Information-rich Path Planning under General Constraints using Rapidly-exploring Random Trees

by

Daniel S. Levine

B.S. Aerospace Engineering with Information Technology,
Massachusetts Institute of Technology (2008)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2010

Author .....
Department of Aeronautics and Astronautics
May 21, 2010

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

Accepted by .....
Eytan H. Modiano
Associate Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Information-rich Path Planning under General Constraints using Rapidly-exploring Random Trees

by

Daniel S. Levine

Submitted to the Department of Aeronautics and Astronautics
on May 21, 2010, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

This thesis introduces the Information-rich Rapidly-exploring Random Tree (IRRT), an extension of the RRT algorithm that embeds information collection as predicted using Fisher information matrices. The primary contribution of this trajectory generation algorithm is target-based information maximization in general (possibly heavily constrained) environments, with complex vehicle dynamic constraints and sensor limitations, including limited resolution and narrow field-of-view. Extensions of IRRT both for decentralized, multiagent missions and for information-rich planning with multimodal distributions are presented. IRRT is distinguished from previous solution strategies by its computational tractability and general constraint characterization. A progression of simulation results demonstrates that this implementation can generate complex target-tracking behaviors from a simple model of the trade-off between information gathering and goal arrival.

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

Acknowledgments

I offer my sincerest thanks to my advisor, Prof. Jonathan How, who has challenged and put up with me these last five years. His honesty, guidance, insight, and dry wit are indispensable.

I thank my professors at MIT, especially Profs. Emilio Frazzoli, John Deyst, Karen Willcox, and Nick Roy, whose passionate instruction is fundamental to any success I may find. My enjoyment at MIT has been vastly enhanced by having known Profs. Erica Funkhouser, William Corbett, Joe Haldeman, and David Barber, whose poetry and writing are sources of much inspiration.

To my colleagues at the Aerospace Controls Lab, especially Brandon Luders, Sameera Ponda, Frantisek Sobolic, Buddy Michini, Josh Redding, and Brett Bethke, you deserve my deepest gratitude for your advice, camaraderie, and abetment of my vexing sense of humor. I thank Kathryn Fischer and the other department administrators, who hold Aero-Astro together despite the barrage of harmlessly errant students and the byzantine schedules of faculty.

I am fortunate to have such supportive friends, especially Alexa Muller, Melissa Dere, Dane Cohen, Camilla Cohen, the entire Cohen family, Brian Saplicki, Ben Zipken, Ben Gleitzman, Björk, Ante Vulin, George Marzloff, Evan Karlik, Anne Chin, and Allie Anderson.

Finally, I cannot fathom where I would be without the love of my mother, father, and sister, who have always believed that I can be the best geneticist, archaeologist, clergyman, poet, rocket scientist, or whatever nonsense. They have defended me in seemingly indefensible times and have looked after my happiness with the fondest attention. I am forever indebted to them.

This work is funded in part by AFOSR FA9550-08-1-0086 and the MIT Lincoln Laboratory (POC: Dr. Peter Cho).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The use of unmanned systems has been identified as a key factor in increasing the frequency, persistence, safety, and robustness – and decreasing the risk and cost – of intelligence, surveillance, and reconnaissance (ISR) missions [3]. Most currently operational unmanned systems involve human interaction at the teleoperation and path planning levels. However, as the frequency and duration of unmanned operations increases, the need for greater levels of autonomy will have to be realized.

Mobile ISR missions are predicated on information collection via sensor measurements [1, 2]. Accordingly, the quality of individual measurements along a mobile agent's planned trajectory could have a significant impact on the mission-level performance. As sensor measurements are generally subject to noise, and the quality of an individual measurement is a function of both the noise realization and the state at which the measurement is received, the ability to plan paths that best utilize onboard sensing can dramatically improve performance. In the context of ISR missions, it is desirable that plans be both *informative* and *efficient*, whereby a tradeoff between information collection and mission duration is effected. Moreover, planned trajectories must comply with such mission parameters as sensor limitations (e.g., small field-of-view), environmental constraints (e.g., cluttered space), and dynamic constraints (e.g., the collection platform).

This thesis presents a new online algorithm for information-rich path planning. As elucidated in Section 1.3, the distinction between many information-rich planning problem formulations is evident in the constraint characterizations considered. In order to solve problems germane to real-world applications, which consider sensor limitations, environmental constraints, and dynamic constraints, this thesis explores the design and implementation of path planning algorithms that incorporate general constraint characterizations.

1.2 Problem Statement

The general problem addressed in this thesis is as follows: A vehicle with nonlinear dynamics must traverse a bounded, non-convex environment from a start state to a goal state while minimizing the path distance and maximizing the information gain of point features that are stationary in the environment. While the solution proposed is sufficiently abstracted and amenable to this general problem, we shall focus, for demonstrative purposes, on specific target localization problems. The selected vehicles are Dubins cars and quadrotor helicopters, carrying a sensing platform of one camera (i.e., a monocular bearings-only sensor) with a limited field-of-view. The sensor is used to estimate the 3D locations of stationary targets in the environment, a subset of $\mathbb{R}^3$. The target classification problem, in which the existence of a target in the sensor field-of-view is inferred, and the data association problem, in which individual targets are correctly distinguished from a group, are beyond the scope of this thesis.

1.3 Previous Work

In the past several decades, many formulations of the information-rich path planning problem have been considered. While the problem titles are endemic to particular research communities, the problem formulations can be roughly distinguished according to their inherent constraint characterizations. This section briefly reviews

the previous, relevant work in the areas of sensor delivery, sensor placement, heuristic path shape design, analytical solutions, receding horizon optimization, POMDPs, and sample-based methods.

The field of sensor delivery involves problems in which target visitation is either a constraint or is rewarded. Examples of sensor delivery problems include the periodic revisitation of ground targets by a fixed-wing UAV with a downward facing vision sensor [65], and data exfiltration from isolated sensor nodes by a traveling data collector. Vehicle dynamic constraints are often embedded in the problem. However, the sensor model is constrained such that informative paths consist of visitation at some radius and then departure. Klesh *et al.* propose a problem in which a team of UAVs with omni-directional range sensors must receive at least one bit of information from targets whose positions are known [29]. It can be shown that the problem is a form of the Traveling Salesman Problem with vehicle dynamic constraints and visitation radii specified by individual target signal strengths. The optimal placement of sensors for particular parameter estimation problems has also been explored [17, 44, 45]. Sensor placement problems can be considered a limiting case of the general trajectory generation problem for multiple agents. Such sensor placement problems do not incorporate sensor motion constraints but allow for information-theoretic treatment of the (potentially time-varying) parameter estimation process.

Early problem formulations in trajectory design demonstrate the utility of planning *informative* paths that mitigate possible observability loss. Speyer *et al.* illustrate that proportional homing missile control with bearings-only sensing leads to observability loss in the estimation process [64]. By quantifying the trajectory information content using the Fisher information matrix (FIM), the authors eschew the need to predict the estimation covariance in the trajectory design optimization. The metric used in [64] is the trace of the FIM; later works similarly examine information richness in bearings-only sensing problems using the FIM determinant, or approximations thereof, in the cost function [38–40, 46].

While the above problems are solved via numerical optimization, several analytical solutions exist that use the FIM to quantify trajectory information collection in an

optimal control framework. Such solutions seek to maximize, for example, a lower bound on the determinant of the FIM [42] or the log det of the final FIM [49]. Analytic solutions yield optimal paths for very simple problems but are difficult to scale to more complicated scenarios (e.g., dynamics of order higher than two).

Given a prior distribution on the target location, one solution to both the information-rich trajectory generation problem and the persistent surveillance problem involves precomputing and maintaining some heuristic path shape near the target estimate. This class of methods is motivated by the operational observation, in the case of bearings-only sensing, that agents following paths with certain shapes, e.g., circles, ellipses, and spirals, tend to perform well in steady state. Barber *et al.* propose flight path optimization by finding an optimal altitude and radius for a circular trajectory above a stationary target [7]. Rafi *et al.* similarly analyze circular trajectories at a fixed altitude to track a constant velocity target, in the process optimizing the circle radius [52]. While these heuristically constrained trajectories capture the physical and geometric intuition of bearings-only target tracking, that of reducing range and maximizing orthogonality of measurements, the solutions are essentially ad-hoc and naïve with respect to inherent constraints in the problem; for example, the effects of visibility loss or dynamic/environmental infeasiblity are ill-addressed within the heuristic path shape framework.

Recent research has also considered trajectories constructed by performing receding-horizon control online. This class of solution strategies can be partitioned into discrete-space and continuous-space planners, and into single-step and multiple-step planners. The advent of discrete-space, information-rich planners coincided with attempts to solve a core robotics problem, simultaneous localization and mapping (SLAM), in which a vehicle must localize itself using measurements of features registered in a feature map of the environment, which is both constructed and refined online. Several prominent papers have addressed the SLAM-oriented problem of planning a vehicle path through the environment to maximize the information gain in a temporally local sense [10, 13, 66]; these strategies can be summarized as greedy, next-best-view methods that perform steepest ascent on the information value.

Realizing the need for information-theoretic multi-step planners [27], Sim and Roy presented a global planner for the SLAM framework which performs multi-step planning as a pruned graph search [61]. There have since been a number of multiple-step-ahead planners. Ryan *et al.* provide a formulation of cooperative search and track in the multi-step, receding horizon framework [59]. The solution generated therein attempts to minimize the expected conditional entropy. An upper bound on the cooperative information gain based on pairwise mutual information approximations is used to enable decentralized control. This pairwise-node approximation is revisited more rigorously by Hoffman and Tomlin in [24]. Watanabe *et al.* present a stochastically optimized, n-step-ahead guidance design for monocular vision-based UAV control applications [70]. The algorithm minimizes an expected cost that includes the guidance error and control effort expended on sensor manuevers. The derivation provided is specific to the use of a 2D vision sensor and an extended Kalman filter to perform tracking of a single target.

Ristic and Gunatilaka present an algorithm for detection and subsequent information gain-driven control of a sensing agent for the purpose of estimating the parameters of a radiological point source [56]. The control vectors, which are selected via a multiple-step receding horizon maximization of the Fisher information gain, maneuver the observer and limit its exposure to radiation. The source detection and parameter estimation are executed jointly in a particle filter, though until the detection threshold is met, the measurements are taken on a parallel sweeping (colloquially, "lawnmower") search. Ristic *et al.* extend [56] to enable parameter estimation of multiple radioactive sources, the number of which are also estimated [57]. By specifying information gain in terms of the Rényi divergence between the current and future posterior densities, the need for parallel sweeping before detection is obviated. Moreover, the presented extension permits the use of multiple observers whose measurements are reported back to, and whose control vectors are sent from, an assumed centralized data fusion and control center.

Several continuous-space, receding-horizon planning strategies for generating information-rich trajectories have also been considered. Frew uses the determinant of the target

estimation error covariance as the objective function in a trajectory-generating optimization for 2D ground robots with limited field-of-view [16]. Grocholski *et al.* introduce a decentralized, coordinated control algorithm for multiple sensor platforms that uses the log det of the Fisher information matrix as an objective function [19]. Individual agents act locally by performing steepest ascent on the information value, but share only measurement-derived data relevant to the estimation process. Choi adopts this paradigm, quantifying the information reward in terms of continuous-time mutual information for adaptive sampling problems with weather forecasting applications [11]. Ousingsawat and Campbell formulate a receding horizon optimal control problem that attempts to maximize information, quantified using the FIM, while avoiding risk zones and satisfying terminal location and time constraints [47]. However, the results therein are limited to simple constraint sets: an omnidirectional sensor performs 2D target estimation, vehicles are modeled as point masses, and risk zones are elliptical. An extension of this work [48] uses the low-order receding horizon optimization results of [47] to form the heuristic foundation of a centralized task assignment algorithm for multiple agents gathering information about stationary targets. The task assignment, solved using a large mixed-integer linear program (MILP), requires each target to be visited simultaneously by a pair of agents spaced 90° apart on the target's circular risk zone. Though the benefit of an additional sensor in collecting information is apparent, the extrapolation of [47] requiring agent-pair visitations is difficult to justify for all scenarios. In the case of stationary targets, the temporal correlation of measurements between multiple agents is uninformative; therefore, in the sense of efficiency, it is unclear whether two agents momentarily visiting the target 90° apart can gather more information with less overall cost than one agent flying a 90° arc. Additionally, this algorithm does not appear to be robust to changes in the estimated target location due to sensing actions, nor does it account for relevant sensor constraints.

Ponda [51] uses the A-optimality condition of the FIM as the objective function to optimize the trajectory of a fixed-wing aircraft with a perfectly gimballed camera. The selection of a gimballed camera, together with the absence of obstacles, assumes

the target to be visible from the entire flight space. In reality, the existence of local minima in regions of sensor occlusion limit the effectiveness of such a method. While many works embed a small set of apt constraints – Frew considers limited field-of-view sensing limitations, and Ponda explicitly handles vehicle dynamic constraints – receding-horizon optimization strategies are not extensible to the combination of sensor limitations, environmental constraints, and dynamic constraints.

When generality is desired, the Partially Observable Markov Decision Process (POMDP) framework is widely acknowledged to be the most principled way of solving information-rich planning problems. Le Ny and Pappas describe mobile sensor trajectory optimization to improve estimation of a stochastic, multidimensional Gaussian Markov random field [41]. If the measurement process is linear in the estimation states, the Kalman filter can be shown to be optimal, with which the trajectory optimization problem is a deteriministic optimal control problem. The optimal solution is computationally expensive; Le Ny and Pappas propose a suboptimal, non-greedy trajectory optimization scheme based on forward value iteration. Recent research has also considered belief-space planning for both the target tracking problem and its inverse problem, that of localizing a vehicle through sensor measurements of perfectly known targets in a prescribed environment. He *et al.* use the Belief Roadmap (BRM) to plan vehicle trajectories that maximize the self-localization capability of a hovering vehicle operating in GPS-denied environments [21]. Using a prior map of the environment and the associated measurement samples for a laser range finder, a graph of the covariance propagation between samples can be formed, from which the BRM efficiently selects trajectories that mitigate egomotion drift and aid knowledge of goal arrival. Roy and He use so-called "semi-conditional" planning as a forward-search in the POMDP framework to facilitate target tracking [20, 58]. While the POMDP framework has shown promising results for simple vehicle models, POMDP solutions are currently intractable for vehicle models with complex dynamics.

Finally, we note the work of Kwak and Scerri, which uses a priority queue to expand high-reward nodes from a tree structure, but is generally restricted to overhead sensing of a precomputed 2D cost map [36].

Recall the motivation for this thesis, for which dynamic, sensing, and environmental constraints must be satisfied while performing localization on stationary targets. Previous research has used solution strategies that are either not amenable to the whole of these constraints or, by adopting very general constraint characterizations, are rendered intractable for use, for example, on vehicles with complex dynamic models. This thesis, whose contributions are described in the following section, addresses such a shortcoming in the literature, fulfilling the need for an online motion planner that, while sacrificing optimal performance, is extensible to general constraint characterizations.

