V_\tau}(\tau)$ is singular. However, the relation in (B.1) holds even at time τ, and SOIT at time τ

can be computed as:

$$\mathcal{I}(V_\tau; \mathcal{Z}_\tau) = \mathcal{I}(X_\tau; \mathcal{Z}_t) - \mathcal{I}(X_\tau; \mathcal{Z}_t | V_\tau) \tag{B.4}$$

$$= \mathcal{I}(X_\tau; \mathcal{Z}_t) - \mathcal{I}(V_\tau^C; \mathcal{Z}_t | V_\tau) \tag{B.5}$$

where $V_\tau^C \triangleq X_\tau \setminus V_\tau^C$, or equivalently, $V_\tau^C = M_{VC} X_\tau$ for some M_{VC} such that M_{VC}' spans the null space of M_V. Alternatively, SOIT at τ can be calculated by the value of FOI at τ, because FOI and SOIT are identical at τ by definition.

B.1.2 Information Rate

The rate of SOIT can be quantified by adopting the expression of the rate of SOIF in Proposition 6; the only change is that $S_{X|V_\tau}(t)$ replaces $S_{X|V}(t)$ as follows.

$$\tfrac{d}{dt}\mathcal{I}(V_\tau; \mathcal{Z}_t) = \underbrace{\tfrac{1}{2}\mathtt{tr}\left\{\Sigma_N^{-1} C \Pi_\tau(t) C'\right\}}_{\texttt{Info Supply}}. \tag{B.6}$$

where $\Pi_\tau \triangleq Q_X(S_{X|V_\tau} - S_X)[I + Q_X(S_{X|V_\tau} - S_X)]^{-1} Q_X$.

Notice that, unlike the rate of FOI in (5.54) and the rate of PFOI in (5.58), the quantity in (B.6) is non-negative (except at $t = \tau$ where it is not defined), because the effect of the future process noise over $[t, \tau]$ is all encapsulated in $S_{X|V_\tau}$. Thus, by employing the smoother form expression derived for the forecasting problem, the pure impact of sensing for the tracking problem can be quantified.

B.2 On-the-fly Information for Receding-Horizon Formulations

The previous section analyzes the time derivatives of the on-the-fly information quantities from the perspective of the information supply and dissipation. Despite theoretically important features, the on-the-fly information quantities are not essential if the goal is simply to solve the optimal path planning problems in (**OTP**). However,

receding-horizon approximations of the original optimization problem are often used for better computational tractability and/or for better adaptation to changes in the environment. In this case the computation of on-the-fly information can be essential for this receding-horizon approximation, because the effect of a partial measurement history should be quantified. This section discusses the role of the smoother-form on-the-fly information quantities for the purpose of determining the cost-to-go functions of the receding-horizon formulations.

Consider a tracking decision for the horizon of $[0, \sigma]$, $\sigma < \tau$ when the ultimate goal is to maximize $\mathcal{I}(V_\tau; \mathcal{Z}_\tau)$. For this problem, this work suggests the following formulation based on the smoother-form on-the-fly information:

$$\max_{\mathcal{Z}_\sigma} \mathcal{I}(V_\tau; \mathcal{Z}_\sigma) \equiv \mathcal{I}(X_\sigma; \mathcal{Z}_\sigma) - \mathcal{I}(X_\sigma; \mathcal{Z}_\sigma | V_\tau). \qquad \textbf{(S-RH)}$$

In other words, the decision for the time window $[0, \sigma]$ maximizes the SOIT at the end of the current receding-horizon, which can be calculated by evaluating (B.1) at σ. Since the time derivative of SOIT is non-negative over $[0, \tau)$, the objective value of (**S-RH**) increases as σ increases. It is important to contrast (**S-RH**) to the following formulation based on the filter-form on-the-fly information:

$$\max_{\mathcal{Z}_\sigma} \mathcal{I}(X_\sigma; \mathcal{Z}_\sigma). \qquad \textbf{(F-RH)}$$

In other words, (**F-RH**) aims to minimize the entropy of the current state X_σ, and the underlying premise of this formulation is that an accurate estimate of the current states tends to result in an accurate estimate of the future verification variables.

The formulation (**F-RH**) is equivalent to the formulation in [33] that maximizes $\log\det J_X(\sigma)$ for the interval $[0, \sigma]$. Note that the objective value of (**F-RH**) is not necessarily an increasing function of σ for $\Sigma_N > 0$, because the rate of FOI can be negative if the information dissipation dominates the information supply, in particular, with large Σ_W.

In case $M_V = I$ (i.e., $V_\tau = X_\tau$) and there is no process noise over $(\sigma, \tau]$, (**F-RH**) becomes equivalent to (**S-RH**), because then $\mathcal{I}(X_\sigma; \mathcal{Z}_\sigma | X_\tau) = 0$ as there is no

remaining uncertainty in X_σ for a given X_τ. However, in general $\mathcal{I}(X_\sigma; \mathcal{Z}_\sigma | V_\tau) > 0$; therefore, the solutions to (**S-RH**) and (**F-RH**) differ. Also, the objective value of (**F-RH**) always overestimates that of (**S-RH**).

The difference between (**S-RH**) and (**F-RH**) can be explained in terms of information diffusion. It was shown in [23] that

$$\mathcal{I}(\mathcal{X}_{[\sigma,s]}; \mathcal{Z}_\sigma) = \mathcal{I}(X_\sigma; \mathcal{Z}_\sigma), \ \forall s > \sigma \tag{B.7}$$

where $\mathcal{X}_{[t_1,t_2]} \triangleq \{X_s : s \in [t_1, t_2]\}$. This is because the sufficient statistics for estimation the future state history $\mathcal{X}_{[\sigma,s]}$ is $\widehat{X}_\sigma \triangleq \mathbb{E}[X_\sigma | \mathcal{Z}_\sigma]$, which is identical to that for estimating the current state X_σ based on the past measurement history $\mathcal{Z}_\sigma$.

The relation in (B.7) specifically holds for $s \geq \tau$. In this case, the verification variables at τ become a subset of the future state history: i.e., $V_\tau \subset \mathcal{X}_{[\sigma,\tau]}$. Thus, what the filter-form on-the-fly information quantifies is the influence of the past measurement on the entire future, while the smoother-form on-the-fly information pinpoints the impact on some specified variables at some specific instance of time in the future. In this sense, the smoothing term in (**S-RH**), $\mathcal{I}(X_\sigma; \mathcal{Z}_\sigma | V_\tau)$ represents the portion of information gathered by $\mathcal{Z}_\sigma$ but will be diffused out to the space that is orthogonal to V_τ:

$$\mathcal{I}(X_\sigma; \mathcal{Z}_\sigma | V_\tau) = \mathcal{I}(\mathcal{X}_{[\sigma,\tau]} \setminus V_\tau; \mathcal{Z}_\sigma | V_\tau) \tag{B.8}$$

In case $M_V = I$, (B.8) specifically means the information diffused through the future process noise: $\mathcal{I}(\mathcal{W}_{(\sigma,\tau]}; \mathcal{Z}_\sigma | X_\tau)$ where $\mathcal{W}_{(\sigma,\tau]} \triangleq \{W_s : s \in (\sigma, \tau]\}$. It should be noticed that $\mathcal{I}(\mathcal{W}_{(\sigma,\tau]}; \mathcal{Z}_\sigma | X_\tau)$ can be non-zero, although $\mathcal{Z}_\sigma$ and $\mathcal{W}_{(\sigma,\tau]}$ are uncorrelated each other, because conditioning on X_τ can correlate $\mathcal{Z}_\sigma$ and $\mathcal{W}_{(\sigma,\tau]}$.

The formulations (**S-RH**) and (**F-RH**) are written for the horizon starting from the initial time; extension to the decisions for a later horizon $[\sigma_k, \sigma_{k+1}]$ is straightforward: they simply consider the conditioned mutual information conditioned on the previous measurement decision $\mathcal{Z}_\sigma^\star$.

B.3 Numerical Examples

B.3.1 Tracking Interpretation of Example in Chapter 5

Note that the weather forecasting example in Chapter 5 that finds the measurement history over $[0, \tau]$ to reduce the entropy of $V_T \triangleq M_V X_T$ can be interpreted as a receding-horizon subproblem of the tracking problem with T being replaced by τ and τ begin replaced by σ. In this case, the "optimal" and the "optimal myopic" solutions correspond to **S-RH** and **F-RH** formulations, respectively. Thus, the results in Chapter 5 demonstrates that a decision based on the smoother-form on-the-fly information and the filter-form on-the-fly information can be very different depending on the problem. In this example, the process noise turns out not to be a dominant factor that causes the difference, but the dominating factors are the fact that $\tau \gg \sigma$ and $M_V \neq I$.

B.3.2 Sensor Scheduling

A sensor scheduling problem with some artificial dynamics is considered as the second example; the key comparison made with this example is on the overall performance of solutions from S-RH and F-RH. The system matrices are

$$A = \begin{bmatrix} 0.1 & -0.01 \\ 0.005 & 0.07 \end{bmatrix}, \; P_0 = \begin{bmatrix} 1 & -0.6 \\ -0.6 & 1 \end{bmatrix} \tag{B.9}$$
$$\Sigma_W = \mathtt{diag}(0, 3), \qquad \Sigma_N = 1,$$

and $\tau = 5$. The size of planning horizon is 1, and for each horizon a decision is made either to measure the first state or the second one. Thus, the true optimal solution (OPT) can be found by simply enumerating possible $32(=2^5)$ cases; S-RH and F-RH solutions can also be found by enumeration.

Figure B-1 illustrates the switching sequences for the three solutions. It is first noticed that the three solutions are very different; especially, the filter-form receding horizon solution behaves in an opposite way to the true optimal solution. Fig-

ure B-2 shows the information accumulation in the three solutions, where both the smoother-form accumulated information $\mathcal{I}(X_\tau; \mathcal{Z}_{\sigma_k})$ and the filter-form accumulated information $\mathcal{I}(X_{\sigma_k}; \mathcal{Z}_{\sigma_k})$ for $\sigma_k = 1, 2, \ldots, 5$ are shown for comparison. Looking at the decisions for the first horizon: OPT and S-RH choose to measure state 1, but F-RH selects state 2. While the filter-form information indicates that measuring state 2 (follow dotted red line with marker $\times$) is slightly larger than that of measuring state 1 (follow dotted black line with marker $\square$), the smoother-form information says that the reward for observing state 1 is much larger than the other. It can be seen that the difference in this first decision leads to a larger gap in the final performance.

One important characteristics of the system in (B.9) is that a relatively large process noise is injected only to the dynamics of the second state variable. F-RH keeps measuring the second variable to compensate for the increase in the uncertainty in state 2 by the impact of the process noise over the time interval of size 1; however, in a long term view, the dynamics of the two variables are coupled and the effect of process noise is propagated to the first state variable. This results in a situation that F-RH is far from optimal. S-RH provides relatively good performance by taking account of the effect of the future process noise in the decision for each horizon.

B.3.3 Target Localization

The third example, which is adapted from [33], considers localization of a target whose location is fixed in a nominal sense using a mobile sensor. The main purpose of this example is to validate the presented receding-horizon formulation with consideration of nonlinearity in the measurement and replanning based on the actual measurement data.

The state vector X_t is defined to represent the position of the target, x_{tg} and y_{tg}. Since the target is assumed to be stationary, the system matrix $A = \mathbf{0}_{2\times 2}$. The process noise represents the drift of the target position in x- and y-directions;

$$\Sigma_W = \mathtt{diag}(0.2^2, 0.01^2)$$

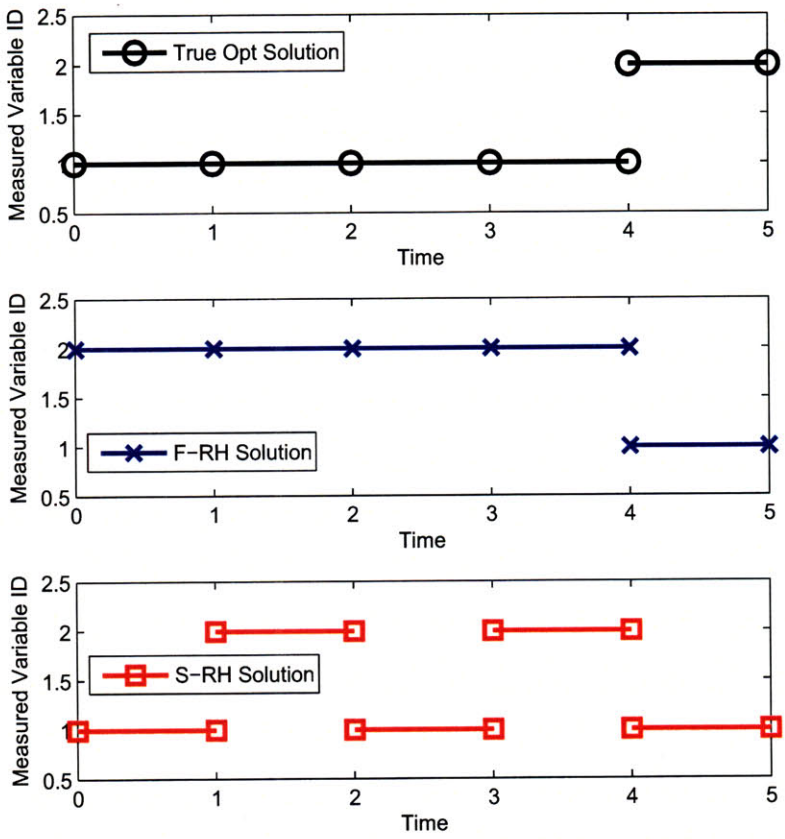

Figure B-1: Switching sequences for OPT, F-RH, and S-RH; F-RH solution behaves exactly opposite to OPT

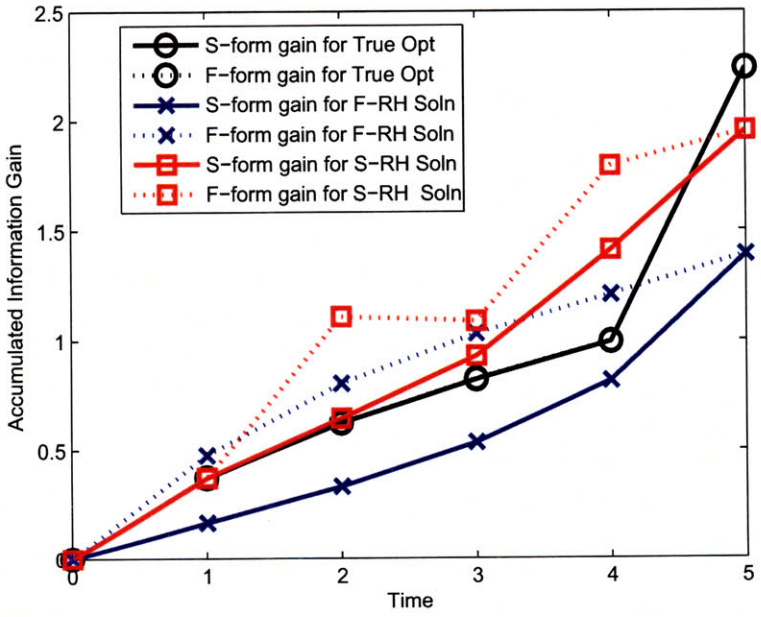

Figure B-2: Smoother-form and Filter-form information accumulation for OPT, F-RH, and S-RH; performance of F-RH is significantly worse than S-RH

is used. Note that the target is subject to a larger drift in x-direction than y-direction. The sensor measures the bearing angle between the target and itself:

$$Z_t = \texttt{atan}\{(y_{tg} - y_s)/(x_{tg} - x_s)\} + N_t \tag{B.10}$$

where the sensor's motion is described by (5.78). The target location is tracked by a *discrete* extended Kalman filter (EKF) using the bearing measurement taken with frequency of 16Hz with noise standard deviation of 2.5deg. Although the underlying estimator is discrete, the sensor planning problem for a horizon $[\sigma_k, \sigma_{k+1}]$ is posed in a continuous domain by linearizing (B.10) around the state estimate at σ_k. Then, the sensing noise intensity for this continuous planning becomes $\Sigma_N = 2.5^2(\pi/180)^2/16$. Once the plan for $[\sigma_k, \sigma_{k+1}]$ is made, the sensor executes it by moving along the planned path and taking discrete measurements every $1/16$ seconds. The decision for the next horizon, $[\sigma_{k+1}, \sigma_{k+2}]$ is made by incorporating the actual measurement up to σ_{k+1}. The total length of the planning window is 14 seconds, which is divided into 14 receding planning horizons. Also, $P_0 = 0.5^2 I$ is used.