1.4 Thesis Contributions

This thesis introduces the Information-rich Rapidly-exploring Random Tree (IRRT), an extension of the RRT algorithm [37] that embeds information collection as predicted using Fisher information matrices [14]. A further extension of IRRT for multi-agent missions is also presented. The primary contribution of this trajectory generation algorithm is target-based information maximization in arbitrary (possibly heavily constrained) environments, with complex vehicle dynamic constraints and sensor limitations, specifically, limited resolution and narrow field-of-view. As IRRT is a sample-based planner, feasible solutions can be easily generated in real-time, and the planner effectiveness scales with the available computational resources. Simulated results have demonstrated that IRRT can produce complex target-tracking behaviors from a simple model that trades off information gathering and goal arrival. The flexibility to plan informative trajectories under general cost functions and feasibility constraints distinguishes the presented solution strategy for planning information-rich trajectories from the previous research.

The structure of this thesis is as follows. Chapter 2 briefly reviews measures of information gain for stochastic systems. Chapter 3 motivates the selection of closed-loop Rapidly-exploring Random Trees as the baseline planning algorithm upon which IRRT is built; the chapter continues with the algorithmic development of IRRT.

Chapter 4 presents a progression of simulation results demonstrating the utility of IRRT in constrained, information-rich planning problems. An extension of IRRT to multi-agent, decentralized scenarios is developed in Chapter 5, and simulation results are presented. A further extension of IRRT that permits multimodal prior and posterior distributions is developed and demonstrated in Chapter 6. Summary analysis is offered, and future work suggested, in Chapter 7.

Chapter 2

Information-Theoretic Measures

In this chapter, several well-studied information-theoretic measures [12] used to quantify the value of observation sequences are briefly reviewed. Entropic information and divergence measures are both applicable to general Bayesian estimation processes, the latter having rich connections to information geometry. Despite these strengths, entropic information and divergence measures both require suitable approximations of the posterior distributions, a computational burden that can become intractable several just a few timesteps into the future. An alternative measure is then described in the Fisher information framework, with several key results that make Fisher information an attractive metric for quantifying the estimation uncertainty reduction in plans of suitably long duration.

2.1 Bayesian Decisions

Before proceeding to the discussion of information measures, some basic concepts of Bayesian decision theory are reviewed. Consider the random variable X, with realization $\mathbf{x}$, that describes an uncertain state drawn from the alphabet $\mathcal{X}$. Prior to any observations, the knowledge or belief about $\mathbf{x}$ is captured entirely by the *prior* distribution $p_X(\cdot)$. Observations, which can also be modeled as random variables Z with realizations $\mathbf{z}$, can be used to update the belief by forming the *posterior*

distribution $p_{X|Z}(\cdot|\cdot)$ according to Bayes Rule, i.e.,

$$p_{X|Z}(\mathbf{x}|\mathbf{z}) = \frac{p_{Z|X}(\mathbf{z}|\mathbf{x})p_X(\mathbf{x})}{\sum_{\mathbf{a}\in\mathcal{X}} p_{Z|X}(\mathbf{z}|\mathbf{a})p_X(\mathbf{a})} \tag{2.1}$$

in the discrete case and

$$p_{X|Z}(\mathbf{x}|\mathbf{z}) = \frac{p_{Z|X}(\mathbf{z}|\mathbf{x})p_X(\mathbf{x})}{\int_{\mathcal{X}} p_{Z|X}(\mathbf{z}|\mathbf{a})p_X(\mathbf{a})\ d\mathbf{a}} \tag{2.2}$$

in the continuous case, where the *likelihood* distribution $p_{Z|X}(\cdot|\cdot)$ is derived from the given observation model. Based on these distributions, a Bayes engine generates a decision $\delta(\cdot)$, referred to as "hard" if the result is an estimate $\hat{\mathbf{x}}(\cdot)$ of $\mathbf{x}$ and "soft" if the result is a distibution $q(\cdot)$ that describes the relative likelihood of different elements of $\mathcal{X}$ based on the observed data. This section will proceed by focusing on soft decisions. Ideally, the Bayes decision engine would produce the distribution

$$q(\mathbf{a}) = \begin{cases} 1 & \mathbf{a} = \mathbf{x} \\ 0 & \text{otherwise,} \end{cases} \tag{2.3}$$

which would identify the realized $\mathbf{x}$ with certainty. However, as $\mathbf{x}$ is unknown, it is not generally possible to construct such a $q(\cdot)$ from only the provided data. The Bayes decision instead minimizes the conditional expectation over realizations $\mathbf{x}$ of some cost criterion $C(\mathbf{x}, \delta(\cdot))$ given the data $\mathbf{z}$. As many cost criteria exist, the selection of an appropriate cost criterion is a natural question. Towards this end, several properties of cost functions are introduced for the discrete case; the continuous case is analogous and merely involves replacing the summations with the appropriate integrals.

Definition 1. *A cost function* $C(\cdot,\cdot)$ *is* proper *if*

$$p_{X|Z}(\cdot|\mathbf{z}) = \underset{\{q(\cdot):\sum_{\mathbf{a}} q(\mathbf{a})=1\}}{\operatorname{argmin}} \mathbb{E}\left[C(\mathbf{x}, q)|Z = \mathbf{z}\right] \text{ for all } \mathbf{z}. \tag{2.4}$$

Definition 2. *A cost function* $C(\cdot,\cdot)$ *is* local *if there exists a function* $\phi : \mathcal{X}\times\mathcal{R} \mapsto \mathcal{R}$ *such that* $C(\mathbf{x}, q) = \phi(\mathbf{x}, q(\mathbf{x}))$ *for all* $\mathbf{x}$.

Proper cost functions yield the true posterior belief $p_{X|Z}$. Local cost functions assess the quality of the estimated belief q only in terms of the probability assigned to the actual outcome.

Consider the log-loss criterion

$$C(\mathbf{x}, q) = -A \log q(\mathbf{x}) + B(\mathbf{x}), \tag{2.5}$$

where $A > 0$ and $B(\cdot)$ is arbitrary. It is clear that through the use of the log function, the log-loss criterion emphasizes distributions $q(\cdot)$ that are "peaky," in the sense of being heavily concentrated within portions of their respective supports. While many alternative cost criteria may be considered, it has been shown that when the alphabet $\mathcal{X}$ consists of at least three values ($|\mathcal{X}| \geq 3$), then the log-loss is the only local, proper cost function [9]. Because $B(\cdot)$ is for any realized $\mathbf{x}$ a constant in the cost function, and A is a relative weighting term thereof, we may proceed in describing the Bayesian information measures assuming, without loss of generality, that $A = 1$ and $B(\cdot) = 0$.

2.2 Entropic Information

As the log-loss cost criterion is proper, the expectation of the cost achieves $p_{X|Z}$. In the absence of observations, $p_{X|Z} = p_X$, the so-called prior cost is

$$\begin{aligned} \min_{q(\cdot):\sum_a q(\mathbf{a})=1} \mathbb{E}\left[C(X, q)\right] &= \mathbb{E}\left[C(X, p_X)\right] \\ &= -\mathbb{E}\left[\log p_X(\mathbf{x})\right] \\ &= -\sum_{\mathbf{a}\in\mathcal{X}} p_X(\mathbf{a}) \log p_X(\mathbf{a}) \\ &\triangleq H(X), \end{aligned} \tag{2.6}$$

where $H(X)$ is called the *entropy* [60], or self-information, of X. Entropy is a measure of the average randomness or uncertainty in X, with "peaky" distributions having lower entropy than more uniform distributions. When base-2 logarithms are used in the computation, the units of entropy are bits, and the entropy conveys the number

of bits necessary to communicate X.

Upon observing $Z = \mathbf{z}$, the posterior probability distribution minimizes the log-loss criterion, with resulting cost

$$\begin{aligned}\min_{q(\cdot):\sum_{\mathbf{a}} q(\mathbf{a})=1} \mathbb{E}\left[C(X,q)|Z=\mathbf{z}\right] &= \mathbb{E}\left[C(X,p_{X|Z})|Z=\mathbf{z}\right] \\ &= -\mathbb{E}\left[\log p_{X|Z}(\mathbf{x}|\mathbf{z})|Z=\mathbf{z}\right] \\ &= -\sum_{\mathbf{a}\in\mathcal{X}} p_{X|Z}(\mathbf{x}|\mathbf{z}) \log p_{X|Z}(\mathbf{x}|\mathbf{z}) \\ &\triangleq H(X|Z=\mathbf{z}). \end{aligned} \tag{2.7}$$

Taking the expectation of (2.7) over the set of possible observations yields the average posterior cost

$$\begin{aligned}\mathbb{E}\left[C(X,p_{X|Z})\right] &= \mathbb{E}\left[\mathbb{E}\left[C(X,p_{X|Z}(X|Z))|Z=\mathbf{z}\right]\right] \\ &= -\sum_{\mathbf{z}} p_Z(\mathbf{z}) H(X|Z=\mathbf{z}) \\ &= -\sum_{\mathbf{a},\mathbf{b}} p_{X,Z}(\mathbf{a},\mathbf{b}) \log p_{X|Z}(\mathbf{a}|\mathbf{b}) \\ &\triangleq H(X|Z), \end{aligned} \tag{2.8}$$

where $H(X|Z)$ is called the conditional entropy of X given Z. The results above can easily be extended for k observations by forming the set $Z^K = \{\mathbf{z}_k, \mathbf{z}_{k-1}, ..., \mathbf{z}_1\}$.

The log-loss cost reduction associated with processing observation $\mathbf{z}$ is exactly the difference between the prior and posterior costs

$$\Delta\mathbb{E}\left[C(X,q)\right] = H(X) - H(X|Z) \triangleq I(X;Z), \tag{2.9}$$

which is referred to as the *mutual information* between X and Z. The mutual information can be equivalently expressed as

$$I(X;Z) = \sum_{\mathbf{a},\mathbf{b}} p_{X,Z}(\mathbf{a},\mathbf{b}) \log \frac{p_{X,Z}(\mathbf{a},\mathbf{b})}{p_X(\mathbf{a}) p_Z(\mathbf{b})}. \tag{2.10}$$

It is straightforward to verify that

$$0 \leq H(X|Z) \leq H(X), \tag{2.11}$$

the implication being that conditioning never increases uncertainty as measured by entropy. From this result, the nonnegativity of mutual information is also implied. Moreover, mutual information is a symmetric measure, i.e.,

$$I(X;Z) = H(X) - H(X|Z) = H(Z) - H(Z|X) = I(Z;X). \tag{2.12}$$

2.3 Divergence Measures

Divergence measures are one way to quantify the difference between two probability distributions. For example, in the event that the true belief $p(\cdot)$ cannot be implemented, and one must approximate the belief as $q(\cdot)$, the approximation loss can be characterized as

$$\begin{aligned} \Delta\mathbb{E}\left[C(X,q)\right] &= -\mathbb{E}_p\left[\log q(X)\right] + \mathbb{E}_p\left[\log p(X)\right] \\ &= \mathbb{E}_p\left[\frac{p(X)}{q(X)}\right] = D_{KL}(p||q), \end{aligned} \tag{2.13}$$

where

$$D_{KL}(p||q) \triangleq \sum_{\mathbf{a}\in\mathcal{X}} p(\mathbf{a}) \log \frac{p(\mathbf{a})}{q(\mathbf{a})} \tag{2.14}$$

is called the Kullback-Leibler (KL) divergence of $q(\cdot)$ from $p(\cdot)$ [22]. It is alternatively referred to as the *information divergence* of $q(\cdot)$ from $p(\cdot)$ or the *relative entropy* of $q(\cdot)$ with respect to $p(\cdot)$. Despite often being referred to in the sense of a distance metric, the KL divergence is a *non-symmetric* measure of the difference between two probability distributions, and $D(p||q) \neq D(q||p)$ in general. There exist many relevant connections between KL divergence (relative entropy) and the entropic information

measures; of chief import is the property

$$I(X;Z) = D_{KL}(p_{X,Z}||p_X p_Z) \tag{2.15}$$

$$= \mathbb{E}_{p_X}\left[D_{KL}\left(p_{Z|X}||p_Z\right)\right] \tag{2.16}$$

$$= \mathbb{E}_{p_Z}\left[D_{KL}\left(p_{X|Z}||p_X\right)\right], \tag{2.17}$$

which explains mutual information as an expectation of the KL divergence.

One generalization of the KL divergence is the Rényi divergence [54], defined as

$$D_\alpha(p||q) = \frac{1}{\alpha - 1} \ln \sum p^\alpha(\mathbf{x}) q^{1-\alpha}(\mathbf{x}), \tag{2.18}$$

where $\alpha \geq 0$ is a parameter that determines how much one emphasizes the tails of the distributions $p(\cdot)$ and $q(\cdot)$ in the metric. The selection of the parameter α serves as an additional degree of freedom over the KL divergence, which can be recovered from the Rényi divergence in the limit as $\alpha \to 1$. Moreover, the selection of $\alpha = 0.5$, which corresponds to the Hellinger affinity [50], has been reported to outperform the KL divergence in scenarios where the minor differences in the distribution tails must be stressed [22, 23].

In the sensor management literature, divergence measures are used to quantify the information gain between the prior distribution and some posterior distribution following an observation sequence [22, 31, 32, 57]. The generality of divergence measures, with their rich connection to information theory [12], recommends their use in difficult hybrid estimation problems. However, computation of the required posterior distributions becomes prohibitive as the estimation horizon increases past several timesteps.

2.4 Fisher Information

This section introduces Fisher information and reviews several key properties that make it a suitable metric for information collection. Whereas in previous sections a

soft decision in the form of a probability distribution $q(\cdot)$ was generated by a Bayes decision device, we begin this section by characterizing hard decisions in the form of an estimator $\hat{\mathbf{x}}(\cdot)$ of the realized $\mathbf{x}$. Although Fisher information was first defined in the non-Bayesian framework,[1] the review hereafter proceeds with the continuous Bayesian form.

In formulating the estimation problem, attention must be restricted to the class of estimators whose evaluations depend only on the data and not on $\mathbf{x}$, which is unknown; such estimators are called *valid*. One must further require that the estimate be *unbiased*, i.e.,

$$b_{\hat{\mathbf{x}}}(\mathbf{x}) \triangleq \mathbb{E}\left[\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x}\right] = 0, \quad \forall \mathbf{x} \in \mathcal{X}. \tag{2.19}$$

Estimators satisfying both requirements are said to be *admissible*.

General statements can be made about the performance of the entire class of admissible estimators. One such statement involves the well-known Cramér-Rao bound, which, when it exists, gives a lower bound on the covariance of any admissible estimator $\hat{\mathbf{x}}$ for $\mathbf{x}$. It is the basis of the information inequality

$$P = \mathbb{E}_{\mathbf{x}}\left\{\left[\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x}\right]\left[\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x}\right]^T\right\} \geq J(\mathbf{x})^{-1}, \tag{2.20}$$

where P is the estimation error covariance matrix (for unbiased $\hat{\mathbf{x}}(\cdot)$) and $J(\mathbf{x})$ is called the Fisher information matrix (FIM), its inverse $J(\mathbf{x})^{-1}$ being the matrix Cramér-Rao Lower Bound (CRLB). In the multivariable case, the FIM may be computed as

$$J(\mathbf{x}) = \mathbb{E}\left\{\left[\nabla_{\mathbf{x}} \log p(\mathbf{x}, \mathbf{z})\right]\left[\nabla_{\mathbf{x}} \log p(\mathbf{x}, \mathbf{z})\right]^T\right\} \tag{2.21}$$

$$= \mathbb{E}\left\{-\nabla_{\mathbf{x}}\left[\nabla_{\mathbf{x}} \log p(\mathbf{x}, \mathbf{z})\right]^T\right\}, \tag{2.22}$$

where $\nabla_{\mathbf{x}}$ is the gradient operator with respect to $\mathbf{x}$, and the second equality holds

[1] It is unnatural in some estimation problems to consider a prior distribution $p_X(\cdot)$. In such cases, $\mathbf{x}$ is modeled as an unknown, non-random parameter.

in general only if the regularity condition

$$\mathbb{E}\left[\nabla_{\mathbf{x}} \log p(\mathbf{x}, \mathbf{z})\right] = \mathbf{0} \tag{2.23}$$

is satisfied.

The CRLB holds only if some weak regularity assumptions on $p(\mathbf{x}, \mathbf{z})$ are satisfied and the Fisher information can be computed. It is clear from (2.21)-(2.22) that the Fisher information cannot be computed in all problems, for example, when the density $p(\mathbf{x}, \mathbf{z})$ is not strictly positive for all $\mathbf{x}$ and $\mathbf{z}$. The Fisher information can also be interpreted as a measure of curvature: it measures, on average, the "peakiness" of $\log p(\mathbf{x}, \mathbf{z})$ as a function of $\mathbf{x}$. This is most apparent in the scalar form of (2.22),

$$J(x) = -\mathbb{E}_x \left\{ \frac{\partial^2}{\partial x^2} \log p(x, \mathbf{z}) \right\}. \tag{2.24}$$

As such, the larger $J(\mathbf{x})$ is (in a matrix norm sense), the better one expects to be able to resolve the value of $\mathbf{x} \in \mathcal{X}$ from the observations, hence, the smaller one would expect P to be.

When an estimator satisfies the Cramér-Rao bound with equality, it is called *efficient* and must be the (unique) minimum-variance, unbiased (MVU) estimator. The converse is, however, not true: even when the CRLB exists, it is sometimes not possible to meet the bound for any $\mathbf{x}$, let alone all $\mathbf{x}$.

The calculation of the FIM according to (2.22) presents an implementation challenge whereby the number of computations necessary to form the FIM increases as each new measurement is processed. What is instead sought is a recursive FIM update law which utilizes previously computed FIMs and whose computational demand does not increase with the number of measurements taken. Tichavsky *et al.* present such a method, and its key results are reviewed here [55, 68].

Let $\hat{\mathbf{x}}_k^{\oplus}$ and $P_k^{\oplus}$ denote the estimate and covariance matrix at the kth time step after processing measurement $\mathbf{z}_k$. The information inequality (2.20) can now be written

as

$$P_k^{\oplus} = \mathbb{E}\left\{\left[\hat{\mathbf{x}}_k^{\oplus} - \mathbf{x}\right]\left[\hat{\mathbf{x}}_k^{\oplus} - \mathbf{x}\right]^T\right\} \geq J_k^{-1}, \tag{2.25}$$

where J_k is the FIM computed at the kth time step. The recursive FIM update relationship introduced in [68] takes the form

$$J_{k+1} = D_k^{22} - D_k^{21}\left(J_k + D_k^{11}\right)^{-1} D_k^{12}, \tag{2.26}$$

where

$$\begin{aligned}
D_k^{11} &= -\mathbb{E}\left\{\nabla_{\mathbf{x}_k}\left[\nabla_{\mathbf{x}_k}\log p(\mathbf{x}_{k+1}|\mathbf{x}_k\right]^T\right\}, \\
D_k^{21} &= -\mathbb{E}\left\{\nabla_{\mathbf{x}_k}\left[\nabla_{\mathbf{x}_{k+1}}\log p(\mathbf{x}_{k+1}|\mathbf{x}_k\right]^T\right\}, \\
D_k^{12} &= -\mathbb{E}\left\{\nabla_{\mathbf{x}_{k+1}}\left[\nabla_{\mathbf{x}_k}\log p(\mathbf{x}_{k+1}|\mathbf{x}_k\right]^T\right\} = \left[D_k^{21}\right]^T, \\
D_k^{22} &= -\mathbb{E}\left\{\nabla_{\mathbf{x}_{k+1}}\left[\nabla_{\mathbf{x}_{k+1}}\log p(\mathbf{x}_{k+1}|\mathbf{x}_k\right]^T\right\}, \\
&\quad - \mathbb{E}\left\{\nabla_{\mathbf{x}_{k+1}}\left[\nabla_{\mathbf{x}_{k+1}}\log p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}\right]^T\right\}.
\end{aligned}$$

Certain models afford considerable simplification of the above D_k^{ij} quantities. Consider the nonlinear-Gaussian system model

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, Q_k), \tag{2.27}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \qquad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, R_k), \tag{2.28}$$

where $\mathbf{f}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$ are the nonlinear process and measurement models, respectively, and $\mathbf{w}_k$ and $\mathbf{v}_k$ are uncorrelated, zero-mean Gaussian white sequences with covariances Q_k and R_k, respectively. Defining $F_k \triangleq \nabla_{\mathbf{x}_k}\mathbf{f}(\mathbf{x}_k)$ and $H_k \triangleq \nabla_{\mathbf{x}_k}\mathbf{h}(\mathbf{x}_k)$ as the Jacobians with respect to the estimation state of the process and observation models,

it can be shown that the D_k^{ij} matrices are simply

$$D_k^{11} = F_k^T Q_k^{-1} F_k, \tag{2.29}$$

$$D_k^{12} = -F_k^T Q_k^{-1}, \tag{2.30}$$

$$D_k^{21} = -Q_k^{-1} F_k, \tag{2.31}$$

$$D_k^{22} = Q_k^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1}. \tag{2.32}$$

Substitution into (2.26) and application of a matrix inversion lemma yield the recursive equation

$$J_{k+1} = \left(Q_k + F_k J_k^{-1} F_k^T\right)^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1}. \tag{2.33}$$

To initialize the recursion (2.33), one evaluates the FIM using (2.21) and the prior distribution $p(\mathbf{x}_0)$, i.e.,

$$J_0 = \mathbb{E}\left\{\left[\nabla_{\mathbf{x}_0} \log p(\mathbf{x}_0)\right]\left[\nabla_{\mathbf{x}_0} \log p(\mathbf{x}_0)\right]^T\right\}. \tag{2.34}$$

If the prior is Gaussian with mean $\overline{\mathbf{x}}$ and covariance P_0, the above reduces to

$$J_0 = P_0^{-1}. \tag{2.35}$$

Recall that Fisher information, with its connection to the CRLB, applies to any admissible estimator and provides performance bounds on the estimation process. Therefore, Fisher information is a suitable objective function for optimization solutions that attempts to improve estimation performance. The relative ease with which the information content of temporally distant measurements can be quantified further recommends the FIM as a metric in proactive, real-time planning algorithms. Fisher information will be revisited in Section 3.4.1 in the context of quantifying the information content of measurement sequences taken along planned trajectories.

Chapter 3

Information-rich RRT

This chapter details the algorithmic development of the Information-rich Rapidly-exploring Random Tree (IRRT), an extension of the closed-loop RRT [15, 34] that uses the Fisher Information framework for quantifying trajectory information content. The selection of closed-loop RRT as a baseline is motivated by the successive utility of sample-based planning methods (Section 3.1), of Rapidly-exploring Random Trees (Section 3.2), and finally of closed-loop RRT (Section 3.3). The IRRT algorithm, as presented in Section 3.4, preserves the beneficial properties of closed-loop RRT while allowing the planner to be both cognizant of and proactive toward information collection.

3.1 Sample-Based Planning Methods

Trajectory planning algorithms are an integral facet of mobile autonomous agents. Over the last several decades, a myriad of path planning algorithms has been proposed to varying effect. Many surveys, for example [18], summarize the notable entries in the field. There does not appear to be a general path planning algorithm that performs well for all problems. In fact, many planning algorithms adopted in practice are highly specialized to the particular problem addressed. What is sought is a class of planning algorithms for which extensibility to multiple problem types and tractability are balanced.

Currently, the predominant path planning algorithms for robotic systems can be classified as roadmap, cell decomposition, potential field, or sample-based methods; they differ primarily in their desciptions of the free state space. Roadmap methods, by fitting a graph to the state space, reduce the problem to a graph search. Similarly, cell decomposition methods seek to partition the free space into convex polyhedrons in a graph, which is subsequently searched. While roadmap and cell decomposition methods are viable approaches in low-dimensional configurations spaces, robotic systems often live in high-dimensional configuration spaces with many points of actuation or degrees of freedom. Extensions of roadmap or cell decomposition methods to arbitrary dimensions are not generally tractable. Potential field-based methods, which use weighted potential functions to impart a simulated force on the vehicle, are easily extensible to arbitrary dimensions and are computationally lightweight. However, the tuning of the individual potential functions is not intuitive, and, despite the efforts of randomization algorithms such as the Randomized Path Planner [8], the existence of local minima remains a persistent issue in the potential field approach.

Sample-based methods approximate the connectivity of the free space $\mathcal{X}_{free}$ by sampling configurations in $\mathcal{X}_{free}$ and attempting various edge connections that are unobstructed. The quality of the $\mathcal{X}_{free}$ connectivity approximation scales with the available computational resources; unlike optimization- or mathematical programming-based methods, there is not a fixed time cost associated with generating a feasible solution. Therefore, sample-based methods afford a beneficial tradeoff between the extensibility to high-dimensional planning problems and the associated computational intensity. A well-studied, sample-based planning algorithm called the Probabilistic Roadmap (PRM) [28] enables planning in high-dimensional configuration spaces, such as in linked manipulator or cellular docking applications.

3.2 Rapidly-exploring Random Trees (RRTs)

Thus far, solution generation in arbitrary dimensions has been a stated requirement. As the intention of this thesis is to develop information-rich planning algorithms for

mobile autonomous agents, particularly aerial, ground, and underwater vehicles, we further require the planner to be capable of generating *dynamically feasible* trajectories. To this end, the framework of the Rapidly-exploring Random Tree (RRT) is pursued. First introduced by LaValle in [37], the RRT is noted for being well suited for high-dimensional planning problems involving nonholonomic constraints in nonlinear dynamical systems.

The fundamental operation in the standard RRT algorithm [37] is the incremental growth of a tree of dynamically feasible trajectories, rooted at the system's current state, through simulations of the system's prediction model. A node's likelihood of being selected to grow the tree is proportional to its Voronoi region for a uniform sampling distribution, yielding a natural bias toward rapid exploration of the state space. Because the path cost and constraint evaluations are performed trajectory-wise, the RRT algorithm can easily handle complex constraints that may cause optimization-based approaches to become computationally intractable [37]. Finally, as a sampling-based algorithm, the RRT planner performance scales with the available computational resources, avoiding the exponential growth in computational complexity often found in information-based planning approaches.

3.3 Closed-loop RRT (CL-RRT)

This section reviews the real-time closed-loop RRT (CL-RRT) algorithm, proposed by Frazzoli [15] and derived in detail by Kuwata *et al.* [33–35]. The CL-RRT algorithm adds a path-tracking control loop in the system's RRT prediction model, such that RRT sampling takes place in the reference input space rather than in the vehicle input space. If the system executes a chosen path using the same prediction model, any deviations are propagated using the same closed-loop dynamics, resulting in more accurate trajectory tracking than with open-loop prediction [43]. The algorithm runs in real-time, continuously growing a tree of feasible trajectories. At the end of each phase of tree growth, the best feasible trajectory is selected for execution, and the process repeats. The two primary operations of the algorithm, tree expansion and

Algorithm 1 CL-RRT, TREE EXPANSION

1: Take a sample x_{samp} from the environment
2: Identify the nearest node N_{near} using mixture of EXPLORATION and OPTIMIZATION heuristics
3: $\overline{x}(t+k) \leftarrow$ final state of N_{near}
4: **while** $\overline{x}(t+k) \in \mathcal{X}_{free}$ **and** $\overline{x}(t+k)$ has not reached x_{samp} **do**
5: Use reference law to generate $\overline{r}(t+k)$
6: Use control law to generate $\overline{u}(t+k)$
7: Use prediction model to simulate $\overline{x}(t+k+1)$
8: $k \leftarrow k+1$
9: **end while**
10: **for** each feasible node N generated **do**
11: Update cost estimates for N
12: Add N to $\mathcal{T}$
13: **end for**

Algorithm 2 CL-RRT, EXECUTION LOOP

1: $t \leftarrow 0$
2: Initialize tree $\mathcal{T}$ with node at $x(0)$
3: **while** $x(t) \neq \mathbf{x}_{goal}$ **do**
4: Update the current state $x(t)$
5: Propagate the state $x(t)$ by $\Delta t \rightarrow \overline{x}(t+\Delta t)$
6: **while** time remaining for this timestep **do**
7: CL-RRT, TREE EXPANSION
8: **end while**
9: Use cost estimates to identify best feasible path $\mathcal{P}_* \leftarrow \{N_{root}, \ldots, N_{target}\}$
10: Apply best feasible path, if one exists
11: $t \leftarrow t + \Delta t$
12: **end while**

the execution loop, are reviewed next; more detailed treatments of these algorithms have been considered in recent papers[34, 43].

3.3.1 Tree Expansion

The tree expansion algorithm, which attempts to add one or more nodes to the tree $\mathcal{T}$, is described in Algorithm 1. Similar to the basic RRT algorithm [37], a sample x_{samp} is generated in a metric space $\mathcal{X}$ (line 1), and the node N_{near} in the tree $\mathcal{T}$ that is "nearest" by some metric (c.f. Section 3.4.2) is identified (line 2). A forward simulation is then generated for this node, beginning at the final state of the parent node, N_{near} (line 3), until the trajectory has become infeasible or has reached the sample x_{samp} (line 4).

In traditional open-loop RRT, one or more candidate input sequences $\overline{u}(t)$ may be generated in the forward simulation to yield a terminal state near x_{samp}. In closed-

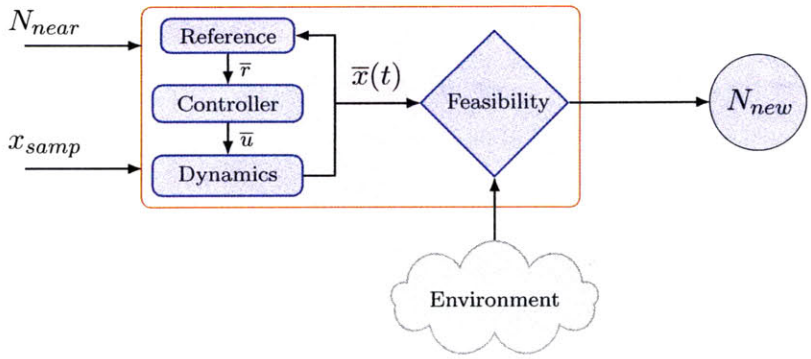


Figure 3-1: Block diagram of CL-RRT.

loop RRT, a closed-loop representation of the vehicle dynamics is instead used to generate trajectories. Consider the block diagram of CL-RRT given in Figure 3-1. The inputs to the prediction model are x_{samp} and N_{near}, which are compared to form a simple reference trajectory $\overline{r}(t)$ (line 5), then passed through a virtual vehicle controller to form the input sequence $\overline{u}(t)$ (line 6). As in the standard (open-loop) RRT, vehicle dynamic constraints are automatically satisfied by forward simulation, in this case forming the closed-loop state response sequence $\overline{x}(t)$ (line 7). Environmental constraints can then be queried for $\overline{x}(t)$ to determine whether the trajectory is feasible. From this forward simulation, one or more feasible nodes N_{new} may be generated. After computing the cost estimates for these nodes (line 11), which are used in the execution loop, the new nodes are added to $\mathcal{T}$.