Figure B-3 shows the trajectories for S-RH and F-RH solutions with the true target locations over $[0, \tau]$, which are very different from each other. The sensor moves mainly in y-direction in the S-RH solution but in x-direction in the F-RH solution. It can be seen in Figure B-4 that this difference in the path results in different characteristics in reducing the uncertainty of the target location. The top pictures that depict the time history of the entropy in the target's x- and y-position, respectively, indicate that the S-RH solution mainly reduces the uncertainty in the estimate of y_{tg}, while F-RH reduces that of x_{tg}. The entropy in x_{tg} is even increasing along the S-RH path up to 8 seconds, but it can be seen in the bottom picture that the overall entropy is decreasing over time. Since the target drifts largely in x-direction, F-RH tries to reduce the uncertainty in x_{tg} due to the process noise. However, S-RH takes a different option of reducing the uncertainty in y_{tg} and increasing the correlation between the estimates of x_{tg} and y_{tg}. The thinner lines in the bottom picture represent the summation of the entropy of x_{tg} and y_{tg}; a large gap between this summation and

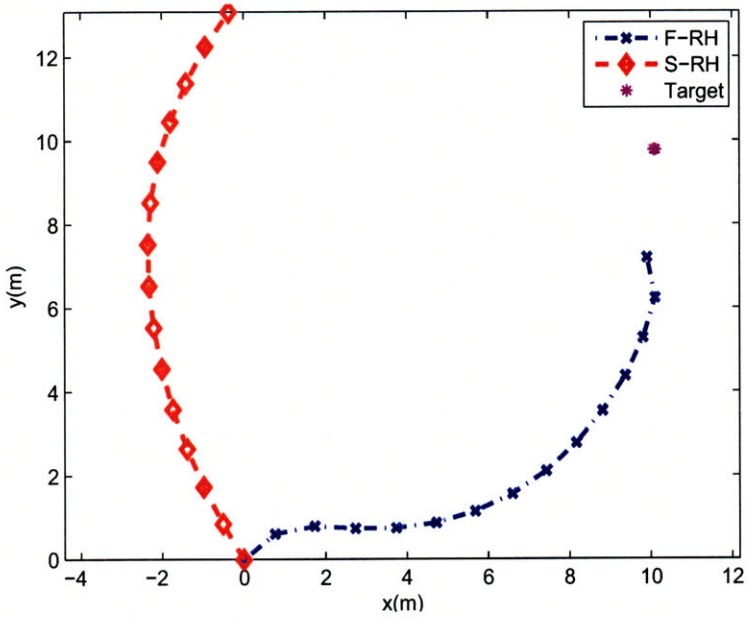

Figure B-3: Sensor trajectories for target localization: F-RH leads to motion in x-direction, while S-RH leads to motion in y-direction

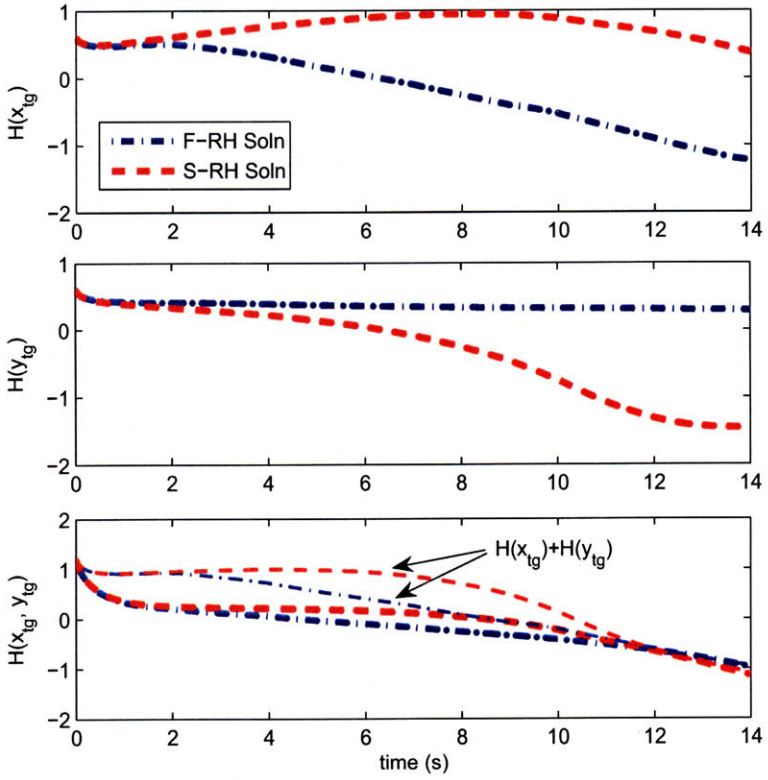

Figure B-4: Time histories of entropy of (a) target's x-position, (b) target's y-position, (c) overall position estimate: F-RH mainly reduces uncertainty in x-position while S-RH mainly reduces uncertainty in y-position

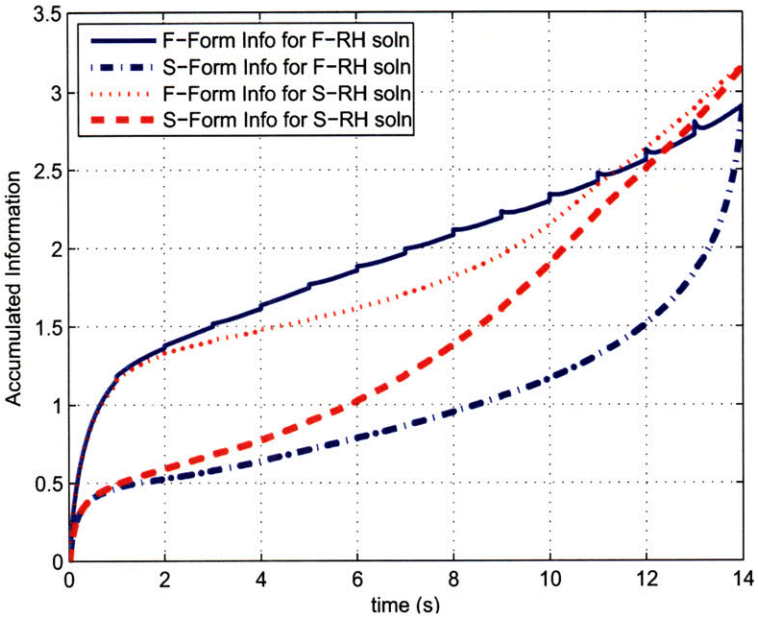

Figure B-5: Smoother-form and filter-form on-the-fly information values used for planning

the overall entropy means high correlation between two position estimates. By looking at the overall entropy at τ=14s in the bottom picture of Figure B-4, it can be also seen that S-RH provides a slightly better performance than F-RH in this example. Since both S-RH and F-RH are suboptimal strategies and the (linear) model used for planning is updated using the actual (nonlinear) measurement in this example, consideration of the *modeled* future process noise in S-RH does not necessarily lead to significantly superior performance to F-RH. However, it is important to notice the characteristics of the two receding-horizon formulations.

Figure B-5 illustrates the on-the-fly information (both S-Form and F-Form) used for the planning. The glitches seen every second are due to the modification of the linear model with the actual measurement values. Comparing the filter-form and the smoother-form information for the F-RH solution – blue solid line and blue dash-dotted line, it can be seen that the smoother-form information gathers a large amount of the information in the final phase (after 12s), while the filter-form assumes that it gathers almost a constant amount of information every time step after 2s. This information from the earlier time periods will experience the process noise for a longer time, and thus tends to be discounted in the smoother-form. Since in this example,

the dynamics of x_{tg} and y_{tg} are decoupled from each other, this discounting effect stands out.

Appendix C

An Outer-Approximation Algorithm for Maximum Entropy Sensor Selection

This appendix presents an efficient algorithm for selection of sensing points representing the largest uncertainty, which is called *maximum entropy sampling.* In the backward selection framework, the maximum entropy sampling is equivalent to the sensing point targeting problem in Chapter 3 with the conditional covariance being diagonal. The specific method presented in this appendix is the outer-approximation algorithm that formulates the sensing point selection process as a mixed-integer convex program. The methodology in this appendix can provide better efficiency in the targeting decision, if extended to the cases with non-diagonal conditional covariance.

C.1 Introduction

The decision making in this thesis to find a set of measurement points that leads to the largest reduction in entropy of certain random variables of interest, which can be referred to as *maximum information gain sampling* (MIGS). As a similar concept addressed in different contexts, *maximum entropy sampling* (MES) is decision making to select a set of design points representing the largest entropy, which was first

introduced for design of experiments [52] and was named by Shewry and Wynn [53]. In the case that all of the random variables are jointly Gaussian, MES corresponds to finding the submatrix of the covariance matrix that has the largest determinant; however, it was shown to be NP-hard even if all the entries of the covariance matrix are rational, independent of whether the size of the submatrix is predetermined or not [7].

MIGS is a different (and usually harder) problem than MES, since information gain, unlike entropy, is not submodular in general as discussed in section A.1; however, MIGS and MES are closely related. MIGS for which the posterior covariance matrix is diagonal can be reduced to a decision very similar to MES, which this appendix refers to as *generalized maximum entropy sampling* (GMES). Since submodularity holds for GMES, solution techniques for MES can incorporate GMES without extensive modification. A sensor selection problem for moving target tracking presented in Williams et al. [54] is a GMES problem. GMES is further reduced to MES, if the posterior covariance matrix is a scalar multiplier of the identity matrix and the number of selection points is given. Although results in Table 3.3 showed that MES decision is not particularly good for the observation targeting for the weather prediction, it has been approximated MIGS when the computation of the conditional entropy is computationally intractable [12], Thus, developing a good solution strategy for MES (or GMES) can be conducive to solving MIGS.

One approach to find the optimal solution of MES (or MIGS) for the Gaussian case is to formulate it as an optimization problem employing binary variables to indicate which rows and columns of the covariance matrix will be chosen. This type of approach has been quite successful, and all existing optimization-based methods have been based on the branch-and-bound (BB) algorithm with various upper bounding mechanisms: largest eigenvalues of the covariance matrix [7, 15, 55, 56], nonlinear programming relaxation using rational and exponential mixture function of the binary variables [15, 57], partition-based spectral bound [58], linear-integer programming bound for improving the spectral bound [59], and factored mask spectral bound [43] that generalizes the eigenvalue bound and the partition-based bound.

In contrast, this appendix addresses the generalized maximum entropy sampling problem within the *outer-approximation* (OA) framework. The OA algorithm, which was first developed by Duran and Grossmann [60], extended to a general mixed-integer convex program (MICP) [61] and to mixed-integer (nonconvex) nonlinear programs (MINLP) [62, 63], alternately solves a primal problem and a relaxed master problem. The primal problem is a nonlinear program (NLP) with all the integer variables being fixed in value, and the relaxed master problem is a mixed-integer linear program (MILP) constructed by linearizing the objective function and the constraints around a solution point of the primal problem. At each iteration, the best primal solution so far provides a lower bound (LBD) (in case of maximization), while the relaxed master problem gives a upper bound (UBD) on the optimal solution value and determines the next integer value to visit. The algorithm terminates at the point that UBD converges to LBD, thus guaranteeing global optimality.

The comparison of OA and BB in terms of computation time is controversial and problem-specific; however, OA has the following nice properties, which are exploited in the algorithm presented herein. First, it is no longer a concern to devise algorithmic heuristics such as the branching order and node selection, which make a lot of difference in BB, because the order of integer values to visit is automatically determined by the relaxed master problem. Second, for pure integer-convex programs (ICP), the primal problem becomes just a function evaluation and only a sequence of MILPs needs to be solved. Thus, with a reliable solver for MILP such as CPLEX [64], an ICP can be solved very efficiently by OA.

This appendix presents a mixed-integer semidefinite program (MISDP) formulation for generalized maximum entropy sampling, in which binary variables indicate selection of the corresponding rows and columns, and continuous variables enable a convex reformulation of the objective function and the constraint functions. It will be shown that this formulation does not require the solution of any primal semidefinite program (SDP), since the primal feasible set is reduced to a singleton; therefore, only MILP relaxed master problems need to be solved. Algorithmic details of the proposed approach are described while highlighting the relative convenience of the computa-

tion of gradient and Hessian information in contrast to the case of the nonlinear programming-based BB (BB-NLP) algorithm. Numerical experiments validate the suggested method and compare its computation time with the BB-NLP method. In particular, the sensor management problem, which addresses measurement selection under a limited communication budget in order to minimize tracking uncertainty of a moving target, is presented to distinguish the performance of the proposed algorithm from that of the BB-NLP algorithm.

C.2 Problem Formulation

C.2.1 Generalized Maximum Entropy Sampling

Maximum entropy sampling determines the set of sensing points from a given search space that represents the largest entropy amongst them. If the joint probability distribution for any subset of the search space is Gaussian, MES corresponds to picking a principal submatrix of the covariance matrix for the search space $P \in \mathbb{R}^{N\times N}$ that provides the largest determinant:

$$\max_{\mathbf{s}\subset\mathcal{S}:|\mathbf{s}|=n} \log\det P[\mathbf{s},\mathbf{s}] \qquad \textbf{(MES)}$$

where $\mathcal{S} \triangleq [1,N]\cap\mathbb{Z}$ and $P[\mathbf{s},\mathbf{s}]$ denotes the $n\times n$ principal submatrix of P consisting of rows and columns indicated by index set $\mathbf{s}$. For the sake of well-posedness of the problem, P should be symmetric positive definite. The cardinality of $\mathbf{s}$ is usually specified, as otherwise the solution of (**MES**) is trivially $\mathcal{S}$ by the principle of "information never hurts." [35] The existence of other constraints may allow for removal of the cardinality restriction, although most existing algorithms for MES have assumed specified cardinality.

This appendix considers the following constrained decision called generalized MES:

$$\begin{aligned} \max_{\mathbf{s}\subset\mathcal{S}} \quad & \log\det P[\mathbf{s},\mathbf{s}] - \log\det Q[\mathbf{s},\mathbf{s}] \\ \text{s.t.} \quad & A_{eq}\mathbf{y} = \mathbf{b}_{eq}, \quad A\mathbf{y} \leq \mathbf{b} \\ & \mathbf{y}^T G_k \mathbf{y} \leq g_k, \quad k = 1, \cdots, m, \end{aligned} \tag{GMES}$$

with $Q \succ 0$ being diagonal. The i-th element of the binary vector $\mathbf{y} \in \{0,1\}^N$ is related to $\mathbf{s}$ such that $y_i = 1$ if $i \in \mathbf{s}$, and 0 otherwise. In the context of sensor networks, Q may represent the posterior covariance matrix for the search space by the backward selection formulation described in Choi et al. [65]. Since Q is diagonal, $\log\det Q[\mathbf{s},\mathbf{s}] = \sum_{i=1}^{N} y_i \log q_{ii}$ where q_{ii} is the (i,i) element of Q. Regarding the constraints, a linear equality constraint can represent the cardinality constraint, while a linear inequality constraint can model power (or economic) budget limitation. The quadratic constraints can be used to represent restriction of communication capability. Note that the quadratic constraints defined by $G_k \in \mathbb{R}^{N\times N}$ and $g_k \in \mathbb{R}$ are in general *nonconvex*. To the authors' best knowledge, no optimization algorithm for MES has taken into account quadratic constraints, although information maximization with consideration of communication budget has been one of the most important issues in sensor network applications.

C.2.2 Mixed-Integer Semidefinite Program Formulation

This appendix poses the following mixed-integer semidefinite program (MISDP), which is a mixed-integer convex program (MICP), to address (**GMES**) described

in the previous section:

$$\max_{\mathbf{y},\mathbf{x}} \; f(\mathbf{y},\mathbf{x}) \equiv \log\det S(\mathbf{y},\mathbf{x}) + \mathbf{c}^T\mathbf{y} + \mathbf{d}^T\mathbf{x} \qquad (\mathbf{P})$$

s.t.

$$S(\mathbf{y},\mathbf{x}) \equiv I + \textstyle\sum_{i=1}^{N} y_i Y_i + \sum_{i=1}^{N-1}\sum_{j>i} x_{ij} X_{ij} \succ 0 \qquad (C.1)$$

$$x_{ij} \le y_i, \quad x_{ij} \le y_j, \quad x_{ij} \ge y_i + y_j - 1, \quad \forall i, \; \forall j > i \qquad (C.2)$$

$$A_{eq}\mathbf{y} + B_{eq}\mathbf{x} = \mathbf{b}_{eq}, \quad A\mathbf{y} + B\mathbf{x} \le \mathbf{b} \qquad (C.3)$$

$$\mathbf{y} \in \{0,1\}^N, \qquad \mathbf{x} \in [0,1]^{N(N-1)/2} \qquad (C.4)$$

where Y_i and X_{ij} are defined as

$$Y_i = (p_{ii} - 1)[\mathbf{e}_i\mathbf{e}_i^T], \quad X_{ij} = p_{ij}[\mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T]. \qquad (C.5)$$

p_{ij} is the (i,j) element of the matrix P, and $\mathbf{e}_i$ is the i-th unit vector. The set of linear constraints in (C.2) equivalently represent the bilinear relation $x_{ij} = y_i y_j$ when y_i and y_j are integers. Thus, (C.2) being satisfied, $S(\mathbf{y},\mathbf{x})$ is related to the original covariance as follows

$$[S(\mathbf{y},\mathbf{x})]_{ij} = \begin{cases} p_{ij}, & \text{if } y_i = y_j = 1 \\ \delta_{ij}, & \text{otherwise} \end{cases} \qquad (C.6)$$

where δ_{ij} is a Kronecker delta. Thus, the determinant of $S(\mathbf{y},\mathbf{x})$ is equal to that of $P[\mathbf{s},\mathbf{s}]$. The linear matrix inequality (LMI) constraint in (C.1) maintains the positive definiteness of $S(\mathbf{y},\mathbf{x})$; note that positive definiteness is always satisfied with binary $\mathbf{y}$ and corresponding $\mathbf{x}$ that satisfies (C.2). Also note that a nonconvex quadratic constraint in (**GMES**) can be written as a linear constraint in terms of both $\mathbf{y}$ and $\mathbf{x}$ by replacing y_i^2 by y_i and $y_i y_j$ by x_{ij} for binary y_i's. Similarly, the linear term $\mathbf{d}^T\mathbf{x}$ in the objective function enables consideration of a bilinear cost function, although it is not involved in (**GMES**).

Observe that (**P**) becomes a convex program if integrality of $\mathbf{y}$ is relaxed, since

LMI and linear constraints comprise a convex feasible set and $\log\det(\cdot)$ is a concave function in the space of symmetric positive definite matrices [46]. It should be pointed out that (**P**) is not the only possible way to formulate a MICP for GMES; however, the authors have found that the rational-exponential mixture formulation given in [57] is not suitable for the purpose of applying the outer-approximation algorithm because the gradient and Hessian are not defined everywhere for that formulation, while (**P**) might not be suitable for implementing branch-and-bound because of the computational burden of solving a large SDP relaxation.

C.3 Algorithm

C.3.1 Primal Problem

The primal problem for the k-th iteration of the OA algorithm is, in general, a convex program finding a best real decision vector $\mathbf{x}^\star(\mathbf{y}^k)$ for a given integer decision vector $\mathbf{y}^k$. In the case of pure integer programming, this reduces to a function evaluation using $\mathbf{y}^k$. It is noticed that the latter is the case for (**P**), although continuous decision variables x_{ij}'s are apparently involved. For any integer $\mathbf{y}^k$, constraint (C.2) restricts the feasible set for $\mathbf{x}$ to a singleton; the primal optimal solution is

$$x_{ij}^\star(\mathbf{y}^k) = y_i^k y_j^k \tag{C.7}$$

for a feasible $\mathbf{y}^k$. Then, the primal optimal objective value $f(\mathbf{y}^k, \mathbf{x}^\star(\mathbf{y}^k))$ becomes an underestimate of the optimal value of (**P**); if it is larger than the tightest lower bound LBD, it replaces LBD.

The integer vector $\mathbf{y}^k$ is the optimal solution to $(k-1)$-th relaxed master problem (section C.3.2) for $k > 1$; such $\mathbf{y}^k$ is always a feasible solution to (**P**), if (**P**) itself is a feasible problem. In order to generate the initial binary vector $\mathbf{y}^1$, this appendix proposes a MILP feasibility problem:

$$\max_{\mathbf{y},\mathbf{x}} \ \textstyle\sum_{i=1}^{N} y_i \log p_{ii} + \mathbf{c}^T\mathbf{y} + \mathbf{d}^T\mathbf{x}$$

subject to the same linear constraints as (**P**). This MILP provides an upper bounding solution to feasible (**P**) [66]; its infeasibility means (**P**) is an infeasible problem. In case the only constraint is cardinality restriction, the greedy solution [65] is a good feasible candidate for $\mathbf{y}^1$.

C.3.2 Relaxed Master Problem

The relaxed master problem is, in general, a mixed-integer linear program that optimizes the linear outer approximation of the objective function linearized at primal solution points over the feasible set. The relaxed master problem of (**P**) for the k-th iteration is written as follows:

$$\max_{\eta_k, \mathbf{y}, \mathbf{x}} \ \eta_k \tag{$\mathbf{M}^k$}$$

s.t.

$$\eta_k \leq f(\mathbf{y}^m, \mathbf{x}^\star(\mathbf{y}^m)) + \nabla f(\mathbf{y}^m, \mathbf{x}^\star(\mathbf{y}^m))^T \begin{pmatrix} \mathbf{y} - \mathbf{y}^m \\ \mathbf{x} - \mathbf{x}^\star(\mathbf{y}^m) \end{pmatrix}, \ \forall m \leq k \tag{C.8}$$

$$x_{ij} \leq y_i, \ x_{ij} \leq y_j, \ x_{ij} \geq y_i + y_j - 1, \ \forall i, \ \forall j > i \tag{C.9}$$

$$A_{eq}\mathbf{y} + B_{eq}\mathbf{x} = \mathbf{b}_{eq}, \ A\mathbf{y} + B\mathbf{x} \leq \mathbf{b} \tag{C.10}$$

$$\eta_k \in \mathbb{R}, \ \mathbf{y} \in \{0,1\}^N, \ \mathbf{x} \in [0,1]^{N(N-1)/2}. \tag{C.11}$$

The outer approximation of the LMI constraint in (C.1) can be neglected because (C.9) defines a subset of the feasible set of the LMI. Note that η_k is non-increasing in k because one constraint is added at every iteration, and it provides an upper bound on the optimal value $f^\star$. Thus, at every iteration η_k represents the tightest upper bound UBD. The algorithm terminates when UBD = LBD at a global optimum; every ($\mathbf{M}^k$) is feasible before termination, if (**P**) is feasible.

The gradient of the objective function $\nabla f(\mathbf{y}, \mathbf{x}^\star(\mathbf{y}))$ can be computed as

$$\frac{\partial f}{\partial y_i} = [S(\mathbf{y}, \mathbf{x})^{-1}]_{ii} (p_{ii} - 1) + c_i \tag{C.12}$$

$$\frac{\partial f}{\partial x_{ij}}\Big|_{\mathbf{x}^\star(\mathbf{y})} = 2 [S(\mathbf{y}, \mathbf{x})^{-1}]_{ij} p_{ij} + d_{ij} \tag{C.13}$$

by exploiting the self-concordance of the log det function [67] where c_i and d_{ij} are corresponding elements in the linear objective term. It is noted that computation of the above gradient does not require inversion of a (possibly) large matrix $S(\mathbf{y}, \mathbf{x})$, which was often required for the case for the NLP-based branch-and-bound algorithm [57], since $S(\mathbf{y}, \mathbf{x})^{-1}$ is a sparse matrix with a very special form. It can be shown that

$$\left[S(\mathbf{y}, \mathbf{x})^{-1}\right]_{ii} = 1, \qquad \text{if } y_i = 0 \tag{C.14}$$

$$\left[S(\mathbf{y}, \mathbf{x})^{-1}\right]_{ij:\, i \neq j} = 0, \qquad \text{unless } y_i = y_j = 1. \tag{C.15}$$

Therefore, $S(\mathbf{y}, \mathbf{x})^{-1}$ can be computed effectively by inverting the submatrix corresponding to those $y_i = 1$.

C.4 Numerical Experiments with MES

For validation of the proposed method, unconstrained MES problems that involve only the cardinality condition are first considered. Monte-Carlo experiments are performed using MATLAB 7.1 with TOMLAB/CPLEX 10.0 [68] to solve the MILP relaxed master problems. The covariance matrix is randomly generated as:

$$P = \frac{1}{L_E - 1} \mathbf{X}\mathbf{X}', \quad \mathbf{X} \in \mathbb{R}^{N \times L_E} \tag{C.16}$$

where each entry $\mathbf{X}(i, j)$ is i.i.d with $\mathcal{N}(0, 1)$. For the purpose of comparison, NLP-based branch-and-bound (BB-NLP) [57] is also implemented for the same setting, with TOMLAB/BARNLP [69] being used to solve the associated NLP relaxations. The greedy rule [7] is adopted to determine the branching order, and a node corresponding to the largest upper bound is selected first. Optimality of the solutions

Table C.1: Average Computation time (sec.) [# of UBD computations]

N	n	OA		BB-NLP		# of cand.
20	10	3.5	[19.4]	11.1	[3.4]	184756
30	10	10.7	[20.8]	124.9	[8.6]	30045015
30	15	288.9	[122.8]	103.6	[10.6]	155117520
40	10	46.4	[36.0]	>1000	[N/A]	847660528

by OA and BB-NLP is verified by comparing them with the solution from explicit enumeration with small-size problems.

Table C.1 represents the average computation time and the average number of upper-bounding problems – MILPs for OA and NLPs for BB-NLP – of both algorithms. Five different P matrices for each (N, n) setting are generated with $M(= 10N)$ sample vectors. The initial binary value $\mathbf{y}^1$ is selected in a greedy way. It is found that both algorithms perform comparably in general; but, for a certain size of problem OA performs much faster than BB-NLP.

C.5 Constrained Sensor Management Problem

C.5.1 Problem Description

This appendix addresses the same sensor management problem described in (3.1.2), with the backward selection formulation:

$$\max_{\mathbf{s}\in\mathbb{S}_n} \mathcal{H}(\mathbf{z}_t^{\mathbf{s}}|\mathbf{z}_{0:t-1}) - \mathcal{H}(\mathbf{z}_t^{\mathbf{s}}|\mathbf{x}_t, \mathbf{z}_{0:t-1}). \tag{C.17}$$

The selection decision incurs communication cost depending on the communication topology. This appendix assumes that direct communication between two sensors incurs a cost proportional to the squared distance between them:

$$\tilde{B}_{ij} = \gamma||l^i - l^j||_2^2 \tag{C.18}$$

with an appropriate scaling coefficient γ, and that distant sensors can communicate each other using a multi-hop scheme. Thus, the communication cost between two arbitrary sensors is the accumulated cost along the shortest (in a squared distance sense) path:

$$B_{ij} = \sum_{k=1}^{n_{ij}} \tilde{B}_{i_{k-1}i_k} \tag{C.19}$$

where $\{i_0, \cdots, i_{n_{ij}}\}$ is the shortest path from the sensor $i = i_0$ to $j = i_{n_{ij}}$. this appendix considers a particular worst case scenario in which every sensor must communicate to every other sensor in the selected set. The communication budget constraint in this case is written as

$$\textstyle\sum_{i,j\in\mathbf{s}_t} B_{ij} \leq B_{\max}. \tag{C.20}$$

Thus, sensor selection with a communication constraint can be written as a generalized maximum entropy sampling problem:

$$\begin{aligned} &\max_{\mathbf{s}_t} \log\det P_{\mathcal{S}}[\mathbf{s}_t, \mathbf{s}_t] - \textstyle\sum_{i=1}^{N} y_i \log R_i \\ &\text{s.t. } \textstyle\sum_{i=1}^{N} \sum_{j>i} B_{ij} y_i y_j \leq B_{max} \end{aligned} \tag{GMES-S}$$

where the covariance matrix of the search space

$$P_{\mathcal{S}} \triangleq H_{\mathbf{s}} \mathbf{P}(\mathbf{x}_t|\mathbf{z}_{0:t-1}) H_{\mathbf{s}}' + R_{\mathcal{S}} \in \mathbb{R}^{N\times N} \tag{C.21}$$

Note that the cardinality of $\mathbf{s}$ is not specified in advance. In this appendix, the state covariance estimate $\mathbf{P}(\mathbf{x}_t|\mathbf{z}_{0:t-1})$ is provided by an extended Kalman filter (EKF). Given this information, the presented outer-approximation algorithm can be implemented to (**GMES-S**) straightforwardly.

C.5.2 Modification of BB-NLP

A modified version of BB-NLP method is considered for comparison with the proposed outer-approximation algorithm; modification is needed because the original BB-NLP cannot handle quadratic constraints and unspecified cardinality. Introducing addi-

tional real variables $x_{ij} = y_i y_j$ with the set of constraints in (C.2) enables BB-NLP to deal with quadratic constraints. The original BB-NLP explicitly utilizes cardinality information to effectively construct the branch-and-bound tree. Two types of modification can be conceived regarding unspecified cardinality. One way is solving (**GMES-S**) with an additional cardinality constraint $\mathbf{1}^T\mathbf{y} = n$ for reasonably chosen values of n – call this way BB-NLP(1). The other way is modifying the branch-and-bound tree in such a way that lower bounds are computed for intermediate nodes as well as the leaf nodes, and leaf nodes are determined by infeasibility of the communication constraint rather than by the cardinality – denote this as BB-NLP(2). It was found empirically that the first way is usually faster than the second for small-size problems, while the opposite is the case for large-size problems.

C.5.3 Numerical Results

For numerical experiments, the following parameter values are set to be the same as in [54]:

$$\Delta t = 0.25,\ \bar{w} = 0.01,\ \rho_1 = 2000,\ \rho_2 = 100,\ R_i = 1. \tag{C.22}$$

A total of N sensors are located at fixed locations determined randomly on a 20×20 two-dimensional space; the pairwise communication cost values B_{ij}'s are computed by solving a shortest-path problem using dynamic programming [70]. The initial state value is $\mathbf{x}_0 = [0,\ 2,\ 0,\ 2]'$, which results in the nominal position at t-th time step $(0.5t, 0.5t)$. The (**GMES-S**) sensor selection is addressed at time $t = 20$, before which an EKF has used randomly selected $n_0 = 10$ sensor measurements for state estimation every time step.

$N = 30, 40$ are used; five randomly generated sets of sensor deployments are considered for each N, while three different values of $B_{\max} = 100, 200, 300$ are taken into account for each deployment. The modified branch-and-bound method, BB-NLP(2) is used, as it performs faster than BB-NLP(1) for most problems of this size. Every MILP relaxed master problem in OA is solved using TOMLAB/CPLEX 10.0; TOMLAB/KNITRO [71] is utilized to solve NLP upper bounding subproblems for

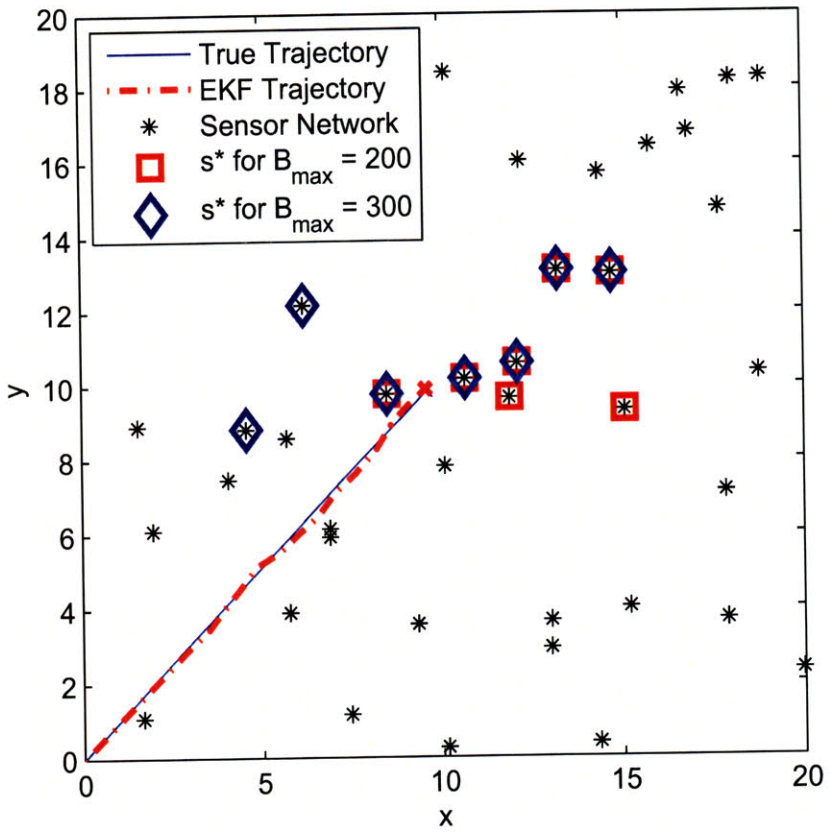

Figure C-1: An illustrative solution representing trade-off between information and communication ($N = 40$)

BB-NLP(2).