3.3.2 Execution Loop

For environments which are dynamic and uncertain, the RRT tree must keep growing during the execution cycle to account for changes in the situational awareness [15]. Furthermore, given the extensive computations involved to construct the tree, as much of the tree should be retained as possible, especially in real-time applications [67]. Algorithm 2 shows how the algorithm executes some portion of the tree while continuing to grow it.

The planner updates the current best path to be executed by the system every

Δt seconds. During each cycle, the current state is updated (line 4) and propagated to the end of the planning cycle (line 5), yielding $\overline{x}(t + \Delta t)$. The tree root is set to the node whose trajectory the propagated state is following; this node's trajectory is committed and must be followed. The remaining time in the cycle is used to expand the tree (lines 6–8). Following this tree growth, the cost estimates are used [15] to select the best feasible[1] path in the tree (line 9); the nature of these cost estimates is discussed further in Section 3.4.3. Assuming at least one such feasible path exists[2], it is selected and executed by the vehicle (line 10).

3.4 Information-rich RRT (IRRT)

An extension of the CL-RRT algorithm that enables information-rich path planning is now described. By determining the anticipated measurement sequence along paths in the tree, the path information contribution can be quantified in the Fisher information framework. The embedding of information metrics in the tree allows for nearest node heuristics and cost functions that explicitly consider information in both the tree growth and execution phases. This section details the information quantification subroutine and introduces the information-based nearest node heuristic and cost function.

3.4.1 Information Quantification

In this section, the single-agent Fisher information quantification J_N of node N is developed. For scenarios where multiple agents collect information, the algorithm presented here is extended in Section 5. The following discussion is also restricted to the case where the prior distribution $p_{\mathbf{X}_f}(\mathbf{x}_f)$ and posterior distribution $p_{\mathbf{X}_f|Z_k}(\mathbf{x}_f|Z_k)$ are modeled as Gaussians. Multi-modal distributions are discussed in Chapter 6.

[1]It is worth noting, though outside the scope of this thesis, that a *lazy check* can be used to reduce the computation time spent checking the tree for feasibility [34]. In this framework, the environmental constraints are queried by CL-RRT once for each node when created, and again only whenever the path is selected as the best in the tree to execute.

[2]The CL-RRT algorithm may be made *safe* by requiring that the system only execute paths for which the vehicle can remain in a safe state in the absence of additional nodes[34].

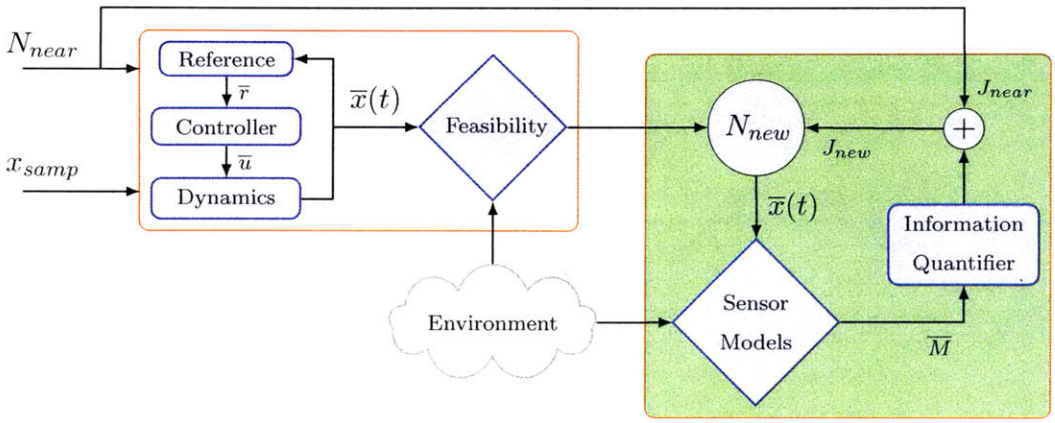


Figure 3-2: Block diagram of IRRT. The red shaded box denotes the typical CL-RRT functionality. The green shaded box denotes the IRRT extension to CL-RRT.

Measurement Pose Sequences

As information collection is predicated on a measurement sequence, quantification of trajectory information content begins by predicting the *measurement pose* sequence along that trajectory. Recall that the closed-loop RRT algorithm generates, for each node N, an anticipated state sequence $\overline{x}(t)$ that is notably accurate with respect to the true state sequence. This accuracy is of benefit not only to the constraint satisfaction of CL-RRT, but also the accuracy with which the measurement pose sequence along a node can be predicted.

Consider a single node N with m measurements, and let the time of the k-th measurement along N be denoted as t_k. Figure 3-3 illustrates the simple case of a single sensor with visibility constraints. Generally, as these measurements may arrive from different sensors, we form the list of all measurement times $\mathbf{t} = \langle t_1, t_2, ..., t_m \rangle$ and the list of all measurement poses $\overline{M} = \langle \mu_1, \mu_2, ..., \mu_m \rangle$ by interleaving the respective lists generated for each sensor.

Therefore, consider a single sensor ζ in the set of sensors $\mathcal{Z}_q$ onboard agent q. If the measurement process is periodic with known frequency f_ζ, then the measurement

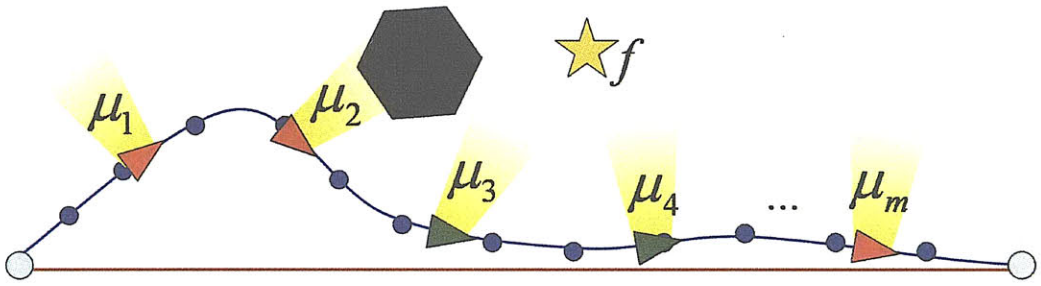


Figure 3-3: Diagram of the path measurement sequence along a single node. The navy blue circles denote elements of the anticipated state sequence. The measurement poses $\overline{\mu}_k$ are interpolations of the state sequence, where the color of the triangle symbolizing the vehicle orientation is green if the target f is visible by the sensor at timestep k, or red if the target is outside of or obstructed in the sensor's (yellow) field of view.

interarrival time

$$T_\zeta = \frac{1}{f_\zeta} \tag{3.1}$$

is deterministic. If the measurements arrive in a stochastic fashion, the time interval used in the pose prediction is the expected value of the measurement interarrival time, i.e.,

$$\overline{T}_\zeta = \mathbb{E}\left[t_{k+1} - t_k\right]. \tag{3.2}$$

With the knowledge that the last measurement from ζ on the parent node occurred at time t'^ζ, the list $\mathbf{t}^\zeta = \langle t_1^\zeta, t_2^\zeta, ..., t_m^\zeta \rangle$ can be formed using the relationship

$$t_k^\zeta = t'^\zeta + kT_\zeta, \quad k = 1, ..., m^\zeta \tag{3.3}$$

until t_k^ζ exceeds the time duration $\tau(N|N_{parent})$ of the node, where N_{parent} is the parent of node N.

The list of measurement times for sensor ζ is merged into a sorted map from all sensor times to the corresponding sensor. Let t_k denote the k-th such measurement time, ζ_k the associated sensor, and μ_k the measurement pose, which can be interpo-

lated from $\overline{x}(t)$. Suppose that $t_k \in [t', t'']$, where $t'' - t' = T_{\text{sim}}$. Then the interpolated pose μ_k at the k-th measurement is

$$\mu_k = (1 - \alpha)\overline{\mathbf{x}}_q(t') + \alpha\overline{\mathbf{x}}_q(t''), \tag{3.4}$$

where $\alpha = (t_k - t')/T_{\text{sim}}$, and T_{sim} is the simulation timestep used to generate the discretized representation of $\overline{x}(t)$.

Fisher Information Application

The Fisher Information framework discussed in Section 2.4 is now used to quantify the node's information content based on the anticipated measurement pose sequence $\overline{M} = \langle \overline{\mu}_1, \ldots, \overline{\mu}_m \rangle$.

Information is always defined with respect to a certain estimation process. In general, the goal is to determine the value of an unknown (potentially time-varying) quantity $\mathbf{x}$ via estimation from (noisy) measurements $\mathbf{z}_k$ from sensor ζ_k. Consider a *feature* f with an unknown state $\mathbf{x}_f$ that one must attempt to estimate. In this section, we restrict our attention to the discrete-time model

$$\mathbf{x}_{f,k+1} = \Phi_p(t_{k+1}, t_k)\mathbf{x}_{f,k} + \mathbf{w}_k \tag{3.5}$$

$$\mathbf{z}_{f,k} = \mathbf{h}\left(\mu_k, \mathbf{x}_{f,k}\right) + \mathbf{v}_k, \tag{3.6}$$

where $\Phi_p(t_{k+1}, t_k)$ is a (linear) state transition matrix for feature f, $\mathbf{h}$ is a (generally nonlinear) function of the measurement pose μ_k and the true feature state $\mathbf{x}_{f,k}$ at timestep k, and $\mathbf{w}$ and $\mathbf{v}$ are uncorrelated, zero-mean, white Gaussian sequences,

with

$$\mathbb{E}\left[\mathbf{w}_k \mathbf{w}_j^T\right] = \begin{cases} Q_k, & j = k \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

$$\mathbb{E}\left[\mathbf{v}_k \mathbf{v}_j^T\right] = \begin{cases} R_k, & j = k \\ 0, & \text{otherwise} \end{cases} \tag{3.8}$$

$$\mathbb{E}\left[\mathbf{w}_k \mathbf{v}_j^T\right] = 0, \quad \forall k, j. \tag{3.9}$$

The matrices $Q_k \geq 0$ and $R_k > 0$ are referred to as the process and sensor noise covariance matrices, respectively.

Systems with discrete-time models of the form

$$\mathbf{x}_{f,k+1} = \Phi_f(t_{k+1}, t_k)\mathbf{x}_{f,k} + \mathbf{w}_k \tag{3.10}$$

$$\mathbf{z}_{f,k} = H_k \mathbf{x}_{f,k} + \mathbf{v}_k, \tag{3.11}$$

are called *linear-Gaussian* systems because both the transition and measurement models are linear, and the additive white sequence is Gaussian. For such systems, recall from Section 2.4 that Fisher Information Matrix (FIM) recursion can be written as

$$J_{k+1} = (Q_k + \Phi_{k+1|k} J_k^{-1} \Phi_{k+1|k}^T)^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1}. \tag{3.12}$$

The linearity assumption on the observation system can be relaxed by utilizing the linearized FIM as an approximation of the CRLB inverse. Consider again systems with discrete measurements $\mathbf{z}$ that are nonlinear in both the target state $\mathbf{x}_f$ and measurement pose μ, and are thus of the form

$$\mathbf{z_k} = \mathbf{h}(\mu_k, \mathbf{x}_f) + \mathbf{v_k}, \tag{3.13}$$

where $\mathbf{v}$ is a vector of zero-mean, white Gaussian sequences. Henceforth, we will approximate the FIM by defining H to be the Jacobian of the nonlinear measurement

function, i.e.,

$$H_k(\overline{\mu}_k, \hat{\mathbf{x}}) \triangleq \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\bigg|_{\mu_k=\overline{\mu}_k,\ \ \mathbf{x}_f=\hat{\mathbf{x}}_f(t)}. \tag{3.14}$$

Note that the assumption of Gaussian noise is retained, a limitation that is relaxed in Chapter 6.

In the case of time-varying features, the recursion in (3.12) can be used to incrementally quantify the nodes of a tree structure containing trajectories. However, information quantification for multiple agents or sensors in the presence of dynamic targets remains cumbersome. The following further assumes stationary targets ($Q_k = 0, \Phi_k = 0, \forall k$); dynamic targets are deferred for future work (c.f. Section 7.2). While this is not an inherent restriction on the solution algorithm presented in this work, it is an assumption which nevertheless simplifies the form of the tree-based information quantification, which now utilizes the recursion

$$J_{k+1} = J_k + H_{k+1}^T R_{k+1}^{-1} H_{k+1}. \tag{3.15}$$

Equation (3.15) suggests that information metrics for a particular path in the tree can utilize the additivity of FIMs for measurements of that path's constituent nodes. Therefore, one need only specify how an individual node contains an *increment* of Fisher Information; operations involving tree expansion (Section 3.4.2) and path selection (Section 3.4.3) can act recursively on the tree. Throughout, it is understood that the node N_j whose information is quantified is the child of a previously quantified node N_i.

The assumption of Gaussianity on the measurement noise is a requirement of a broader assumption that the posterior distribution can be well approximated by a single Gaussian. Another such requirement is that the prior distribution is Gaussian, by which the root node N_{root} of the tree can be initialized by using the actual

information matrix, i.e.,

$$J_{root}(\hat{\mathbf{x}}_f) = P_f(t)^{-1}, \tag{3.16}$$

where

$$P_f(t) = \mathbb{E}\left[(\mathbf{x}_f - \hat{\mathbf{x}}_f(t))(\mathbf{x}_f - \hat{\mathbf{x}}_f(t))^T\right] \tag{3.17}$$

is the error covariance matrix for target f at that instant t. For each target f, the FIM $J_j(\hat{\mathbf{x}}_f)$ of a child N_j is formed by a recursive update from its parent N_i

$$J_j(\hat{\mathbf{x}}_f) = J_i(\hat{\mathbf{x}}_f) + \sum_{k=1}^{m} \nu(\overline{\mu}_k, \hat{\mathbf{x}}_f, \hat{\mathcal{E}}) H_k^T(\overline{\mu}_k, \hat{\mathbf{x}}_f) R_k^{-1} H_k(\overline{\mu}_k, \hat{\mathbf{x}}_f), \qquad \forall f \in \mathcal{F}, \tag{3.18}$$

where m is the number of measurements along the path segment, $\hat{\mathcal{E}}$ is the environment representation, and ν is a binary-valued function capturing the success/occlusion of a measurement. In this way, the tree FIMs are populated and can be recomputed, e.g., after target location estimates have been updated.