Table C.2 shows average computation times and numbers of upper bounding problems for OA and BB-NLP(2) for various $(N, B_{\max})$ settings. The maximum cardinality of feasible sensor selection, $n_{\max}$, is also tabulated as an indicator of the problem complexity. Optimality of the solutions from OA and BB-NLP(2) are verified by crosscheck. First, it is noticeable that OA performs an order-of-magnitude faster than BB-NLP(2) with less than 100 subproblem calculations being needed for all the cases. BB-NLP requires a much larger number of subproblem computations than OA, while it solved less subproblems than OA for unconstrained MES cases. Seeing as unit computation time per UBD computation for BB-NLP is small, it can be inferred that the main cause of large computation time for BB-NLP is not merely introduction of additional variables x_{ij}'s but weakness of upper bounds from its NLP relaxations. The linear representation in (C.2) is equivalent to the bilinear relation $x_{ij} = y_i y_j$ for integral $\mathbf{y}$; however, such x_{ij} can be far from $y_i y_j$ if integrality of $\mathbf{y}$ is relaxed.

Table C.2: Avg. Comp. time (sec.) [# of UBD computations] for SMP

N	$B_{\max}$	OA		BB-NLP		$n_{\max}$
30	100	8.9	[7.0]	633.6	[4216]	6
30	200	20.2	[14.3]	870.6	[6794]	7
30	300	69.9	[27.8]	1419.8	[12017]	7.75
40	100	101.8	[38.0]	>1hr	[N/A]	7
40	200	216.7	[37.3]	>1hr	[N/A]	7.67
40	300	934.9	[83.5]	>1hr	[N/A]	8.33

Regarding scalability of OA, bigger N leads to longer computation time in two aspects: first, it increases the number of decision variables and constraints, and second, it results in a larger total number of feasible candidates for a given $B_{\max}$. For the same value of $B_{\max}$, computation time for $N = 40$ is about ten times longer than for $N = 30$. It is also found that bigger $B_{\max}$ leads to longer computation time for given N; however, the total number of UBD computations does not increase as fast as the computation time in this case. This implies that the computation time grows mainly because unit computation time for solving each MILP increases rather than because upper bounds provided by the MILPs weaken. Note that the feasible set becomes larger as $B_{\max}$ increases; thus, each MILP has to consider a bigger branch-and-cut tree (CPLEX utilizes branch-and-cut algorithms for solving a MILP).

The optimal selection for larger $B_{\max}$ usually consists of more sensors than that for smaller communication budget – on average, 5 and 6.75 sensors for $B_{\max} = 100$ and 300 for both $N = 30$ and 40. On the other hand, Fig. C-1 illustrates the case for which both the solutions for $B_{\max} = 200$ and 300 consist of 7 sensors, to effectively represent the trade-off between information and communication. The solid and dashdotted lines depict the actual and estimated trajectories of the target until $t = 20$ at which the one-step lookahead sensor management decision is made. The optimal solution for $B_{\max} = 300$ (blue diamonds) turns out to be the optimal solution for unconstrained MES with fixed cardinality of 7; thus, it is the best way choosing 7 sensors if an infinite amount of communication is allowed. Under the limitation of the communication budget $B_{\max} = 200$, the optimal solution (red squares) selects

two nearby sensors instead of two sensors far from the other five.

C.6 Concluding Remarks

This appendix presented the outer-approximation approach to a generalized maximum entropy sampling problem. The mixed-integer semidefinite programming formulation was newly proposed; the outer-approximation algorithm resulting in a sequence of mixed-integer linear programs is presented. Numerical experiments verified that the performance of the suggested method is superior to the existing nonlinear programming-based branch-and-bound method especially in solving quadratically constrained problems such as communication-constrained sensor management. Future work will extend the presented outer-approximation algorithm to more general maximum information gain sampling. Also, other outer-approximation-based algorithms such as LP/NLP-based branch-and-bound [72] and branch-and-cut [73] can be adopted within the same MISDP framework.

Appendix D

More Numerical Results for Chapter 5

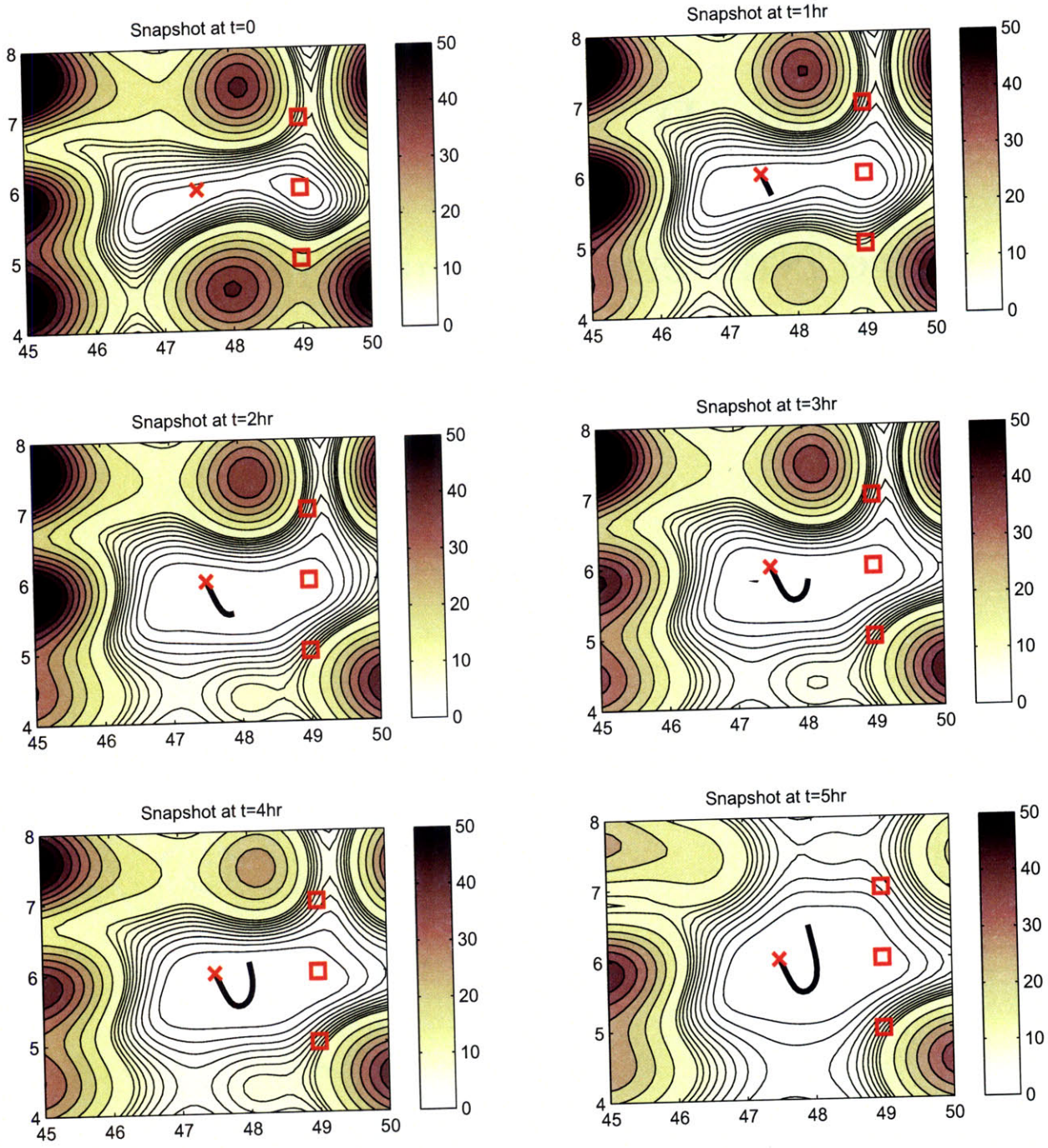

Figure D-1: Snapshots of the optimal trajectory with evolution of information field (Scenario 1)

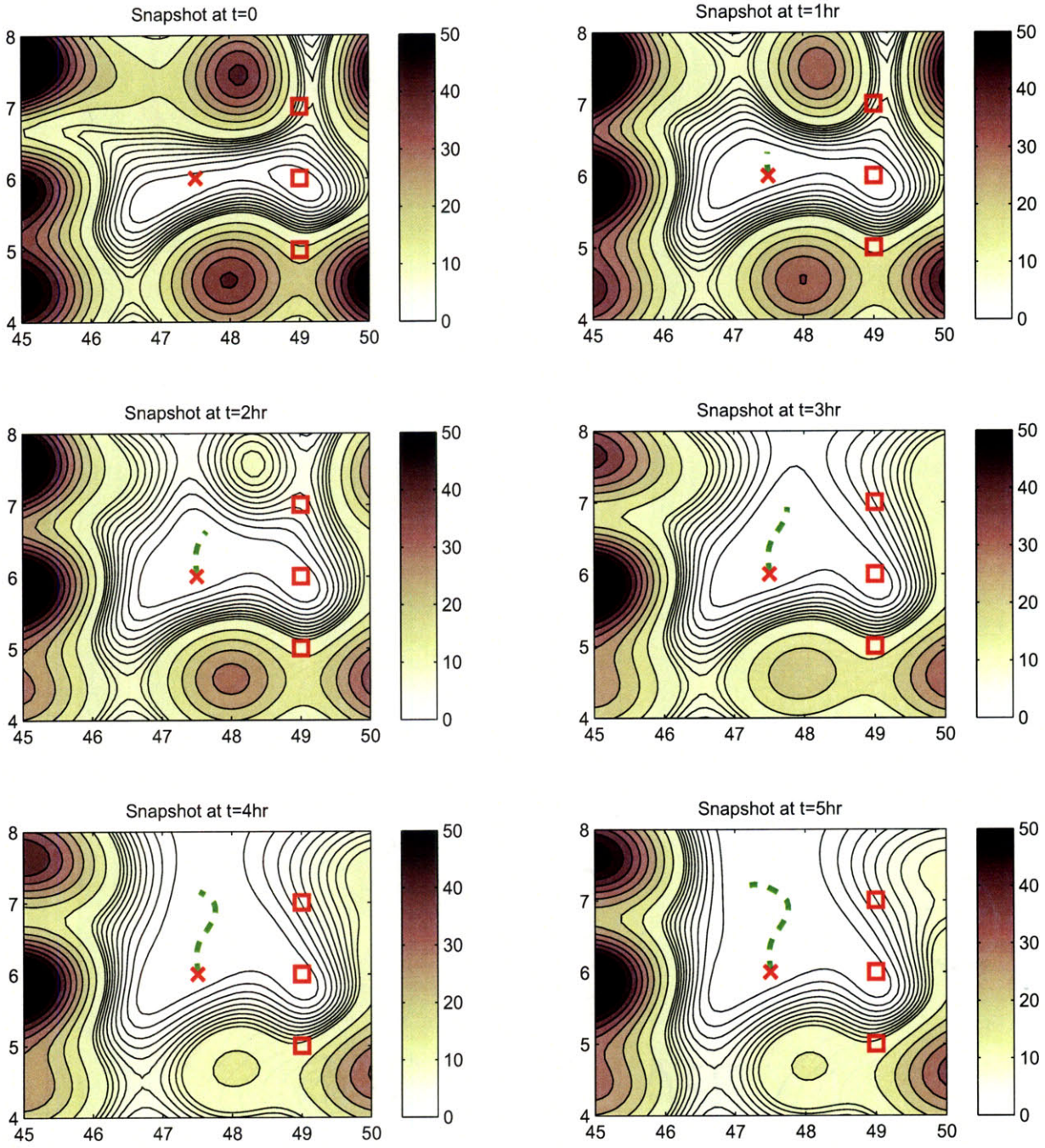

Figure D-2: Snapshots of the trajectory of real-time steering with evolution of information field (Scenario 1)

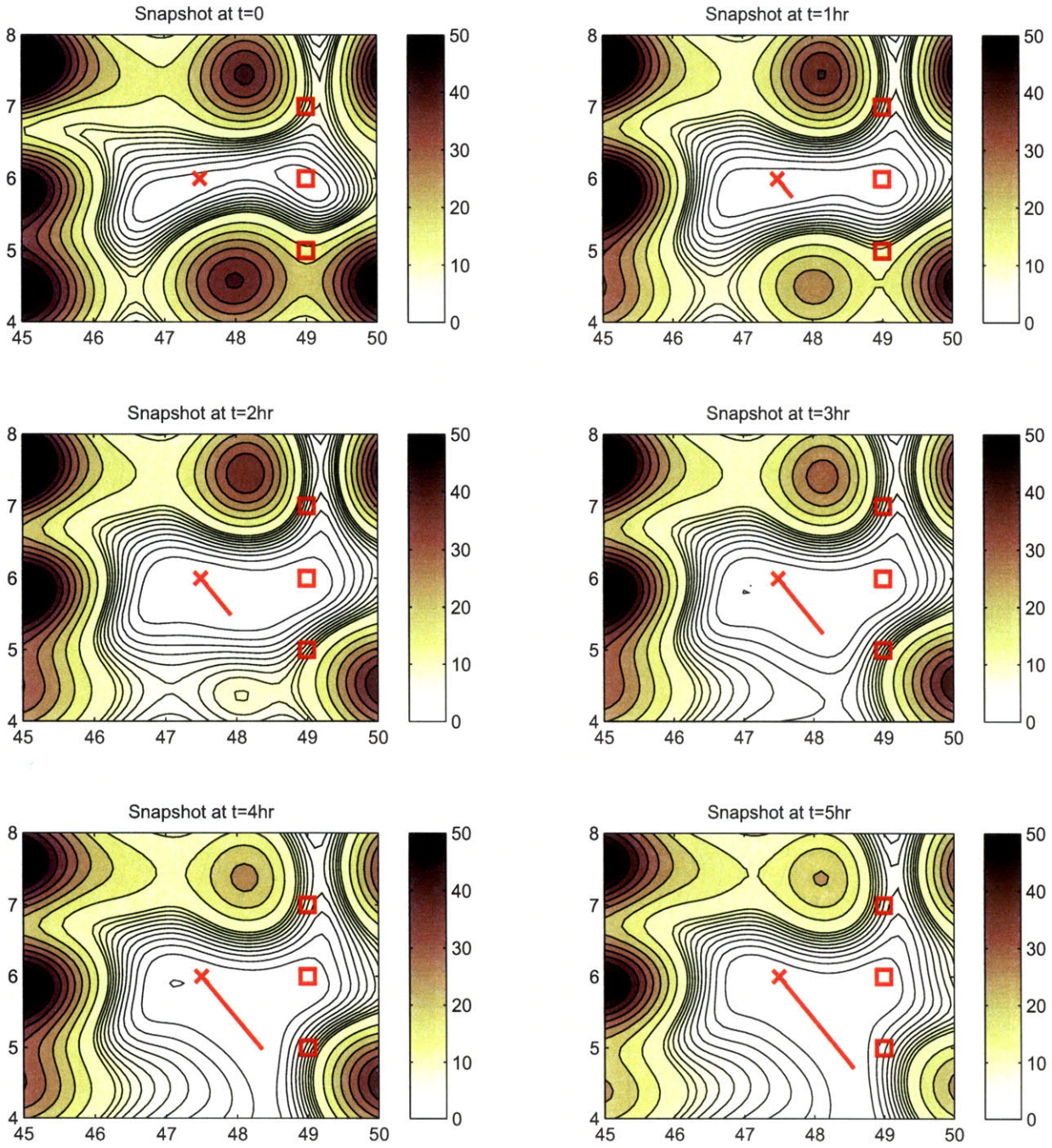

Figure D-3: Snapshots of the best straight-line trajectory with evolution of information field (Scenario 1)

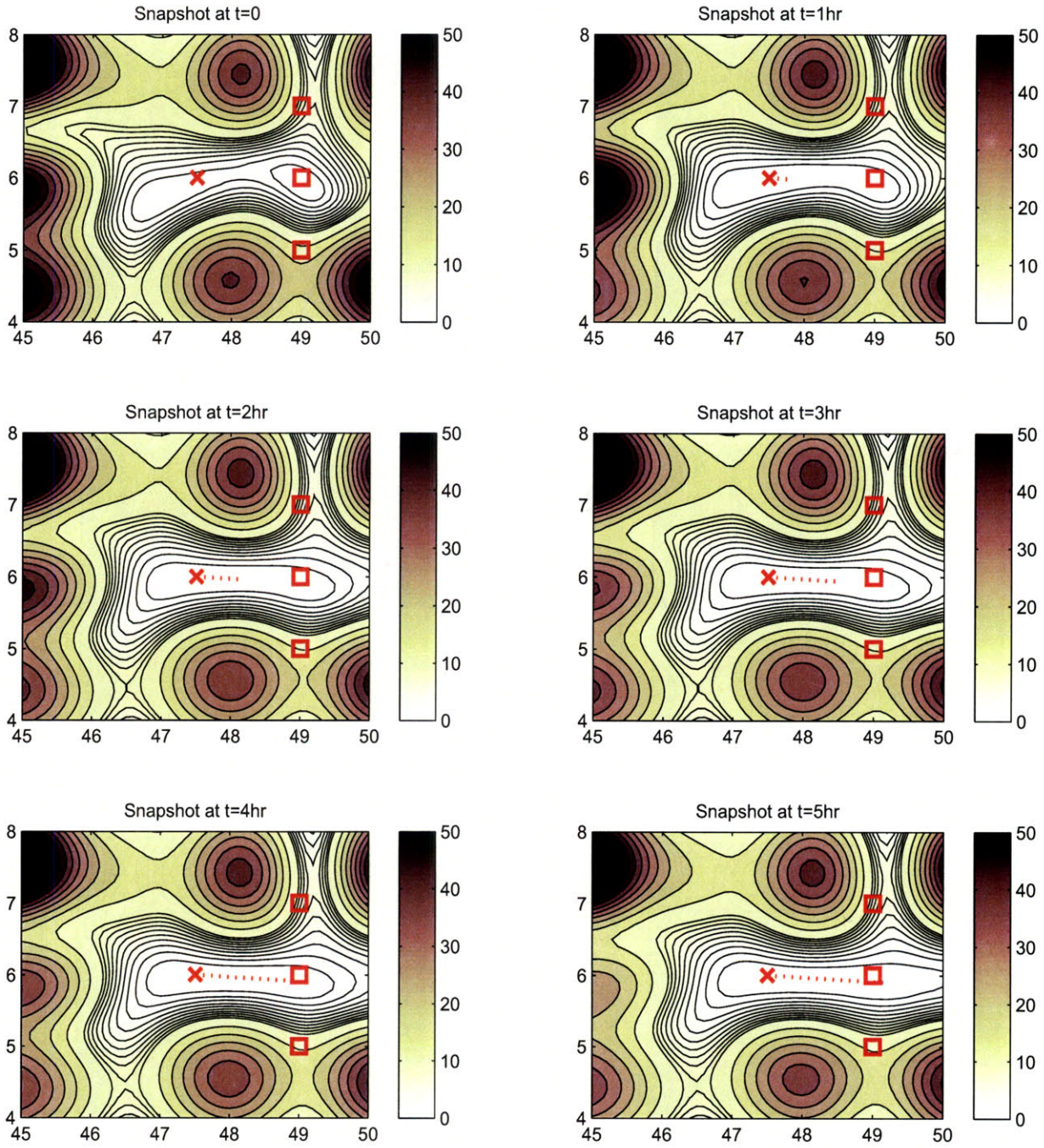

Figure D-4: Snapshots of the worst straight-line trajectory with evolution of information field (Scenario 1)

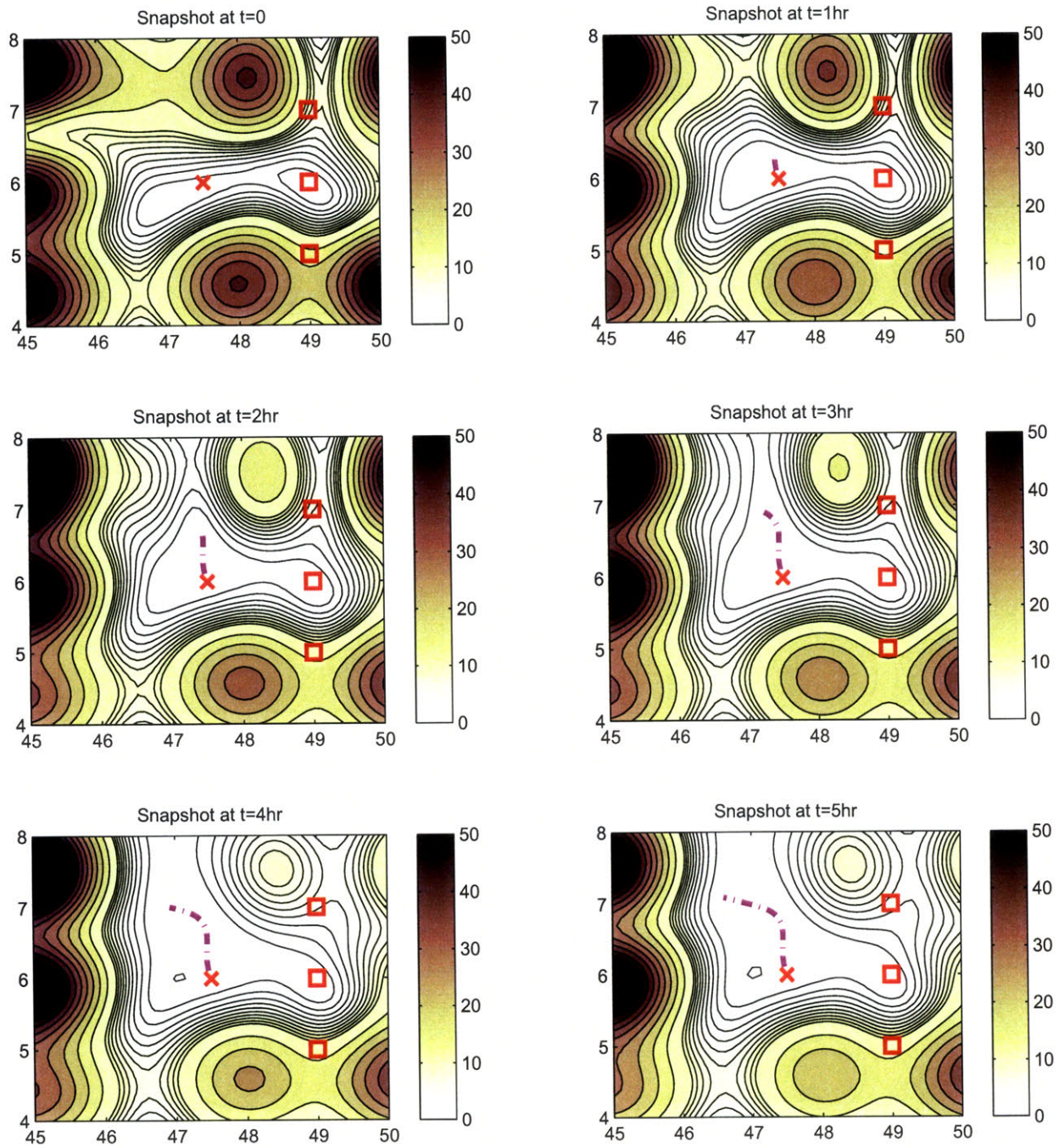

Figure D-5: Snapshots of the myopic optimal trajectory with evolution of smoother-form information field (Scenario 1)

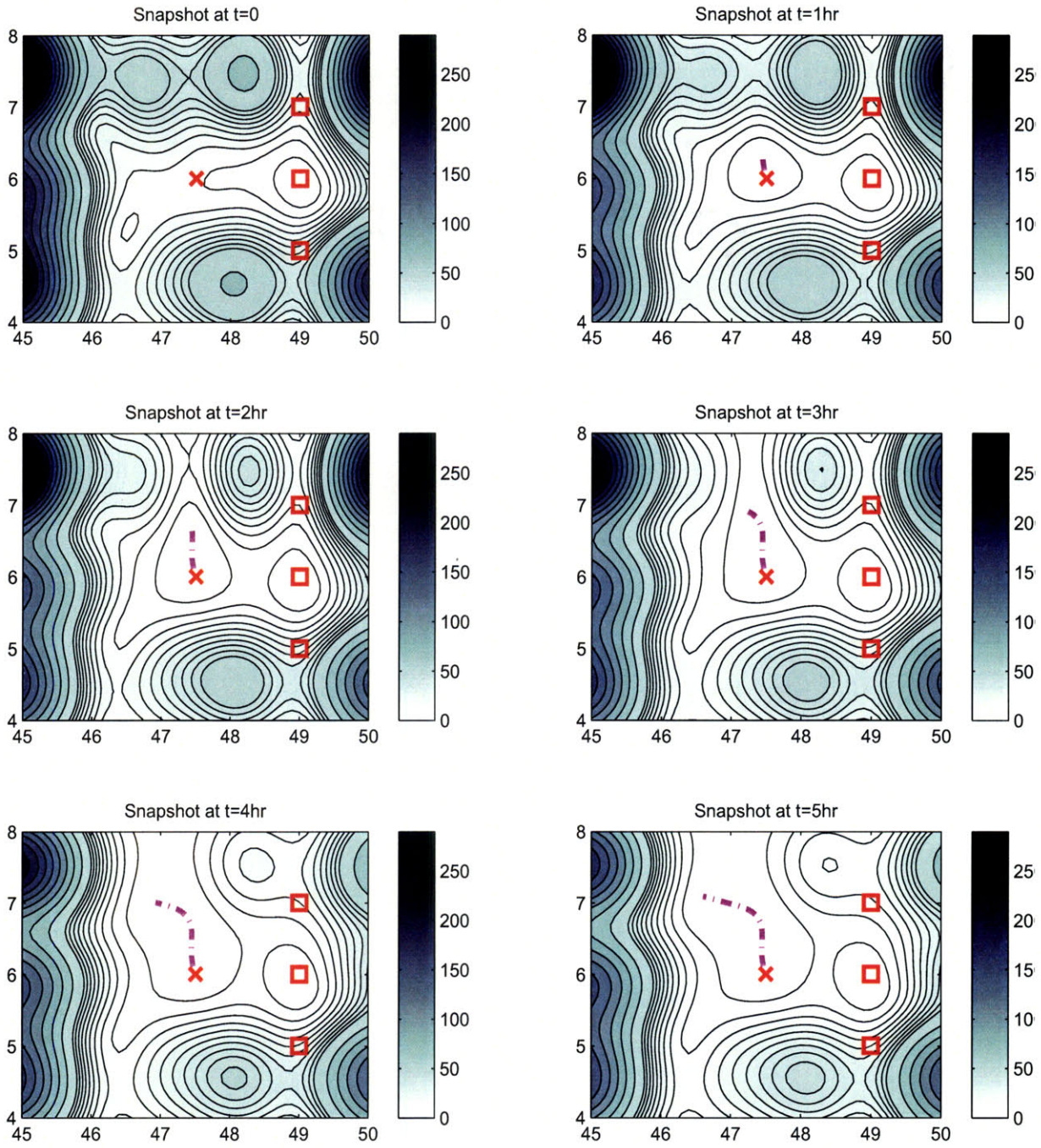

Figure D-6: Snapshots of the myopic optimal trajectory with evolution of filter-form information field (Scenario 1)

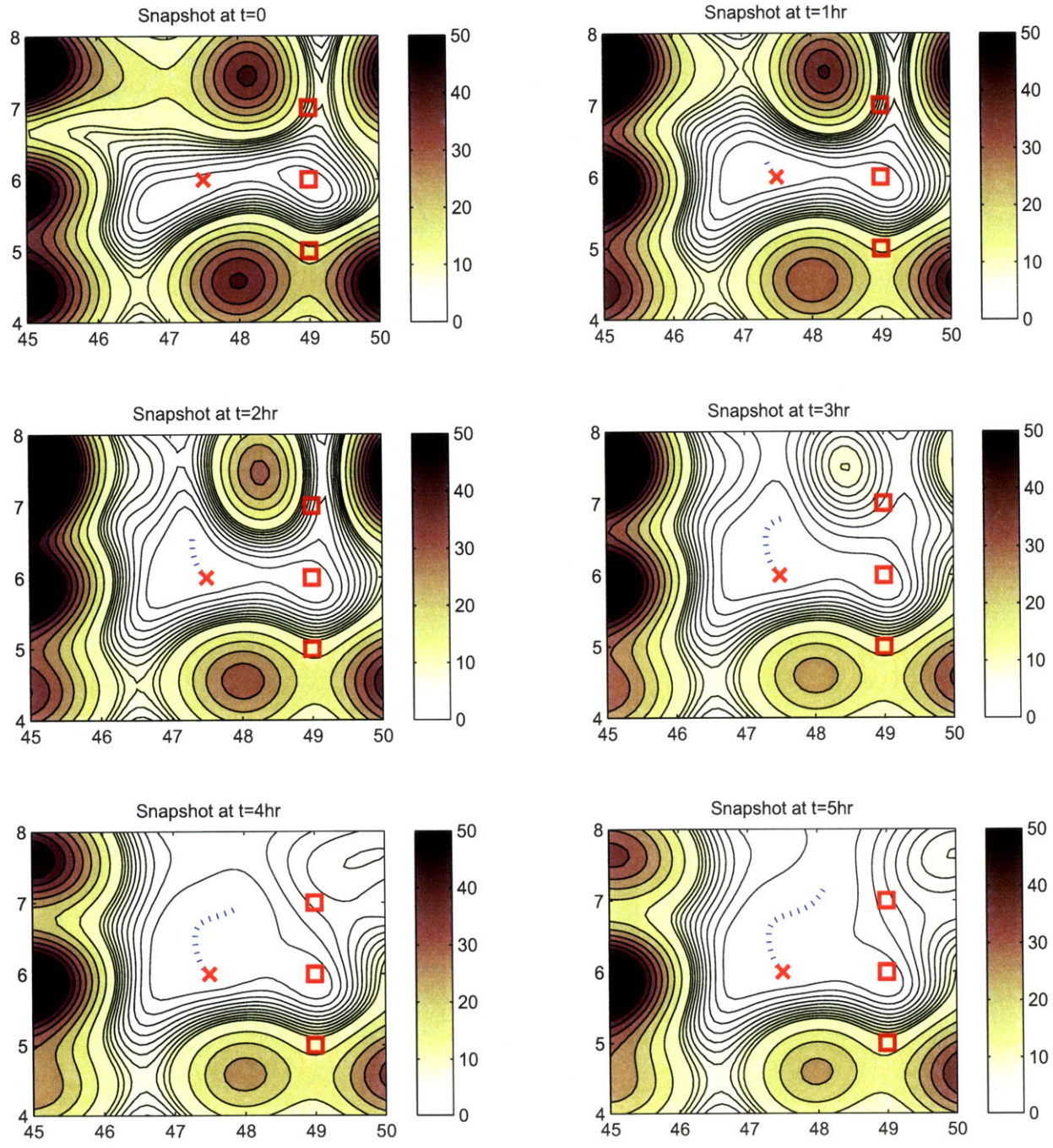

Figure D-7: Snapshots of the trajectory of myopic real-time steering with evolution of smoother-form information field (Scenario 1)