In the presented approach, the cost associated with information for target f at node N_i is specified as the A-optimality condition on the FIM,

$$\mathcal{I}_i(\hat{\mathbf{x}}_f) = \texttt{trace}(J_i^{-1}(\hat{\mathbf{x}}_f)), \tag{3.19}$$

which has been shown to be better suited than other FIM optimality conditions for the 3D target tracking case [51]. In the multi-sensor or multi-target case, convex combinations of the FIM A-optimality costs

$$\mathcal{I}_i = \sum_{f \in \mathcal{F}} w_f \mathcal{I}_i(\hat{\mathbf{x}}_f), \qquad \sum_{f \in \mathcal{F}} w_f = 1 \tag{3.20}$$

with relative weights w_f can be used to bias information collection, e.g., towards mission-critical targets. Summation of the A-optimality costs is consistent with the nature of the multi-objective problem. Moreover, it should be noted that simply summing the FIMs (and not the associated A-optimality costs) over all targets at a given

Algorithm 3 IRRT, TREE EXPANSION

1: Take a sample x_{samp} from the environment
2: Identify the nearest node N_{near} using mixture of EXPLORATION, OPTIMIZATION, and INFORMATION heuristics
3: $\overline{x}(t+k) \leftarrow$ final state of N_{near}
4: **while** $\overline{x}(t+k) \in \mathcal{X}_{free}$ **and** $\overline{x}(t+k)$ has not reached x_{samp} **do**
5: Use reference law to generate $\overline{r}(t+k)$
6: Use control law to generate $\overline{u}(t+k)$
7: Use prediction model to simulate $\overline{x}(t+k+1)$
8: $k \leftarrow k+1$
9: **end while**
10: **for** each feasible node N generated **do**
11: Update cost estimates for N
12: Compute simulated measurement poses $\overline{M}$
13: Compute FIM using (3.15)
14: Add N to $\mathcal{T}$
15: **end for**

measurement pose is imprudent; for example, two targets with singular FIMs could in their sum form a nonsingular FIM, thereby masking the momentary unobservability of each target's estimation process.

The ability to simulate expected measurement poses is used in two ways to extend the CL-RRT algorithm for information gathering. First, these expected measurements are used to bias tree growth toward regions of high information gain. Second, the vehicle selects paths from the tree that minimize a cost function which explicitly considers information, in addition to path cost and remaining cost-to-go. Both extensions are discussed below in detail.

3.4.2 Tree Growth

This section considers the tree expansion algorithm for IRRT, an extension of CL-RRT tree expansion (Algorithm 1) which incorporates the predicted collection and utilization of information while growing the tree. The IRRT tree expansion algorithm is presented in Algorithm 3, with the modifications (in red) discussed below.

The nearest node selection scheme alternates between a collection of heuristics to identify the node(s) nearest to a sample [15]. In the CL-RRT algorithm (Algorithm 1, line 2), one of two heuristics are probabilistically selected, depending on whether

Algorithm 4 IRRT, Execution Loop

1: $t \leftarrow 0$
2: Initialize tree $\mathcal{T}$ with node at $x(0)$
3: **while** $x(t) \neq \mathbf{x}_{goal}$ **do**
4: Update the current state $x(t)$ and target estimates $\hat{x}_f \ \forall f$
5: Propagate the state $x(t)$ by $\Delta t \rightarrow \overline{x}(t + \Delta t)$
6: **while** time remaining for this timestep **do**
7: IRRT, TREE GROWTH
8: **end while**
9: Update FIMs throughout $\mathcal{T}$ using (3.15)
10: Use information-based cost metric to identify best feasible path, $\mathcal{P}_* \leftarrow \{N_{root}, \ldots, N_{target}\}$
11: Apply best feasible path, if one exists
12: $t \leftarrow t + \Delta t$
13: **end while**

or not a feasible path to the goal has been found. The EXPLORATION heuristic

$$i^* = \underset{i}{\text{argmin}} \ \hat{\tau}(x_{\text{samp}}|N_i) \tag{3.21}$$

uses a simple time metric $\hat{\tau}(x_{\text{samp}}|N_i)$ (e.g., Dubins distance divided by average speed) from the candidate node N_i to the sample x_{samp}, biasing the tree growth toward unexplored regions of the environment. The OPTIMIZATION heuristic

$$i^* = \underset{i}{\text{argmin}} \ \hat{\tau}(x_{\text{samp}}|N_i) + \alpha_\tau \tau(N_i|N_{root}), \quad \alpha_\tau \in [0, 1) \tag{3.22}$$

uses both the simple time metric $\hat{\tau}$ and accumulated path duration $\tau(N_i|N_{root})$ from the root N_{root} to N_i, weighted by α_τ, to bias tree growth towards lower-cost paths[15]. The relative weight $\alpha_\tau \in (0, 1)$ "rewards progress" toward the goal, with higher values of α_τ corresponding to less tolerance of suboptimal paths. The likelihood of each heuristic being used depends on the current tree, favoring the OPTIMIZATION metric when at least one path to the goal has been found and the EXPLORATION metric otherwise [33].

To facilitate the addition of information-maximizing paths to the tree, the IRRT algorithm devotes a significant fixed percentage of its nearest node operations (Algorithm 3, line 2) to an additional, information-based heuristic. This INFORMATION heuristic selects the node that will yield the greatest reduction in the combined in-

formation/distance cost when connected to the sample, specifically by approximating the information gain along a simplified reference path connecting the sample x_{samp} to a nearest-node candidate N_i. The heuristic first approximates the expected number of measurements $\hat{m}$ along the path by multiplying path duration by sensor sampling frequency, then discritizes the path into m' segments, each with weight λ_k and measurement pose μ_k, $k \in \{1, \ldots, m'\}$. The approximate trajectory FIM $\tilde{J}_{s|i}(\hat{\mathbf{x}}_f)$ at the sample x_{samp} for each target f as a result of taking the connecting tree path is given by

$$\tilde{J}_{s|i}(\hat{\mathbf{x}}_f) = J_i(\hat{\mathbf{x}}_f) + \frac{\hat{m}}{m'} \sum_{k=1}^{m'} \lambda_k \nu(\overline{\mu}, \hat{\mathbf{x}}_f, \mathcal{E}) H_k^T(\overline{\mu}_k, \hat{\mathbf{x}}_f) R_k^{-1} H_k(\overline{\mu}_k, \hat{\mathbf{x}}_f), \qquad \sum_{k=1}^{m'} \lambda_k = 1, \tag{3.23}$$

where $J_i(\hat{\mathbf{x}}_f)$ is the FIM for target f at the nearest-node candidate N_i. For each such $\tilde{J}_{s|i}(\hat{\mathbf{x}}_f)$,

$$\tilde{\mathcal{I}}_{s|i} = \sum_{f \in \mathcal{F}} w_f \, \texttt{trace}(\tilde{J}_{s|i}^{-1}(\hat{\mathbf{x}}_f)), \qquad \sum_{f \in \mathcal{F}} w_f = 1. \tag{3.24}$$

Thus, the index i^* of the candidate node yielding the greatest reduction in the combined distance/A-optimality cost is

$$i^* = \underset{i}{\text{argmin}} \; \hat{\tau}(x_{\text{samp}} | N_i) + \alpha_\tau \tau(N_i | N_{root}) + \alpha_{\mathcal{I}} \tilde{\mathcal{I}}_{s|i}, \quad \alpha_\tau \in [0, 1), \tag{3.25}$$

where $\alpha_{\mathcal{I}} \in \mathbb{R}^+$ is a user-specified information weighting parameter to be described in the next section.

Whenever new feasible nodes N_{new} are generated for the tree, the predicted measurement poses $\overline{M}$ are stored within the node (line 12). These measurement poses are used to compute the FIM based on the current target estimates $\hat{x}_f \; \forall f$, both when the node is created (line 13) and whenever the best path is selected, as discussed next.

3.4.3 Path Selection

This section considers the path selection algorithm for IRRT, an extension of the CL-RRT execution loop (Algorithm 2) which incorporates information-gathering into the selection of paths to execute. The provided formulation allows the vehicle to achieve the dual objectives of gathering a desired amount of information about some target(s) and arriving at a goal state x_{goal}, giving the operator the freedom to specify the relative importance of these tasks. The IRRT execution loop is presented in Algorithm 4, with the modifications (in red) discussed below.

Given a tree of feasible trajectories, the algorithm must periodically identify the "best" path for the vehicle to execute, in terms of some cost metric [15], from the root x_{root}. Since every node in the tree is connected to the root via a single path, it is sufficient to iterate over the individual nodes to identify a cost-minimizing "target node" N_{target}, implicitly defining the path $\{N_{root}, \ldots, N_{target}\}$. In the CL-RRT algorithm, the cost metric used (Algorithm 2, line 9) typically depends on whether or not a feasible path to the goal has been found. If at least one node has feasibly reached the goal, the node N_i among that set which minimizes the total path duration $\tau(N_i|N_{root})$ is selected. If no feasible path to the goal has been found, any number of cost metrics might be appropriate, such as minimizing the remaining distance to the goal [15].

In the IRRT algorithm, a single, multi-objective cost metric is used (Algorithm 4, line 10), which considers both progress toward the goal and the value of information collection. This cost function here takes the form

$$C(N_i) = \alpha_\tau \tau\left(N_i|N_{root}\right) + \tau^*\left(N_i\right) + \alpha_\mathcal{I}\mathcal{I}_i, \tag{3.26}$$

where $\tau(N_i|N_{root})$ is the simulated time to travel from the root node N_{root} to node N_i, $\tau^*(N_i)$ is the lower-bound cost-to-go (e.g., Euclidean or Dubins length divided by average speed) from N_i to the goal, and $\mathcal{I}_i$ is the information-related cost component. The weights α_τ and $\alpha_\mathcal{I}$ can be adjusted to reflect the relative importance of information gathering and of following minimal-time paths to the goal. To ensure all

recent measurements are taken into account, the latest target estimates are measured at the beginning of each execution loop (line 4), which are then used to update the FIM of each node in the tree (line 9). Though this FIM update is performed on the entire tree on each pass, this is a computationally efficient operation compared to other aspects of the algorithm, such as constraint evaluation.

Of particular note with this cost function is that it can be shown to result in "smooth" mission-level behaviors, in the sense that negligible churning between information collection and goal directedness exists. Rather, the planner is always conscious of the inherent tradeoff and will generate behaviors that, for example, conclude missions by manuevering to collect information while remaining relatively close to the goal. It should also be noted as a limitation of IRRT, and RRTs in general, that mission-critical requirements like maximum allowable duration and/or minimum required information collection are not well handled; it is difficult enough to find, let alone guarantee that one could find, a feasible solution to such requirements in finite time. Despite this, IRRT can be shown through simulations in Chapter 4 to perform well empirically under a number of previously prohibitive general constraints.

Chapter 4

IRRT Scenarios

This chapter presents simulation results demonstrating the effectiveness of the IRRT algorithm in managing the competing objectives of information-gathering and prompt goal arrival in real-time, while satisfying a complex constraint set. Results are ordered so as to present a progression in capability. The initial scenario is used as a demonstrative example of how the algorithm exhibits simple information-gathering behaviors, consisting of a single holonomic vehicle estimating a single target without sensing constraints in an uncluttered environment. In the subsequent scenarios, more complex extensions to the problem (in line with Section 1.2) are considered, including non-holonomic vehicles, limited-field-of-view sensing, cluttered obstacle environments, multiple targets, and finally three dimensional flights. Even subject to these constraints, which render many existing approaches in the literature intractable, the IRRT algorithm generates paths with emergent information-gathering characteristics.

Some basic analysis is provided to illuminate the trade-off between information-gathering and goal arrival that is taking place. Towards describing the statistical performance of IRRT, an extensive comparison of CL-RRT and IRRT, each with various heuristics, is also provided.

4.1 Simulation Overview

Before proceeding to the simulation results, a brief overview of the simulation environment and software is provided.

The IRRT algorithm as presented in Section 3.4 has been implemented in real-time Java with the modular RRT-Sim software package developed by the author and Brandon Luders at the MIT Aerospace Controls Laboratory. RRT-Sim retains the object-oriented paradigm of its development language in allowing one to plan multiple open-loop, closed-loop, or information-rich RRTs for vehicles with modular dynamic, reference, and sensor models. The environment model is similarly modular, and the obstacle space can be constructed from arbitrary polyhedrons and spheres. Vehicles may be simulated in software or controlled over the network in the RAVEN testbed [26]; so far, RRT-Sim has been used in physical demonstrations for a diverse array of wheeled and aerial robots.

While the IRRT algorithm as presented admits a comparable level of abstractness, the scenarios of this chapter will, for demonstrative purposes, focus on bearings-only sensing of fixed targets whose unknown three-dimensional position is estimated using an extended Kalman filter (EKF). Some remarks are in order. The use of bearings-only sensing is motivated only by the intention of providing intuitive, geometric insights into the performance of IRRT. Furthermore, as IRRT is concerned with minimizing the Cramér-Rao Lower Bound on the estimation error covariance, and not explicitly the perfomance of particular estimators, we eschew discussion of the implementation issues inherent in more recent filtering solutions (e.g., of particle filters [55]). Comparison of EKF and particle filter implementations and their impact on the selection of information-rich paths is given, e.g., in [51].