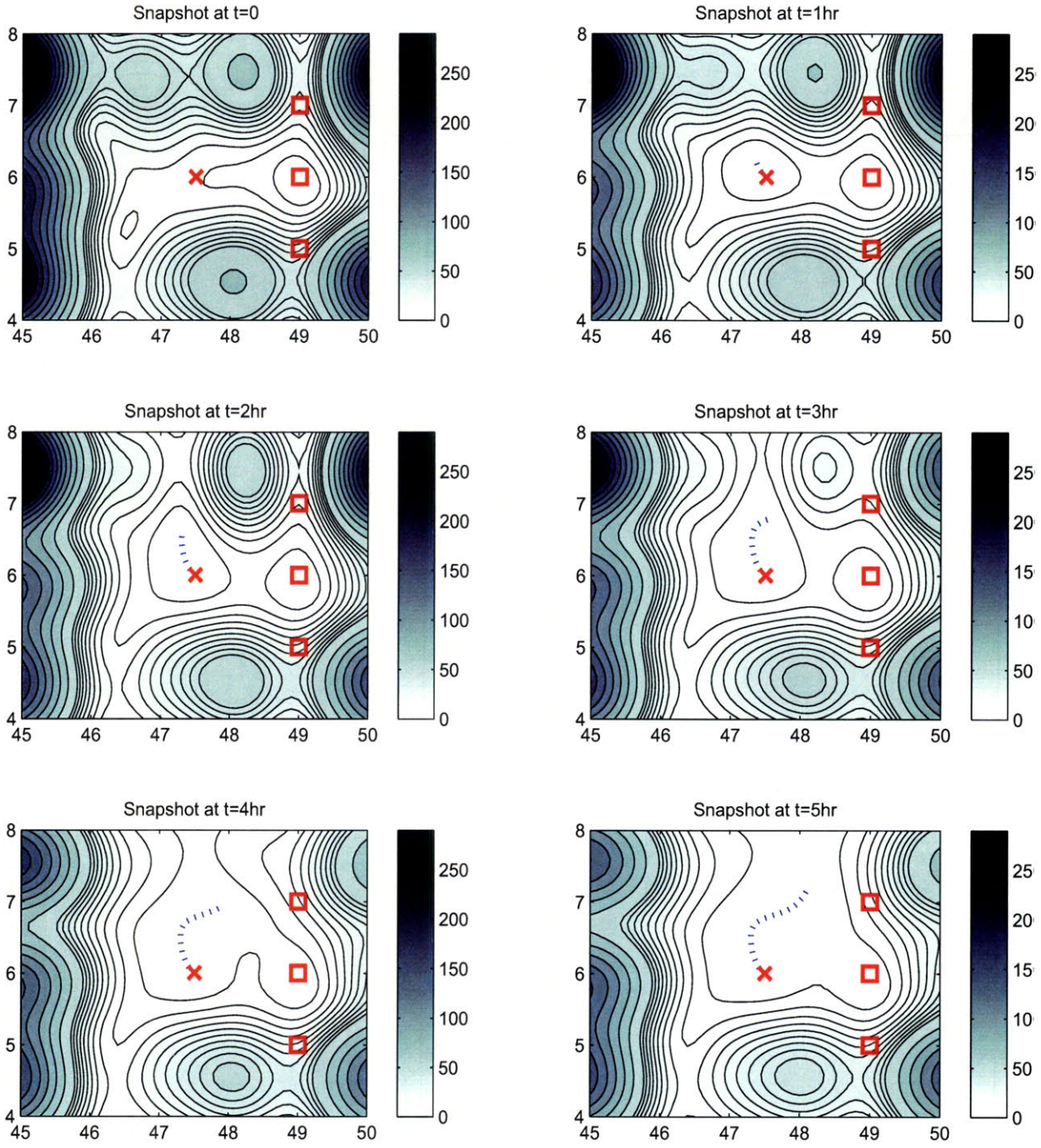

Figure D-8: Snapshots of the trajectory of myopic real-time steering with evolution of filter-form information field (Scenario 1)

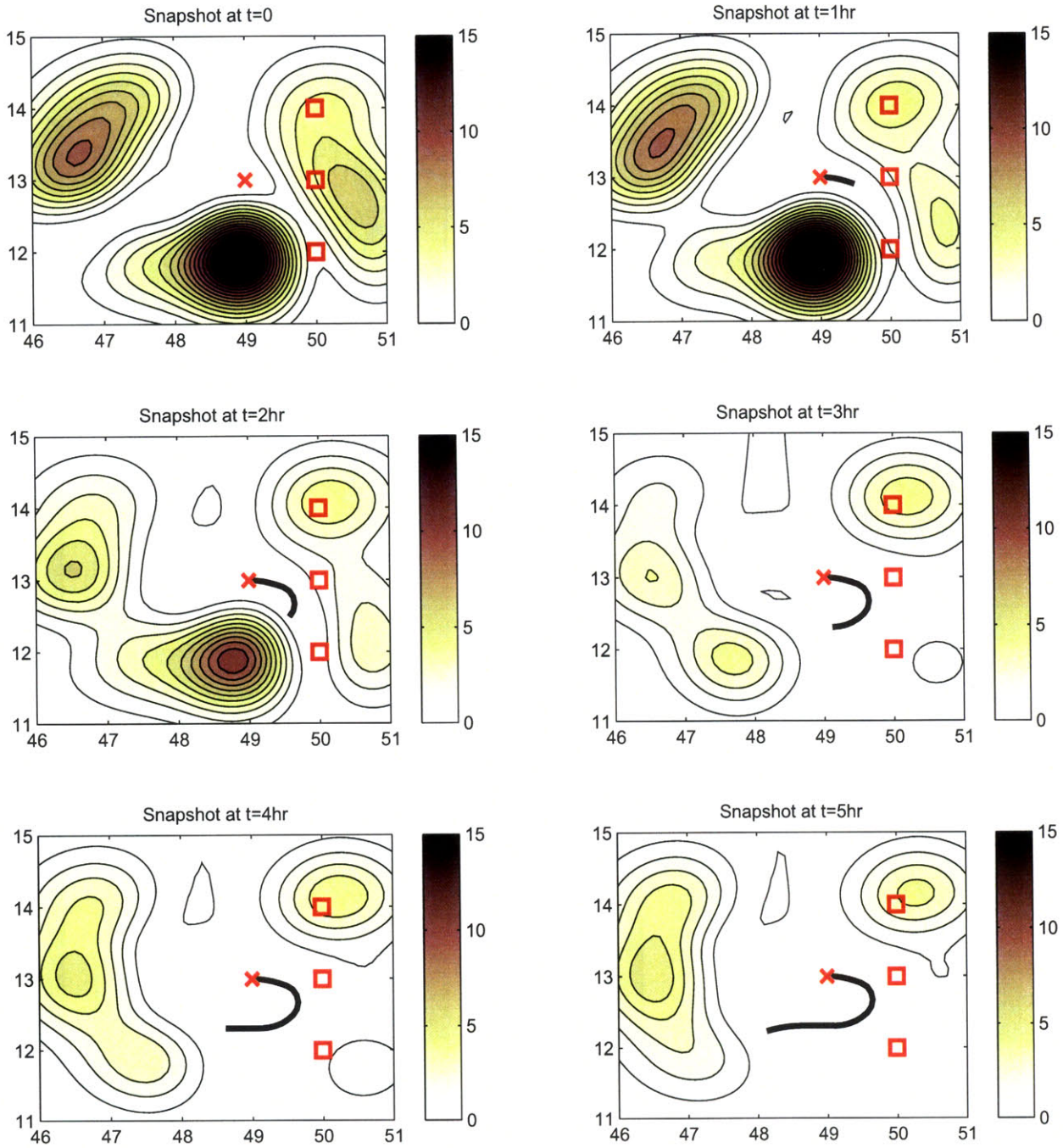

Figure D-9: Snapshots of the optimal trajectory with evolution of information field (Scenario 2)

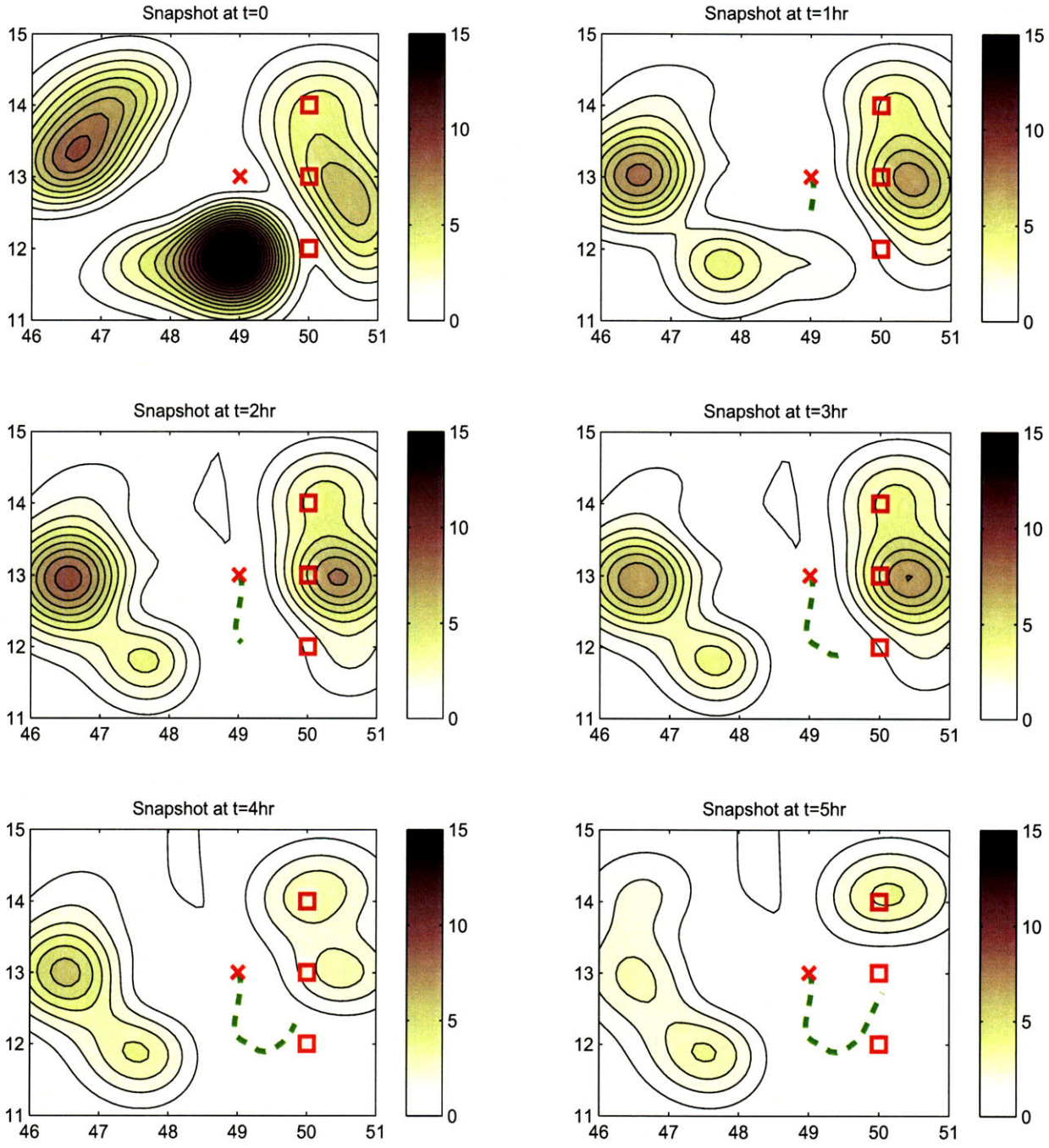

Figure D-10: Snapshots of the trajectory of real-time steering with evolution of information field (Scenario 2)

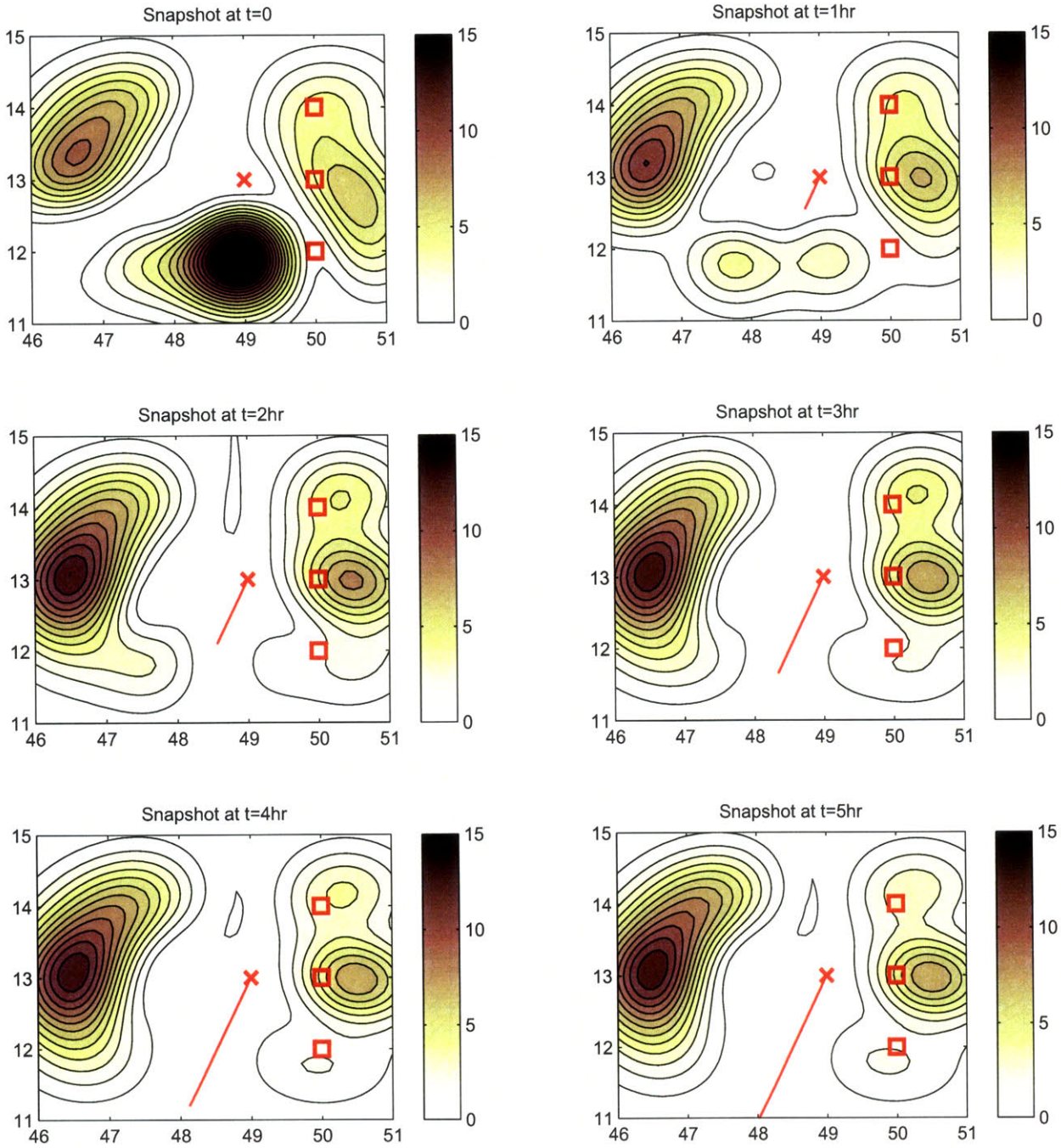

Figure D-11: Snapshots of the best straight-line trajectory with evolution of information field (Scenario 2)