The EKF used is now briefly described. Recall from Section 3.4.1 that the target dynamics are assumed to be linear, while the measurement process remains nonlinear. It is further assumed that the process and measurement noises are additive. Therefore,

the system dynamics for each target f are

$$\mathbf{x}_k^{\ominus} = \Phi_{k|k-1}\mathbf{x}_{k-1}^{\oplus} + \mathbf{w}_{k-1} \tag{4.1}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{4.2}$$

where $\Phi_{k|k-1}$ is the (linear) state transition matrix between timesteps $k-1$ and k, $\mathbf{h}$ is the nonlinear measurement function, and $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ are uncorrelated, zero-mean, white Gaussian sequences with respective covariance matrices Q_{k-1} and R_k. The filter equations for the EKF are divided into a prediction phase

$$\hat{\mathbf{x}}_k^{\ominus} = \Phi_{k|k-1}\hat{\mathbf{x}}_{k-1}^{\oplus} \tag{4.3}$$

$$\hat{\mathbf{z}}_k^{\ominus} = \mathbf{h}\left(\hat{\mathbf{x}}_k^{\ominus}\right) \tag{4.4}$$

$$P_k^{\ominus} = \Phi_{k|k-1}P_{k-1}^{\oplus}\Phi_{k|k-1}^T + Q_{k-1}, \tag{4.5}$$

a linearization phase

$$H_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^{\ominus}}, \tag{4.6}$$

and an update phase

$$K_k = P_k^{\ominus}H_k^T\left[H_kP_k^{\ominus}H_k^T + R_k\right]^{-1} \tag{4.7}$$

$$\hat{\mathbf{x}}_k^{\oplus} = \hat{\mathbf{x}}_k^{\ominus} + K_k\left(\mathbf{z}_k - \hat{\mathbf{z}}_k^{\ominus}\right) \tag{4.8}$$

$$P_k^{\oplus} = P_k^{\ominus} - P_k^{\ominus}H_k^T\left[H_kP_k^{\ominus}H_k^T + R_k\right]^{-1}H_kP_k^{\ominus}. \tag{4.9}$$

For any kth measurement, the relative vector between target f and agent q is denoted by

$$\mathbf{r}_k \triangleq \begin{bmatrix} r_x & r_y & r_z \end{bmatrix}_k^T = \mathbf{x}_q - \mathbf{x}_f. \tag{4.10}$$

In a target-centric frame translated (but not rotated) from a global coordinate frame common to all agents, the azimuth angle β and elevation ϕ, as shown in Figure 4-1,

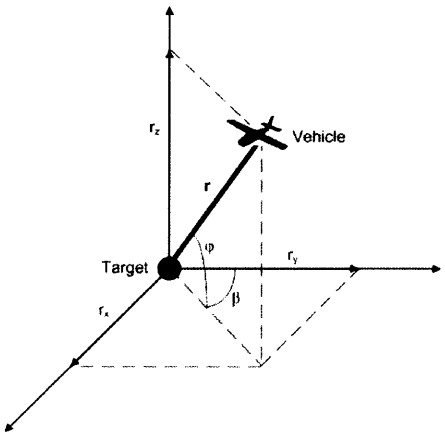


Figure 4-1: Diagram of the the azimuth β and elevation ϕ between the vehicle and one target [51]. The axes are assumed to be parallel to those of a global coordinate frame common to all agents in the scenario.

are exactly

$$\beta = \tan^{-1}\left(\frac{r_x}{r_y}\right) \tag{4.11}$$

$$\phi = \tan^{-1}\left(\frac{r_z}{\sqrt{r_x^2 + r_y^2}}\right). \tag{4.12}$$

Therefore, the (nonlinear) measurement model is

$$\mathbf{h}(\mathbf{x}_{f,k}) = \begin{bmatrix} \tan^{-1}\left(\frac{r_x}{r_y}\right) \\ \tan^{-1}\left(\frac{r_z}{\sqrt{r_x^2+r_y^2}}\right) \end{bmatrix}, \tag{4.13}$$

for which the Jacobian is

$$H_k = \begin{bmatrix} -\frac{r_y}{r_x^2+r_y^2} & \frac{r_x}{r_x^2+r_y^2} & 0 \\ \frac{r_x r_z}{(r_x^2+r_y^2+r_z^2)\sqrt{r_x^2+r_y^2}} & \frac{r_y r_z}{(r_x^2+r_y^2+r_z^2)\sqrt{r_x^2+r_y^2}} & -\frac{\sqrt{r_x^2+r_y^2}}{(r_x^2+r_y^2+r_z^2)} \end{bmatrix}. \tag{4.14}$$

In practice, since the true position $\mathbf{x}_f$ of target f is unknown, the linearized measurement matrix H, and by extension the Fisher information matrices, are computed

using the estimate $\hat{\mathbf{x}}_f$. That is, the above linearization is evaluated for

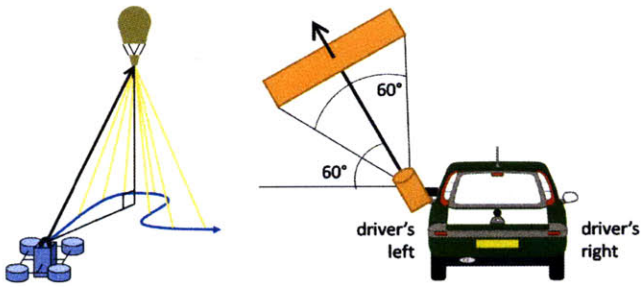
$$\hat{\mathbf{r}}_k = \mathbf{x}_q - \hat{\mathbf{x}}_f. \tag{4.15}$$

All simulations were performed on an Intel 2.53 GHz quad-core laptop with 3.48GB of RAM. The vehicle's current path is selected from the tree at a rate of 4Hz; the tree capacity is specified to be 2000 nodes. A 10cm buffer is also placed around the vehicle for safety reasons. All (bearings-only) sensors are assumed to have a measurement rate of 15 Hz.

4.2 Quadrotor Scenario

Consider a quadrotor unmanned aerial vehicle (UAV) navigating through an obstacle environment at a fixed altitude while tracking a stationary aerial target at significantly higher altitude. We assume for now that the onboard sensor has an unobstructed view of the target at all times (Figure 4-2(a)); any obstacles in the environment still obstruct motion but do not provide visual occlusion. An onboard sensor makes periodic bearing (heading and inclination) observations of the aerial target, which are passed through the EKF to reduce uncertainty in the target's location. The UAV's objective is to gather information about the target through measurements while efficiently traveling to some goal location; the relative importance of these tasks is governed through the weights in the cost function (3.26).

In this scenario, the 40-cm UAV begins at $\mathbf{x}_q(t_0) = (1.5, 1.0, 1.0)^T$ m with an unobstructed path to the goal at $\mathbf{x}_g = (0.0, 1.0, 1.0)^T$ m, but also uncertain knowledge of a target located at $\mathbf{x}_f = (0.0, -1.75, 4.0)^T$ m beyond a single obstacle (Figure 4-3). For this scenario, we have selected $\alpha_\tau = 0.5$ and $\alpha_\mathcal{I} = 6000$ sm^{-2}. A "carrot" reference law moves the reference toward the next waypoint at a fixed rate of 0.3 m/s; an LQR control law is then used by the quadrotor to track this reference. The agent begins with equal uncertainty of the target in all directions ($P(t_0) = 8.0I_3$ m^2), but as bearing measurements are taken, the longest axis of the uncertainty ellipsoid

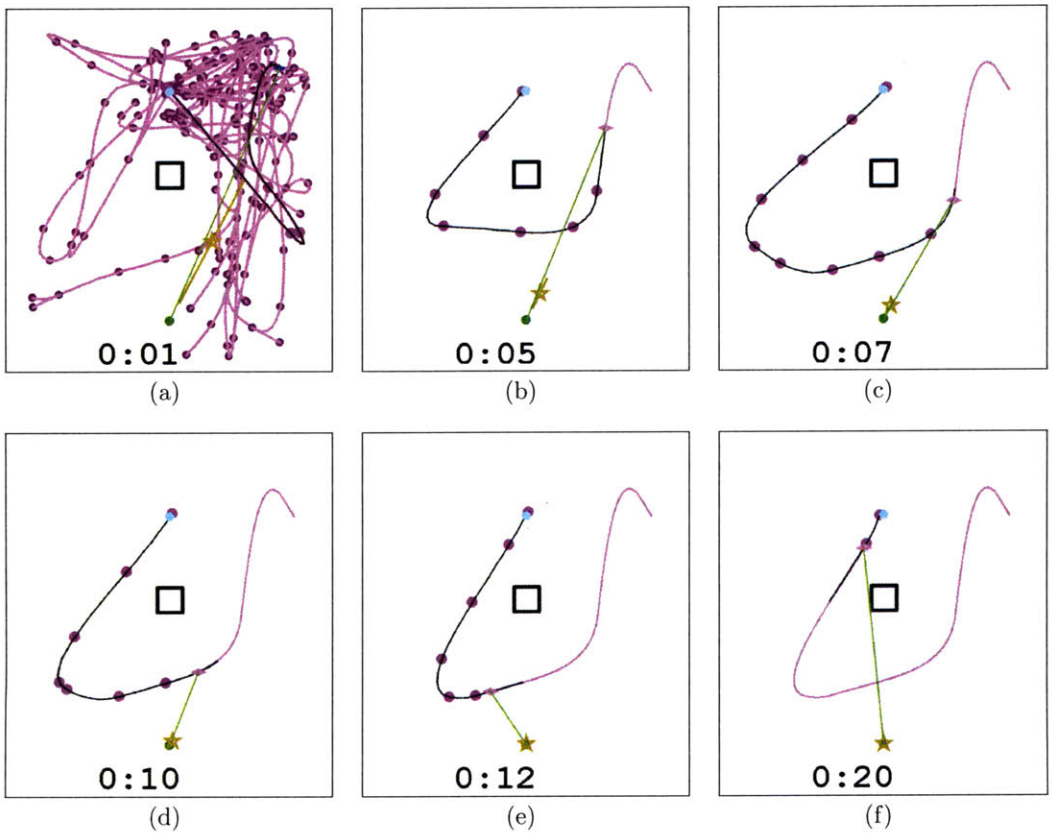


(a) (b)

Figure 4-2: (a) Diagram of the quadrotor UAV scenario. As the UAV navigates at a fixed altitude below that of the aerial target, the onboard sensor has an unobstructed view of the aerial target. (b) Dubins car diagram with a pointed monocular vision sensor.

aligns with the camera axis. This agrees with the intuition that with a bearings-only sensor, the greatest degree of uncertainty is in the range estimate to the target.

An example trial of the scenario is depicted in Figure 4-3. Whereas the basic RRT algorithm would direct the UAV straight to the goal, the paths chosen here represent the value of deviating from a minimal-time path to gather target information. The initially selected plan (Figure 4-3(a)) specifies a path that moves the sensor in a direction orthogonal to the current line-of-sight and, by virtue of using bearings-only sensing, the largest axis of the uncertainty ellipsoid. As the target estimate evolves and converges toward the true target location (Figures 4-3(b-c)), the planner identifies paths that lower the total cost by both decreasing range to the estimated target location and increasing the number of measurements (by successively lengthening the path). As information is gathered, the information cost component of (3.26) becomes sufficiently small that the total cost is optimized by returning to the goal; at this point, the planner begins selecting successively shorter paths to the goal (Figures 4-3(d-e)). After 20 seconds have elapsed, the vehicle has collected sufficient information on the target and arrives at the goal (Figure 4-3(f)). From a simple cost tradeoff in (3.26) between path length and uncertainty, complex target tracking behavior for the autonomous UAV has emerged naturally.



(a) (b) (c) (d) (e) (f)

Figure 4-3: Snapshots of a typical trajectory for a simulated quadrotor navigating toward a goal while tracking an aerial target. The vehicle (magenta diamond, highlighted in blue in (a)) starts at top-right, and attempts to gather information about the target (green, bottom-center) and reach the goal waypoint (cyan, top-center) while avoiding obstacles (black). Relative uncertainty in the target location is represented with a gold ellipse, with the current estimate denoted by a gold star. The vehicle's line-of-sight to the target is denoted with either a green or red line (not seen in this figure); the former denotes positive visibility, the latter a loss thereof. The current RRT tree is marked in magenta, but is suppressed after (a) in the interest of clarity. The vehicle's currently selected path is emphasized in black, where the magenta dots correspond to nodes. All timestamps are in seconds.

4.3 Dubins Car Scenario

This scenario considers a more complex problem formulation, specifically by considering non-holonomic vehicle dynamics and sensing constraints. Consider a small Dubins car agent traversing an obstacle-free environment while esimating the location of a stationary aerial target. As opposed to the previous example, the agent's monocular sensor is limited to a field of view of 40° in each of the horizontal and vertical axes. The sensor is yawed 90° (out the driver's left side) and pitched up by 60° from the horizontal plane (Figure 4-2(b)); thus the agent must achieve a proper combination of lateral distance and heading to see the target.

In this scenario, the 20-cm car begins at $\mathbf{x}_q(t_0) = (-2.5, -3.5, 1.0)^T$ m with an unobstructed path to the goal at $\mathbf{x}_g = (-2.5, 3.5, 1.0)^T$ m, but also uncertain knowledge of a target located at $\mathbf{x}_f = (0.0, 0.0, 2.0)^T$ m. For this scenario, we have selected $\alpha_\tau = 0.5$ and $\alpha_\mathcal{I} = 8000$ sm^{-2}. The car is assumed to move at a fixed velocity of 0.4 m/s; a variation of the pure pursuit reference law [35] is applied for steering control, assuming forward direction only. Note that this vehicle model could also be used to represent a fixed-wing vehicle operating at a fixed velocity and altitude.

A typical trajectory generated by a trial of this secenario is given in Figure 4-4. The agent quickly plans a winding path that both anticipates measurements about the estimated target position and reaches the goal (Figure 4-4(a)). The uncertainty ellipsoid is markedly elongated in the line-of-sight direction, indicating large uncertainty in depth. As the estimate improves (Figures 4-4(b-d)), the planned path tightens around the estimated target position, in order to take an extended sequence of measurements at close range. Given the relatively high value of $\alpha_\mathcal{I}$, the path ultimately loops itself (Figure 4-4(e)) in order to take additional measurements before finally turning toward the goal (Figure 4-4(f)). Though the vehicle states where measurements can be taken were never explicitly defined, the IRRT algorithm is able to identify these regions and execute a path which spends significant time gathering useful measurements within those regions.

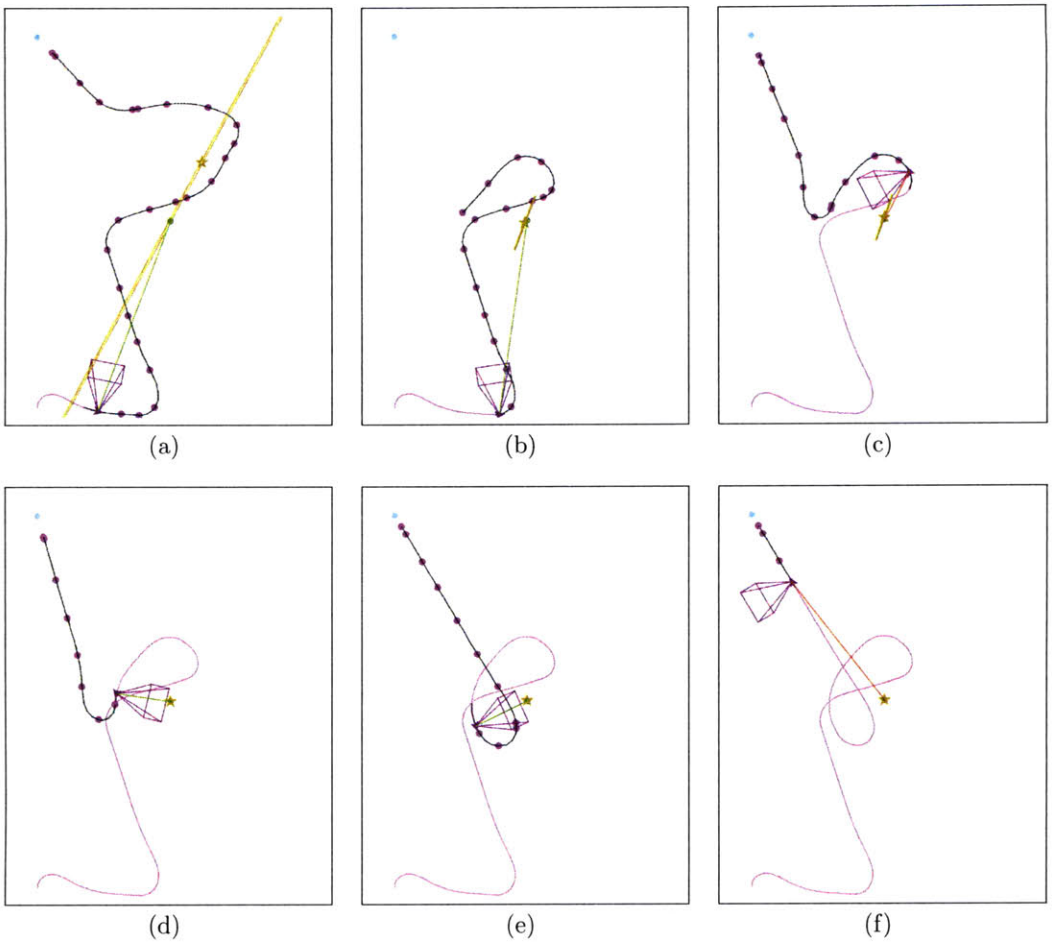


Figure 4-4: Snapshots of a simple Dubins car simulation with side mounted camera navigating toward a goal while tracking one aerial target. See Figure 4-3 for a full legend; the field of view is denoted by the magenta pyramid. The RRT tree has been suppressed for clarity.

4.4 Complex Dubins Scenario

Consider now the full problem statement as outlined in Section 1.2 for the Dubins car, extending the previous example. A Dubins car agent traverses a cluttered environment $\mathcal{E}$ (a bounded, nonconvex subset of $\mathbb{R}^3$) while estimating the location of *multiple* targets, all sufficiently above $\mathcal{E}$. Its monocular sensor is mounted on the driver's left side, pitched up by 60° as before, and has horizontal and vertical fields of view of 60° each. In this scenario, the 20-cm car begins at $\mathbf{x}_q(t_0) = (2.5, -3.5, 1.0)^T$ m with an unobstructed path to the goal at $\mathbf{x}_g = (-2.5, 3.5, 1.0)^T$ m.

The presence of a cluttered obstacle environment presents several challenges over the previous example for the planning algorithm. First, the vehicle must be able to maintain feasibility by avoiding these obstacles; this is itself a challenging problem, since the vehicle moves at a fixed speed and thus cannot safely come to a stop. Second, obstacles in the environment can provide occlusion between the sensor and the targets, greatly complicating the representation of the region of vehicle states where the target(s) are observable. Whereas most heuristic approaches would have to adjust the path in an *ad hoc* manner to achieve feasibility and visibility, these characteristics are embedded naturally in the IRRT algorithm.

An example trial of the scenario is depicted in Figure 4-5; here the RRT trees have been left visible to demonstrate how the set of feasible paths evolves over time. Due to anticipation of occlusion between the sensor and targets, the planner selects paths that result in long periods of visibility. The agent initally plans to move toward the goal and then loiter in its vicinty, occasionally making distant measurements of the targets (Figure 4-5(a)). As the agent approaches the goal, the tree identifies a path which is able to take a better set of measurements while still avoiding obstacles (Figure 4-5(b)). As the target locations are made more precise, subsequent snapshots show the agent carefully moving through the obstacle field, attempting to take closer measurements while ensuring a safe return trajectory to the goal is available (Figures 4-5(c-e)). When the vehicle has gathered enough information with respect to its cost function, it expeditiously plans a path to the goal through a tight corridor (Figure 4-5(f)).

4.5 Analysis

Before proceeding to more complex examples, it is instructive to analyze how effective the IRRT algorithm is in gathering information along its path, and how that capacity is weighed against the objective of reaching the goal. In this section, we revisit the complex Dubins scenario considered in Section 4.4, with particular focus on reduction in target uncertainty over time.

Figure 4-6 plots the value of the information A-optimality cost, (3.19), for the

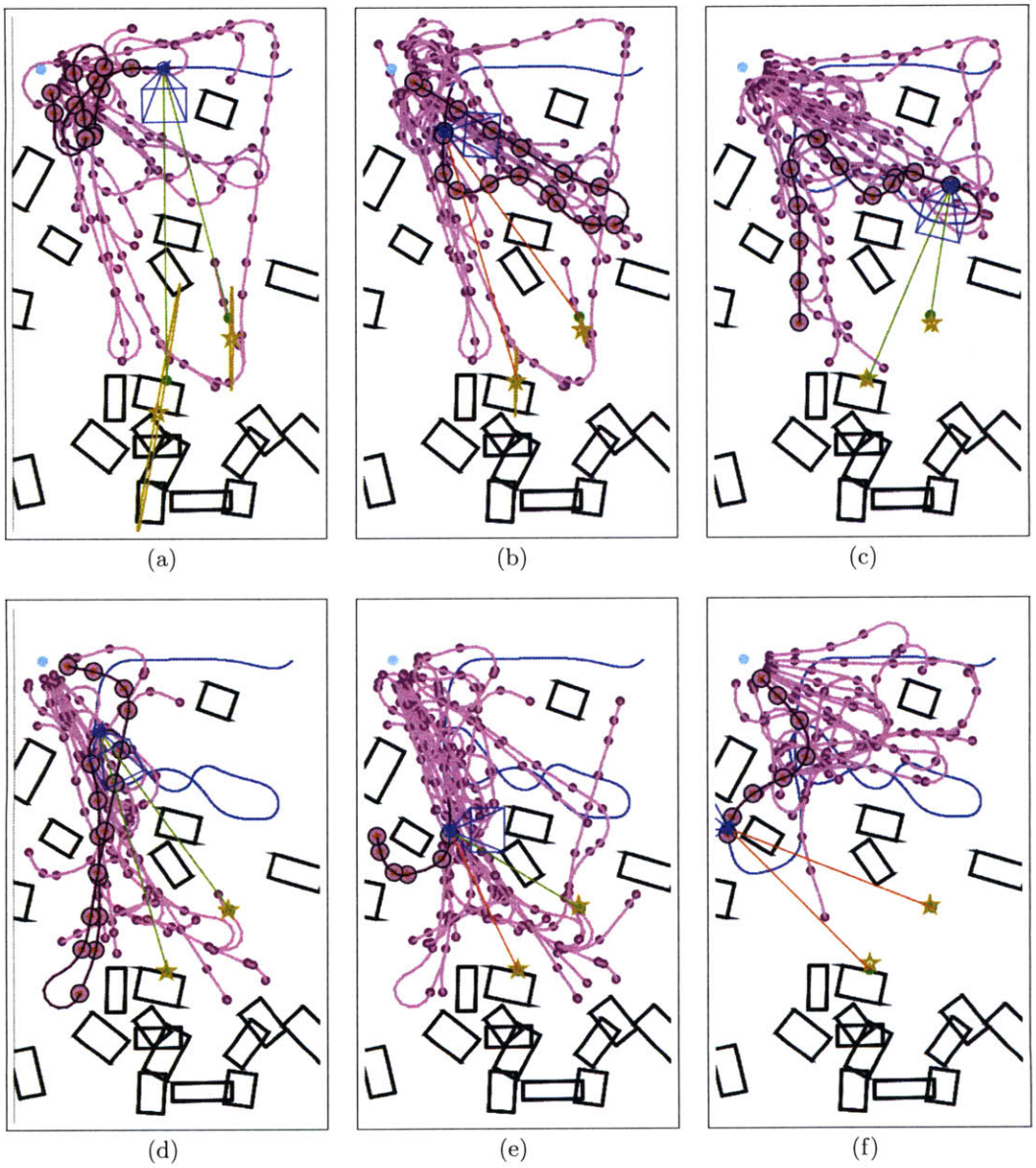


Figure 4-5: Snapshots of a complex simulation of a Dubins car with side mounted camera navigating toward a goal while tracking two aerial targets. See Figure 4-3 for a full legend; the field of view is denoted by a magenta pyramid. The RRT tree is left visible in all figures to demonstrate how the tree evolves over time; for clarity, the vehicle is represented in blue here.

complex Dubins scenario trial shown in Figure 4-5 as a function of time. The colored bars at the bottom of the figure correspond to the time intervals during which each target is visible for measurement by the agent. It is apparent that reduction in the A-optimality occurs when the targets are visible, with the slope of the curve depending on which targets are visible. As Target 2 is more visible in the opening phase of the mission, there is a diminishing return associated with taking data on this target later in the mission, as compared with that of Target 1.

Another important consideration is the effect of varying $\alpha_{\mathcal{I}}$, a user-specified parameter, on the trade-off between uncertainty reduction and final path length. To evaluate its impact, we performed multiple simulations of the complex Dubins scenario for different values of $\alpha_{\mathcal{I}}$, recording the final A-optimality and path duration at the conclusion of each simulation. Seven values of $\alpha_{\mathcal{I}}$ were considered, $\alpha_{\mathcal{I}} = 10^b$, where $b = \{-1, 0, 1, \ldots, 5\}$. Note that as $b \to -\infty$, $\alpha_{\mathcal{I}} \to 0$, approximating the standard, information-naïve RRT algorithm. For each value of b, 25 trials were performed, consisting of 5 trials on the same 5 instances of the complex Dubins scenario, each with a randomized (feasible) obstacle arrangement and initial target estimate.

Figure 4-7 shows the resulting relationship between average mission duration and average terminal A-optimality as a function of $\alpha_{\mathcal{I}}$, which increases from $b = -1$ at bottom-right to $b = 5$ at top-left. As expected, as $\alpha_{\mathcal{I}}$ increases the final A-optimality decreases, at the expense of a longer final path. For the lowest values of $\alpha_{\mathcal{I}}$, the algorithm essentially behaves as standard RRT, ignoring the target in pursuit of the goal. As $\alpha_{\mathcal{I}}$ increases, the A-optimality value becomes relatively more important when selecting paths, and the algorithm will opt to select longer paths which take more measurements of the target.

4.6 Three-Dimensional Scenario

The IRRT formulation can be applied in any number of dimensions; the following scenario demonstrates the capability of IRRT to design information-rich paths for a vehicle operating in a realistic, fully three-dimensional environment. Consider a

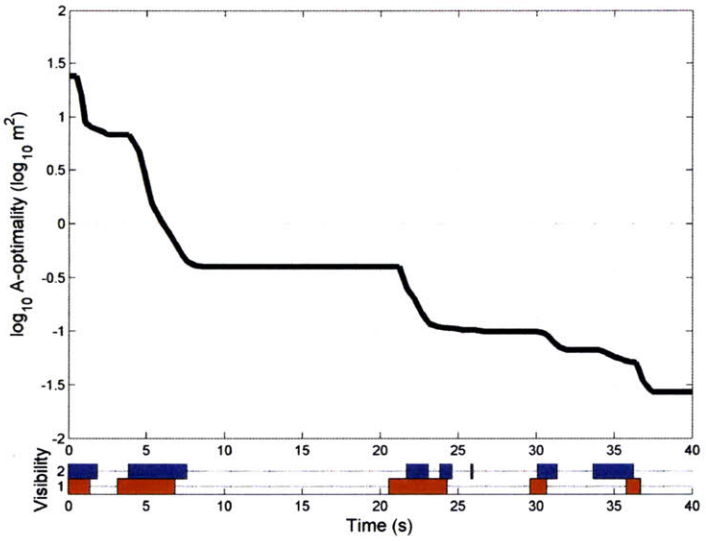


Figure 4-6: Comparison of the information A-optimality cost versus time for the complex Dubins scenario as shown in Figure 4-5. The colored bars at the bottom of the figure correspond to the time intervals during which each target is visible for measurement by the agent.

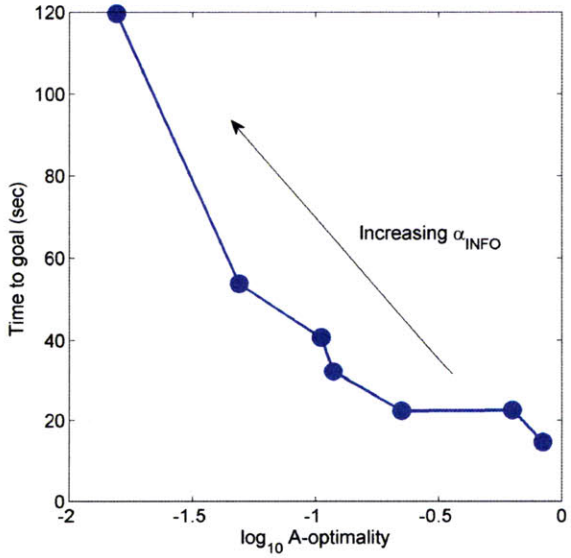


Figure 4-7: Comparison of average mission duration versus average terminal information A-optimality cost. Data points are parameterized by the relative information cost weighting term from (3.26) with values $\alpha_{\mathcal{I}} = 10^b, b = \{-1, 0, \ldots, 5\}$. Each data point corresponds to one value of b, with $b = -1$ at bottom-right and $b = 5$ at top-left.

quadrotor UAV agent navigating through an obstacle environment to track a stationary aerial target. Unlike the first scenario, the agent is free to change both its altitude and heading. In this sense, the RRT is actually sampling in four dimensions, three for position and one for heading. The agent's monocular sensor is mounted on the vehicle's front, parallel to the ground, so it may be advantageous for the agent to change its heading to gain a vantage point for the target.

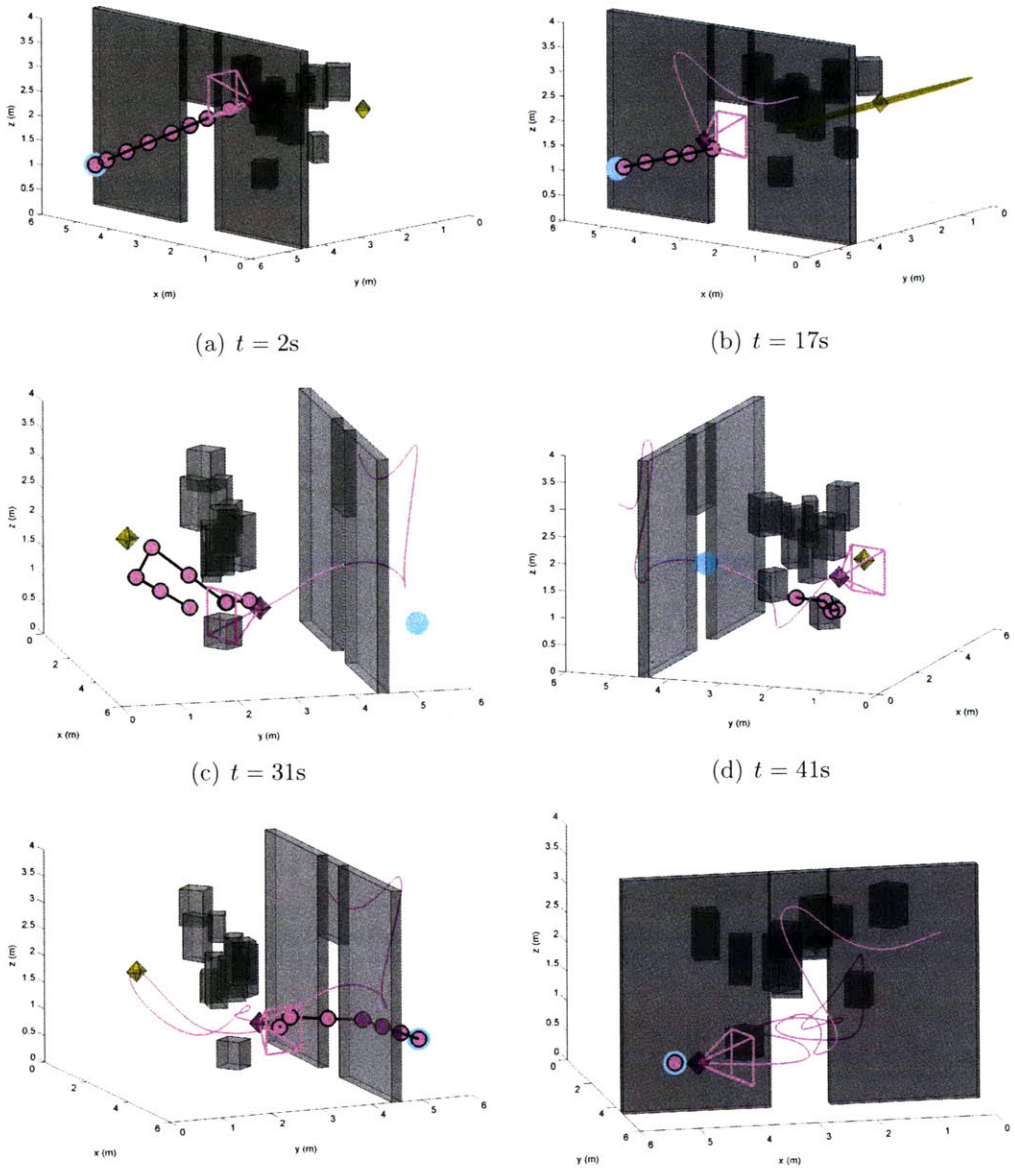
In this scenario, the agent begins on one end of a hallway at $\mathbf{x}_q(t_0) = (0.75, 5.25, 3.0)^T$ m, with an unobstructed path to the goal at $\mathbf{x}_g = (5.25, 5.25, 1.0)^T$ m. However, the agent also seeks to gather information on a target at $\mathbf{x}_f = (2.0, 1.0, 2.0)^T$ m, which is located in a room off the hallway and behind a cluttered region of obstacles.