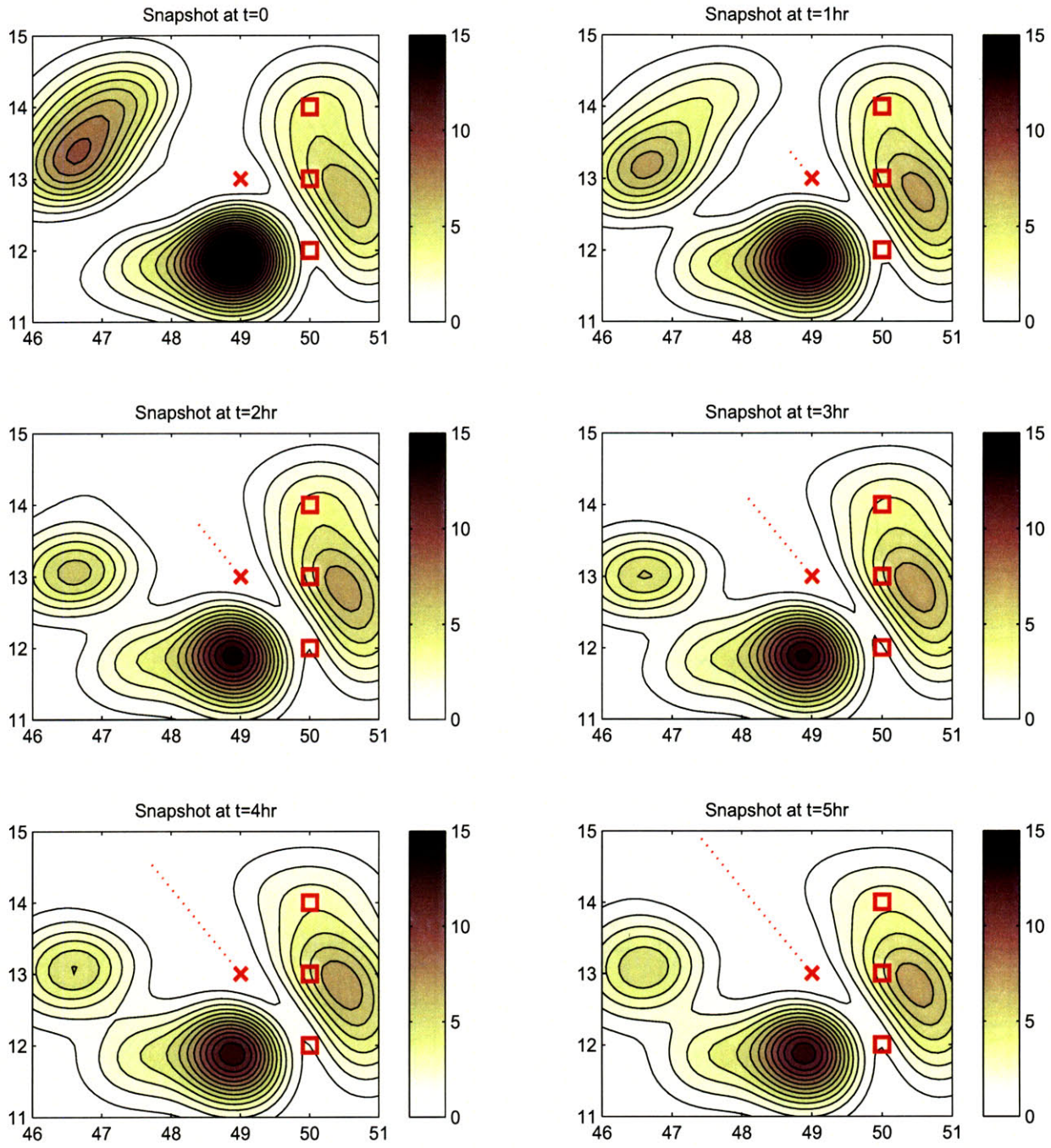

Figure D-12: Snapshots of the worst straight-line trajectory with evolution of information field (Scenario 2)

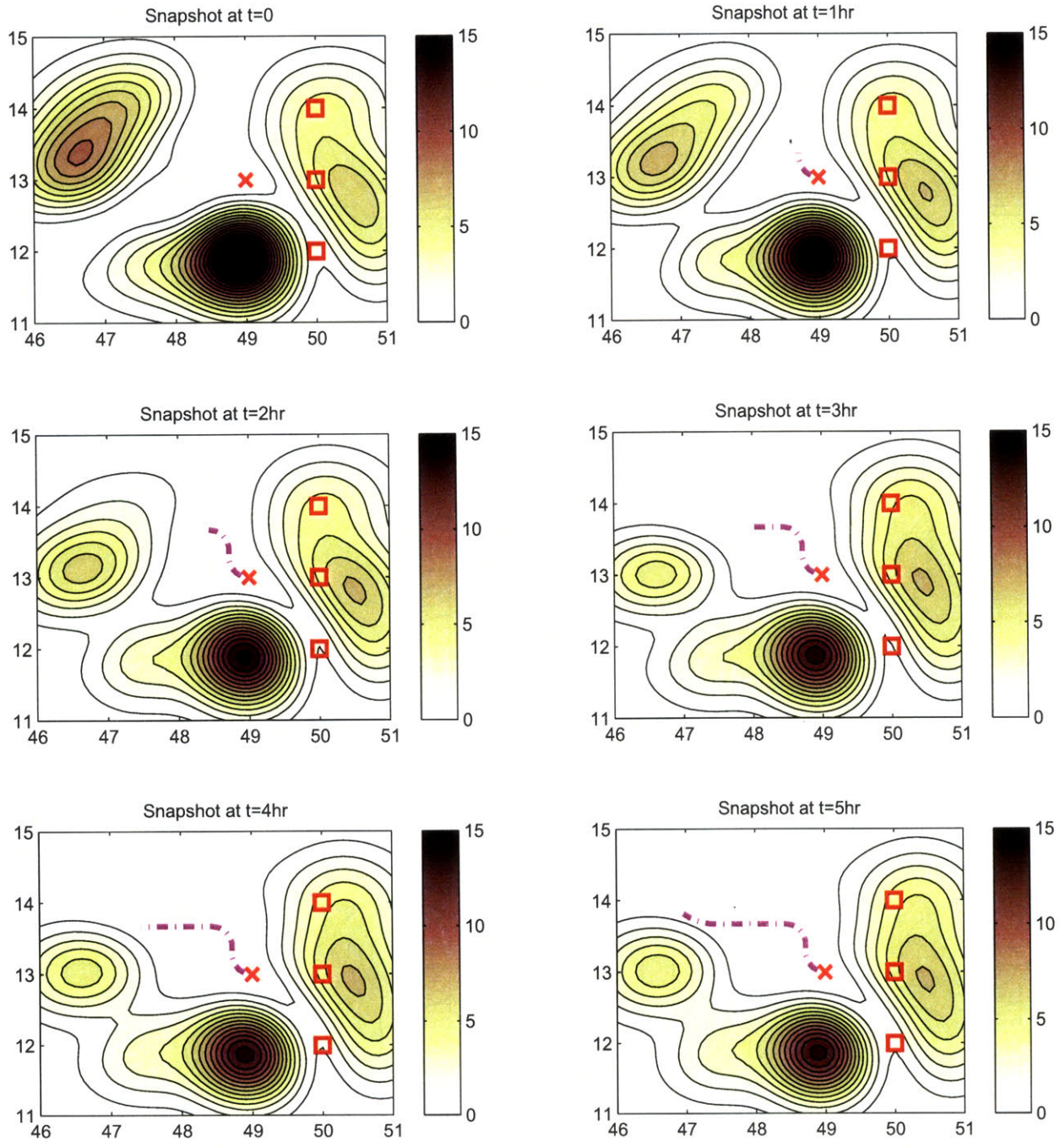

Figure D-13: Snapshots of the myopic optimal trajectory with evolution of smoother-form information field (Scenario 2)

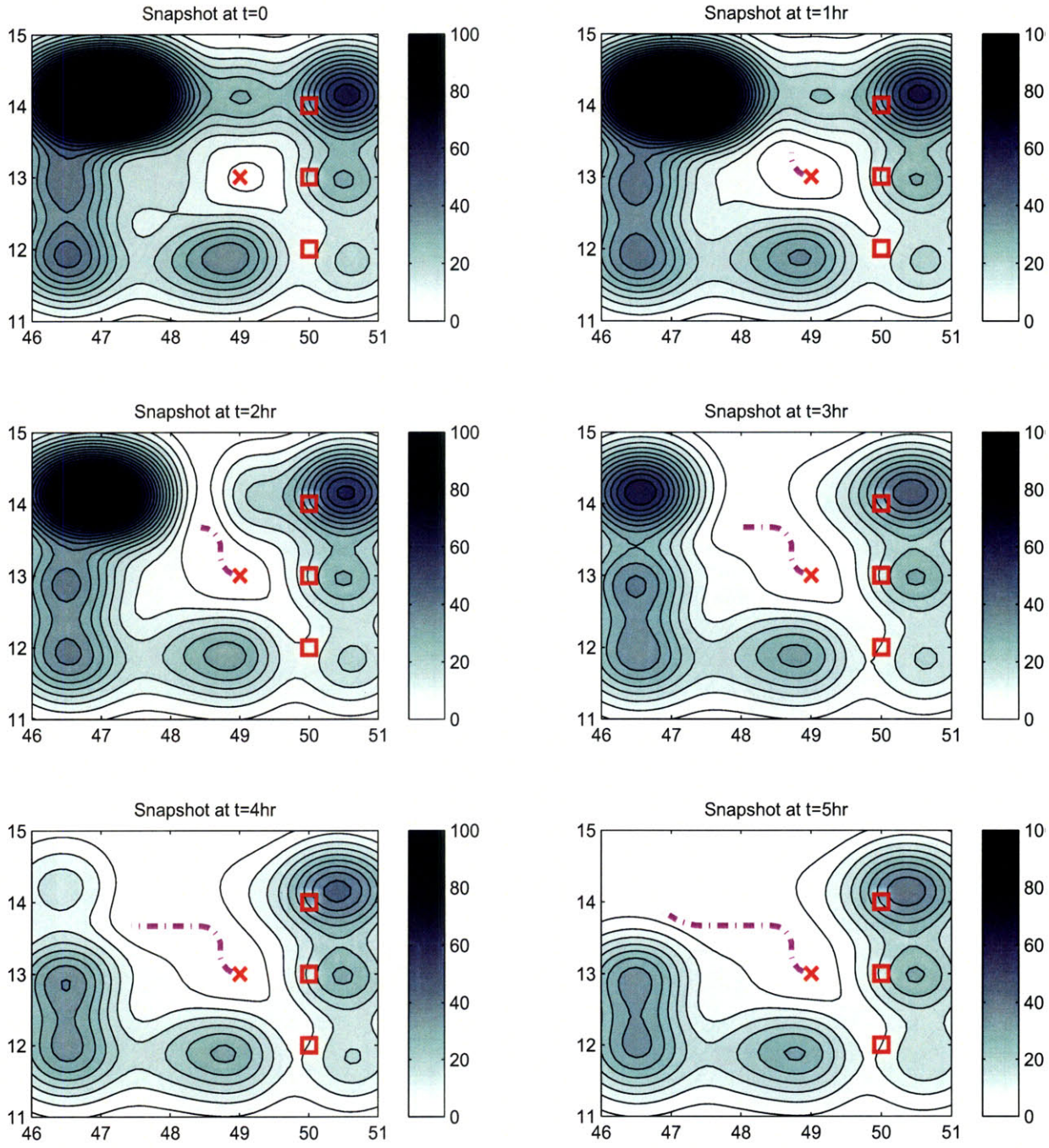

Figure D-14: Snapshots of the myopic optimal trajectory with evolution of filter-form information field (Scenario 2)

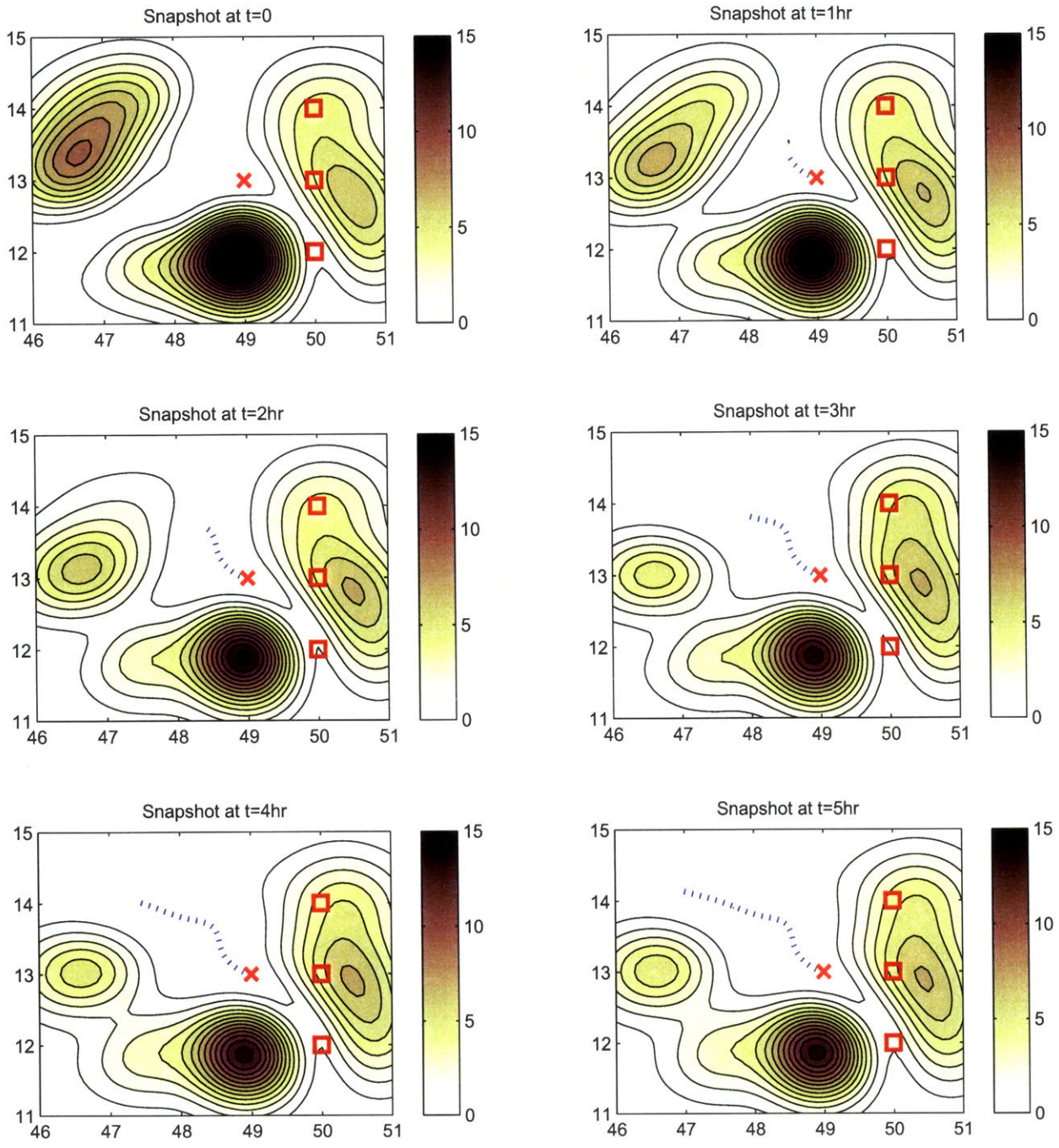

Figure D-15: Snapshots of the trajectory of myopic real-time steering with evolution of smoother-form information field (Scenario 2)

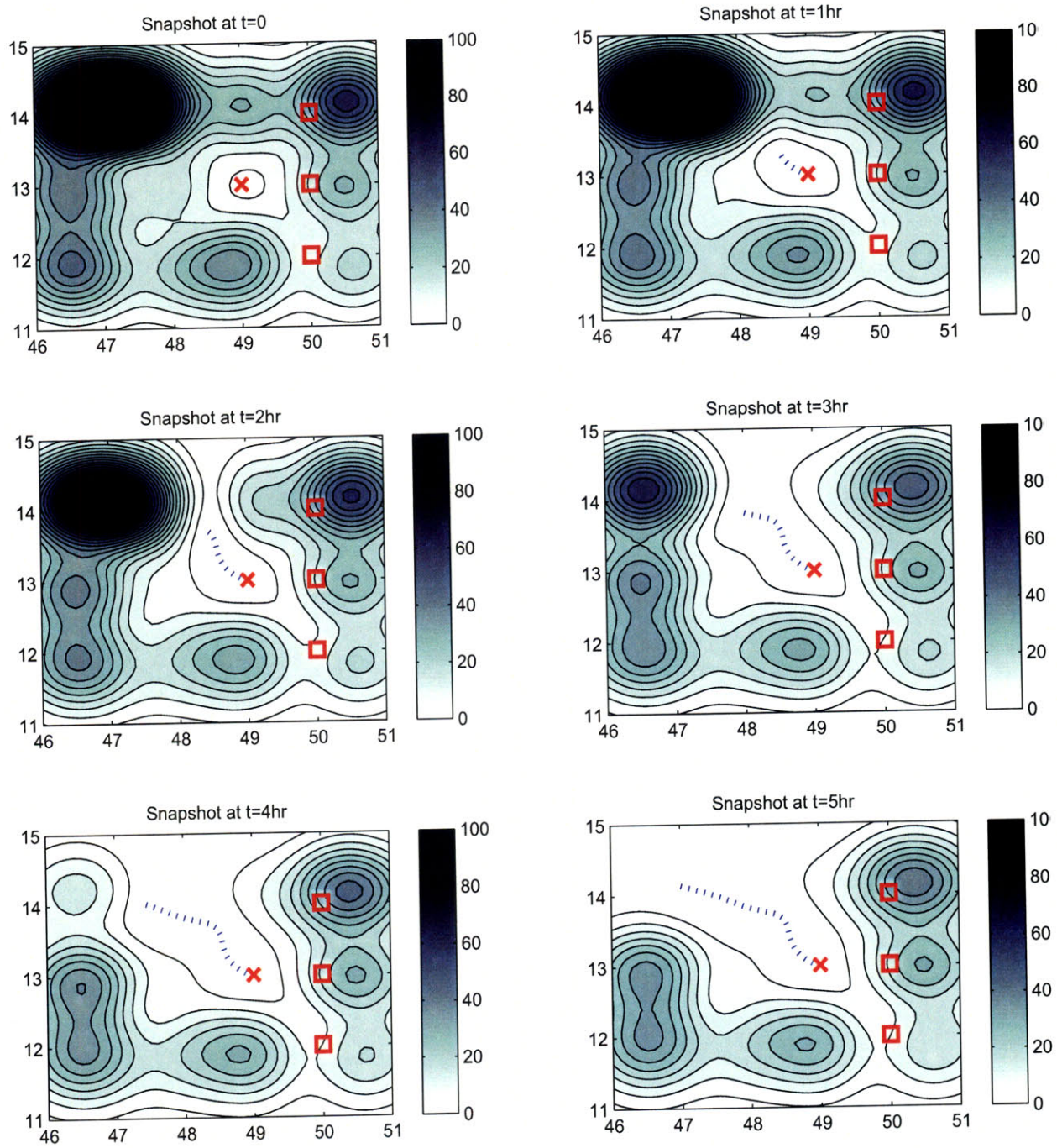

Figure D-16: Snapshots of the trajectory of myopic real-time steering with evolution of filter-form information field (Scenario 2)

Bibliography

[1] T.N. Palmer, R. Gelaro, J. Barkmeijer, and R. Buizza, "Singular vectors, metrics, and adaptive observations," *Journal of the Atmospheric Sciences*, vol. 55, no. 4, pp. 633–653, 1998.

[2] D.N. Daescu and I.M. Navon, "Adaptive observations in the context of 4D-var data assimilation," *Meteorology and Atmospheric Physics*, vol. 85, no. 111, pp. 205–226, 2004.

[3] E.N. Lorenz and K.A. Emanuel, "Optimal sites for supplementary weather observations: simulation with a small model," *Journal of the Atmospheric Sciences*, vol. 55, no. 3, pp. 399–414, 1998.

[4] S.J. Majumdar, C.H. Bishop, B.J. Etherton, and Z. Toth, "Adaptive sampling with the ensemble transform Kalman filter. Part II: Field programming implementation," *Monthly Weather Review*, vol. 130, no. 3, pp. 1356–1369, 2002.

[5] M. Leutbecher, "A reduced rank estimate of forecast error variance changes due to intermittent modifications of the observing network," *Journal of the Atmospheric Sciences*, vol. 60, no. 5, pp. 729–742, 2003.

[6] A. Krause and C. Guestrin, "Near-optimal nonmyopic value of information in graphical models," in *Proc. 21st Conf. on Uncertainty in Artificial Intelligence*, Edinburgh, Scotland, Jul. 2005.

[7] C.-W. Ko, J. Lee, and M. Queyranne, "An Exact Algorithm for Maximum Entropy Sampling," *Operations Research*, vol. 43(4), pp. 1–2, 1995.

[8] C. Guestrin, A. Krause, and A. Singh, "Near-optimal sensor placements in Gaussian processes," in *Proc. 22nd Int. Conf. on Machine Learning*, Bonn, German, Apr. 2005, pp. 265–272.

[9] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Near-optimal sensor placements: maximizing information while minimizing communication cost," in *Proc. 5th Int. Conf. on Information Processing in Sensor Networks*, Nashville, TN, Apr. 2006, pp. 2–10.

[10] J.L. Williams, J.W. Fisher III, and A.S. Willsky, "Approximate dynamic programming for communication-constrained sensor network management," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, Aug. 2007.

[11] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. 2, pp. 235–284, 2008.

[12] H. Wang, K. Yao, G. Pottie, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *Proc. 3rd Int. Symposium on Information Processing in Sensor Networks*, Berkeley, CA, Apr. 2004, pp. 36–45.

[13] E. Ertin, J.W. Fisher III, and L.C. Potter, "Maximum mutual information principle for dynamic sensor query problems," *Lecture Notes in Computer Science: Information Processing in Sensor Networks*, vol. 2634/2003, pp. 2351–2365, 2003.

[14] G.M. Hoffmann, S.L. Waslander, and C.J. Tomlin, "Mutual information methods with particle filters for mobile sensor network control," in *Proc. IEEE 45th Conf. on Decision and Control*, San Diego, CA, Dec 2006, pp. 1019–1024.

[15] K.M. Anstreicher, M. Fampa, J. Lee, and J. Williams, "Maximum-entropy remote sampling," *Discrete Applied Mathematics*, vol. 108, no. 3, pp. 211–226, 2001.

[16] T.E. Duncan, "On the Calculation of Mutual Information," *SIAM Journal on Applied Mathematics*, vol. 19(1), pp. 1–2, 1970.

[17] T.T. Kadota. M. Zakai, and J. Ziv, "Mutual Information of the White Gaussian Channel With and Without Feedback," *IEEE Trans. on Information Theory*, vol. 17(4), pp. 1–2, 1971.

[18] D. Guo, S. Shamai (Shitz), and S. Verdú, "Mutual Information and Minimum Mean-Square Error in Gaussian Channels," *IEEE Trans. on Information Theory*, vol. 51(4), pp. 1–2, 2005.

[19] T.E. Duncan and B. Pasik-Duncan, "Estimation and Mutual Information," in *Proc. IEEE Conference on Decision and Control*, New Orleans, LA, Dec. 2007, pp. 1–2.

[20] Y. Tomita, S. Omatu, and T. Soeda, "An Application of the Information Theory to Filtering Problems," *Information Sciences*, vol. 11, pp. 13–27, 1976.

[21] E. Mayer-Wolf and M. Zakai, "On a Formula Relating the Shannon Information to the Fisher Information for the Filtering Problem," *Lecture Notes in Control and Information Sciences 61*, pp. 164–171, 1984.

[22] S.K. Mitter and N.J. Newton, "Information and Entropy Flow in the Kalman-Bucy Filter," *Journal of Statistical Physics*, vol. 118(112), pp. 1–2, 2005.

[23] N.J. Newton, "Dual Nonlinear Filters and Entropy Production," *SIAM Journal on Control and Optimization*, vol. 45(3), pp. 998–1016, 2006.

[24] ——, "Dual Kalman-Bucy Filters and Interactive Entropy Production," *SIAM Journal on Control and Optimization*, vol. 46(5), pp. 1637–1663, 2007.

[25] S. Martinez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.

[26] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte, "Information-theoretic coordinated control of multiple sensor platforms," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan, Sep. 2003, pp. 1521–1526.

[27] B. Ristic and M. Arulampalam, "Tracking a manoevring target using angle-only measurements: algorithms and performance," *Signal Processing*, vol. 83, no. 6, pp. 1223–1238, 2003.

[28] N. Cressie, "The Origins of Kriging," *Mathematical Geology*, vol. 22(3), pp. 1–2, 1990.

[29] C.K.I. Willams and C.E. Rasmussen, *Gaussian Processes for Regression: Advances in Neural Information Processing Systems 8.* MIT Press, 1996.

[30] R. Buizza and T.N. Palmer, "Impact of Ensemble Size on Ensemble Prediction," *Monthly Weather Review*, vol. 126, pp. 2503–2518, 1998.

[31] T.M. Hamill, J.S. Whitaker, and C. Snyde, "Distance-Dependent Filtering of Background Error Covariance Estimates in an Ensemble Kalman Filter," *Monthly Weather Review*, vol. 129, pp. 2776–2790, 2001.

[32] J.A. Hansen, "Personal communication with Dr. James A. Hansen in Naval Research Laboratory, Monterey, CA," 2008.

[33] B. Grocholsky, "Information-theoretic control of multiple sensor platforms ," Ph.D. dissertation, University of Sydney, 2002.

[34] B. Grocholsky, J. Keller, V. Kumar, and J. Pappas, "Cooperative Air and Ground Surveillance," *IEEE Robotics and Automation Magazine*, vol. 13(3), pp. 16–25, 2006.

[35] T.M. Cover and J.A. Thomas, *Elements of Information Theory.* Wiley Series In Telecommunications, 1991.

[36] J.S. Whitaker and H.M. Hamill, "Ensemble data assimilation without perturbed observations," *Monthly Weather Review*, vol. 130, no. 7, pp. 1913–1924, 2002.

[37] G. Evensen and P.J. van Leeuwen, "Assimilation of altimeter data for the Agulhas current using the ensemble Kalman filter with a quasigeostrophic model," *Monthly Weather Review*, vol. 123, no. 1, pp. 85–96, 1996.

[38] E.N. Lorenz, "Designing Chotic Models," *Journal of Atmospheric Sciences*, vol. 62, pp. 1574–1587, 2005.

[39] M.S. Grewal and A.P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB.* Upper Saddle River, NJ: Prentice-Hall, Inc., 2001.

[40] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition.* MIT Press, 2001.

[41] B.S. Andersen, J.A. Gunnels, F. Gustavson, and J. Waśniewski, "A recursive formulation of the inversin of symmetric positive definite matrices in packed storage data format," *Lecture Notes in Computer Science*, vol. 2367, pp. 287–296, 2002.

[42] LAPACK, " `http://www.netlib.org/lapack`."

[43] S. Burer and J. Lee, "Solving Maximum-Entropy Samping Problems Using Factored Masks," *Mathematical Programming*, vol. 109(2-3), pp. 263–281, 2007.

[44] J.E. Wall Jr., A.S. Willsky, and N.R. Sandell Jr., "On the Fixed-Interval Smoothing problem," *Stochastics*, vol. 5(1&2), pp. 1–41, 1981.

[45] J.B.Lasserre, "A trace inequality for matrix product," *IEEE Trans. on Automatic Control*, vol. 40(8), pp. 1500–1501, 1995.

[46] R.A. Horn and C.R. Johnson, *Matrix Analysis.* Cambridge University Press, 1985.

[47] R. Subramanian and K.V. Bhagwat, "On a theorem of Wigner on products of positive matrices," *Proceedings Mathematical Sciences*, vol. 88A-3(1), pp. 31–34, 1979.

[48] TOMLAB SNOPT, "`http://tomopt.com/tomlab/products/snopt/`."

[49] N. Misra, H. Singh, and E. Demchuck, "Estimation of the Entropy of a Multivariate Normal Distribution," *Journal of Multivariate Analysis*, vol. 92, pp. 324–342, 2005.

[50] A. Genz, "Numerical Computation of Multivariate Normal Probabilities," *J. Comp. Graph. Stat.*, vol. 1, pp. 1–2, 1992.

[51] R.M. Hodur, "The Naval Research Laboratory's Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS)," *Monthly Weather Review*, vol. 125(7), pp. 1414–1430, 1997.

[52] Y. Zhang and Q. Ji, "An Algorithm for the Construction of 'D_k-Optimal' Experimental Designs," *Technometrics*, vol. 16, pp. 203–210, 1974.

[53] M.C. Shewry and H.P. Wynn, "Maximum Entropy Sampling," *Journal of Applied Statistics*, vol. 46, pp. 165–170, 1987.

[54] J.L. Williams, J.W. Fisher III, and A.S. Willskys, "An Approximate Dynamic Programming Approach to a Communication Constrained Sensor Management," in *Proc. 8th Conference of Information Fusion*, 2005, pp. 1–2.

[55] J. Lee, "Constrained Maximum-Entropy Sampling," *Operations Research*, vol. 46(5), pp. 1–2, 1998.

[56] M.C. Bueso, J.M. Angulo, and F.J. Alonso, "A State-Space Model Approach to Optimum Spatial Sampling Design Based on Entropy," *Environmental and Ecological Statistics*, vol. 5, pp. 29–44, 1998.

[57] K.M. Anstreicher, M. Fampa, J. Lee, and J. Williams, "Using Continuous Nonlinear Relaxations to Solve Constrained Maximum-Entropy Sampling Problems," *Mathematical Programming*, vol. 85, pp. 221–240, 1999.

[58] A. Hoffman, J. Lee, and J. Williams, "New Upper Bounds for Maximum-Entropy Sampling," *MODA 6 – Advances in Model-Oriented Design and Analysis*, pp. 143–153, 2001.

[59] J. Lee and J. William, "A Linear Integer Programming Bound for Maximum-Entropy Sampling," *Mathematical Programming, Series B*, vol. 94, pp. 247–256, 2003.

[60] M.A. Duran and I.E. Grossmann, "An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs," *Mathematical Programming*, vol. 36, pp. 307–339, 1986.

[61] R. Fletcher and S. Leyffer, "Solving Mixed Integer Nonlinear Programs by Outer Approximation," *Mathematical Programming*, vol. 666, pp. 327–349, 1994.

[62] P. Kesavan and P.I. Barton, "Decomposition Algorithms for Nonconvex Mixed-Integer Nonlinear Programs," *AIChE Symposium Series*, vol. 96(323), pp. 458–461, 1999.

[63] P. Kesavan, R.J. Allgor, E.P. Gatzke, and P.I. Barton, "Outer Approximation Algorithms for Separable Nonconvex Mixed-Integer Nonlinear Programs," *Mathematical Programming, Series A*, vol. 100, pp. 517–535, 2004.

[64] Cplex, "`http://www.ilog.com/products/cplex/`."

[65] H.-L. Choi, J.P. How, and J.A. Hansen, "Ensemble-Based Adaptive Targeting of Mobile Sensor Networks," in *Proc. American Control Conference*, New York City, NY, Jul. 2007, pp. 1–2.

[66] H.-L. Choi and J.P. How, "A Multi-UAV Targeting Algorithm for Ensemble Forecast Improvement," in *Proc. AIAA Guidance, Navigation, and Control Conference*, Hilton Head, SC, Aug. 2007, pp. 1–2.

[67] S. Boyd and L. Vandeberghe, *Convex Optimization.* Cambridge University Press, 2004.

[68] TOMLAB CPLEX, "`http://tomopt.com/tomlab/products/cplex/`."

[69] TOMLAB BARNLP, " http://tomopt.com/tomlab/products/barnlp/."

[70] D. Bertsekas, *Dynamic Programming and Optimal Control: Vol.I.* Athena Scientific, 2005.

[71] TOMLAB KNITRO, " http://tomopt.com/tomlab/products/knitro/."

[72] K. Abhishek, S. Leyffer, and J.T. Linderoth, "Outer-Approximation-Based Solver for Nonlinear Mixed Integer Programs," *Argonne National Lab Report ANL/MCS-P1374-0306*, 2007.

[73] P. Bonami, T.B. Lorenz, A.R. Conn, G. Cornuéjols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter , "An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs," *IBM Research Report RC23771(W0511-023)*, 2005.