An example trial of the scenario is depicted in Figure 4-8. The agent begins with a path directly to the goal (Figure 4-8(a)), but the planner then identifies a path which gives the agent sufficient time to rotate and peer into the doorway (Figure 4-8(b)); upon doing so, the agent views the target. Now possessing more accurate knowledge of the target, the planner decides to send the agent into the room and underneath the obstacles (Figure 4-8(c)) to get a much closer view of the target behind them (Figure 4-8(d)). The planner then seeks to return the agent to the goal, and after some wandering succeeds in doing so (Figures 4-8(e-f)).

4.7 INFORMATION Heuristic Comparison

Having characterized the influence of the user-specified parameter $\alpha_{\mathcal{I}}$ on the mission duration and terminal average uncertainty in Section 4.5, it is natural to inquire into the utility of the INFORMATION heuristic. More precisely, it is desired that the performance sensitivity to $p_{\mathcal{I}}$, the probability of selecting this heuristic to use as the nearest node metric, be understood.

Towards this end, consider the following scenario (Figure 4.7). The environment, which is identical for all trials, consists of an axis-aligned box in the first octant of $\mathbb{R}^3$ with dimensions $(10.0, 10.0, 6.0)^T$ m. The cube is populated by 12 randomly generated box obstacles whose placements are uniformly sampled within the environment, and



(a) $t = 2$s

(b) $t = 17$s

(c) $t = 31$s

(d) $t = 41$s

(e) $t = 57$s

(f) $t = 70$s

Figure 4-8: Snapshots of a typical trajectory for a simulated yawed quadrotor navigating toward a goal while tracking an aerial target in three dimensions. The vehicle (magenta diamond) attempts to gather information about the target (estimate and uncertainty in gold) and reach the goal waypoint (cyan) while avoiding obstacles. The agent's field of view is denoted by the magenta pyramid. The vehicle's current reference path is denoted by magenta dots.

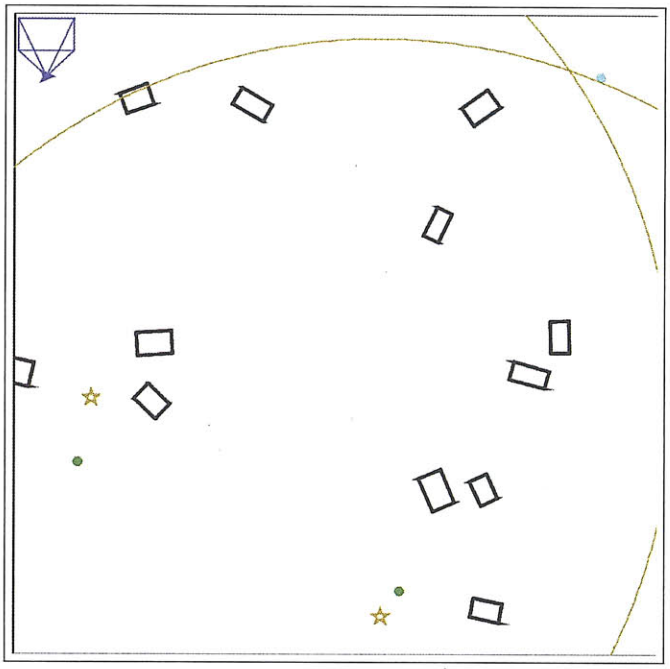


Figure 4-9: Scenario for the INFORMATION heuristic comparison.

whose lengths and widths are uniformly selected from the intervals $[0.4, 0.6]$ m and $[0.25, 0.4]$ m, respectively.

The first target is located at $\mathbf{x}_{f_1} = (1.0, 1.0, 3.0)^T$ m with initial estimate $\hat{\mathbf{x}}_{f_1}(t_0) = (1.2, 4, 3.4)^T$ m. The second target is located at $\mathbf{x}_{f_2} = (6.0, 1.0, 3.0)^T$ m with initial estimate $\hat{\mathbf{x}}_{f_2}(t_0) = (5.7, 0.6, 3.4)^T$ m. Both target covariances are initialized as $P_f(t_0) = \sigma^2 I_3$, $\sigma = 3.0$ m.

The agent is a Dubins car with diameter 0.2 m, initial position $\mathbf{x}_0^{[q]} = (0.5, 9.0, 1.0)^T$ m, initial yaw $\psi_0^{[q]} = 0$ rad, and goal position $\mathbf{x}_g^{[q]} = (9.1, 9.0, 1.0)^T$ m. The onboard monocular (bearings-only) sensor, which operates at 15 Hz and has 60° vertical and horizontal fields of view, is yawed 90° counterclockwise from the front of the car and pitched up 30° from the plane. Simulated bearing measurements are corrupted by a zero-mean, additive white Gaussian noise sequence with a standard deviation of 5°. The parameters of the cost function (3.26) are set to $\alpha = 0.5$ and $\alpha_{\mathcal{I}} = 3000$ sm^{-2} for all agents.

For each value of $p_{\mathcal{I}} \in \{0.0, 0.1, 0.3, 0.5, 0.7\}$, 100 trials were executed. The statistical results are given Figure 4-10 and Table 4.1. Defining the mission-level cost

Table 4.1: Tabulated performance of the INFORMATION heuristic comparison. The total cost is computed as $c = \Delta t + \alpha_{\mathcal{I}} \mathcal{I}_{term}$, where Δt is the mission duration, $\mathcal{I}_{term}$ is the final A-optimality cost, and $\alpha_{\mathcal{I}} = 3000$ sm^{-2}.

		$p_{\mathcal{I}}$				
Metric		0.0	0.1	0.3	0.5	0.7
Mission Duration [s]	median	78.84	79.42	78.50	82.17	84.26
	mean	80.26	80.69	79.44	83.00	87.15
	std	11.90	11.87	11.29	14.31	15.84
Terminal Information Cost [10^{-3} m^2]	median	5.00	3.72	4.01	3.82	3.56
	mean	5.14	3.92	4.10	3.94	3.59
	std	1.52	1.14	1.12	1.20	0.87
Total Mission Cost [s]	median	93.61	91.16	91.67	93.09	95.42
	mean	95.67	92.44	**91.74**	94.83	97.91
	std	11.04	10.98	**10.57**	12.88	14.81

as

$$c = \Delta t + \alpha_{\mathcal{I}} \mathcal{I}_{term}, \tag{4.16}$$

where Δt is the mission duration, $\mathcal{I}_{term}$ is terminal average uncertainty (A-optimality), and $\alpha_{\mathcal{I}}$ is the same information cost weight used in (3.26), the general trends with increasing $p_{\mathcal{I}}$ are increasing Δt and decreasing $\mathcal{I}_{term}$. The data suggests a significant reduction, on the order of 20%, in the mean $\mathcal{I}_{term}$ as a result of increasing $p_{\mathcal{I}}$ from 0 to 0.3. Though this reduction typically comes at the expense of an increased Δt, the difference in mean durations between $p_{\mathcal{I}} = 0$ and $p_{\mathcal{I}} = 0.3$ is slightly negative, though not statistically significant. The mean cost is the most appropriate indicator of which $p_{\mathcal{I}}$ value is best suited for the particular scenario. Within statistical significance, the results presented match the intuition of Section 3.4.2 that the overall performance can be improved by embedding information at both the path selection and tree growth levels. Moreover, the INFORMATION heuristic can only be selected for a fraction of the time, so as to better allow the other heuristics, EXPLORATION and OPTIMIZATION, to impose their respective biases.

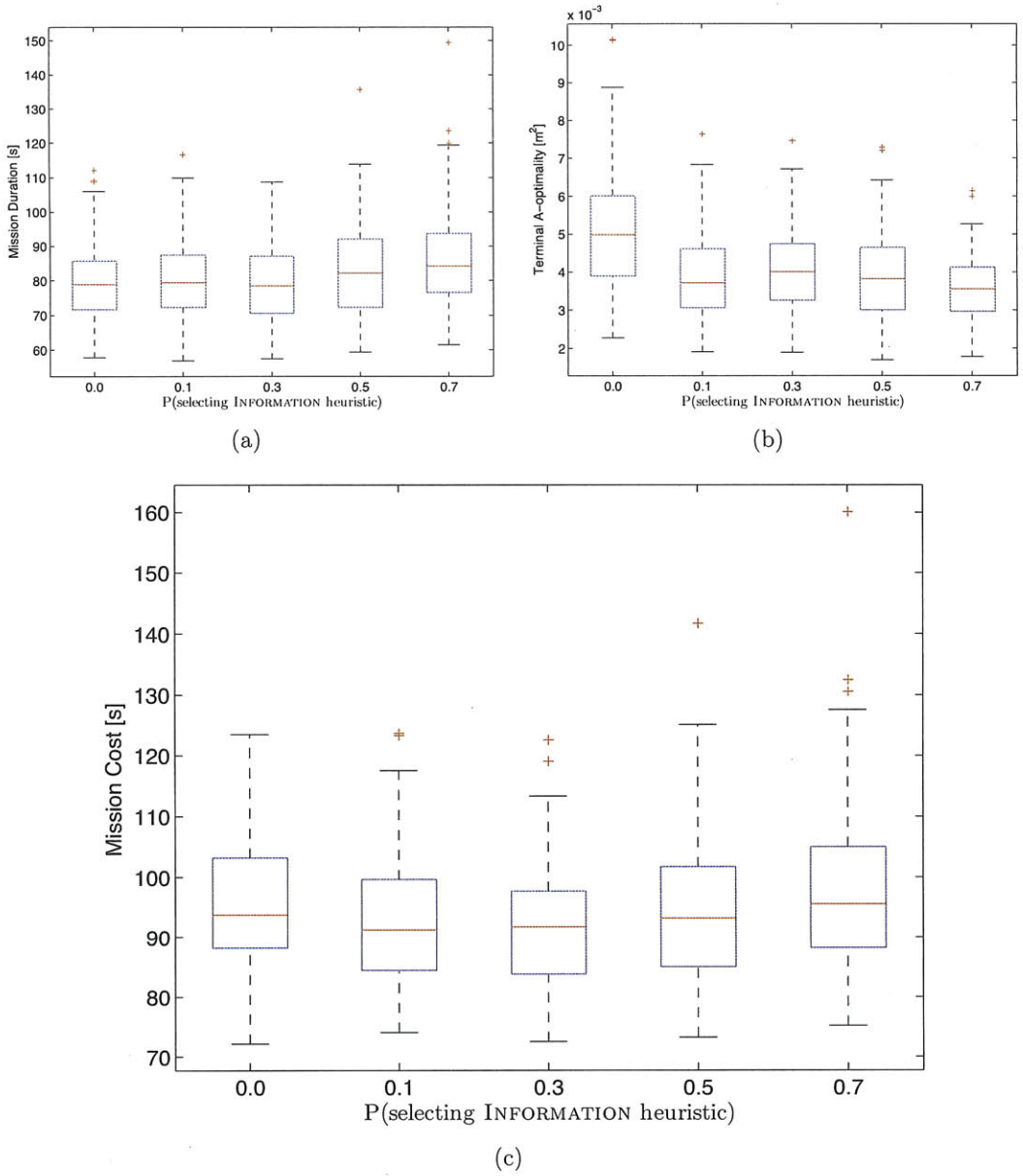


Figure 4-10: Boxplot of the the INFORMATION heuristic comparison. (a) The mission duration exhibits an overall increase as $p_{\mathcal{I}}$ increases. (b) The terminal A-optimality cost exhibits an overall decrease as $p_{\mathcal{I}}$ increases. (c) The overall cost, a linear combination of mission duration and A-optimality cost, can be reduced by raising $p_{\mathcal{I}}$ from 0 to 0.3, after which the overall cost increases.

Chapter 5

Multiagent IRRT

In this chapter, the IRRT algorithm as previously stated in Chapter 3 is extended for multiagent missions with decentralized planning and information collection. The assumptions that enable this extension are briefly reviewed in Section 5.1. The full multiagent IRRT algorithm is then described in Section 5.2. Simulation results provided in Section 5.3 demonstrate the utility of IRRT in constrained, multiagent scenarios.

5.1 Assumptions

The main assumptions of this extension involve the communication capabilities and requirements. Each agent in a network $\mathcal{Q}$ is assumed to be graph-connected to all other agents, though any two agents need not be adjacent or share a particular edge. To limit the necessary communication between agents, each agent is assumed to be executing a local information filter which pre-processes measurements by that agent and sends out only the information contributions over the network. A class of decentralized estimators exists for which the locally produced estimates exactly match those produced by a hypothetical central filter with access to all measurements. An example of such a decentralized estimator can be found in [53]. For the purposes of information gain prediction, it is further assumed that the parameters of each agent, e.g. dynamic and sensor models, are known to all other agents at the beginning of, and are not changed during, the mission.

5.2 Algorithm

The algorithm is described in terms of the relationship between one agent and the rest of the network. Consider that agent $q \in \mathcal{Q}$ grows a tree $\mathcal{T}^{[q]}$ using IRRT (c.f. Section 3.4.2) and periodically selects its most favorable path $\mathcal{P}_*^{[q]}$, defined as the minimal-cost feasible path in $\mathcal{T}^{[q]}$ (c.f. Section 3.4.3). The general path $\mathcal{P}_{N_i}^{[q]}$ is simply the sequence of nodes $(N_{root}, ..., N_i)$ beginning at the root node N_{root} and terminating at node N_i, where the interstitial elements of the sequence are determined by the lineage of N_i. Due to the nature of the cost function (3.26), the cost $C\left(\mathcal{P}_{N_i}^{[q]}\right)$ of a path $\mathcal{P}_{N_i}^{[q]}$ is exactly the cost $C(N_i)$ associated with terminal node N_i. Although the nodes of $\mathcal{P}_*^{[q]}$ encode the entire reference state sequence, simulated state sequence, and measurement sequence, agent q need only transmit the *waypoints*, or terminal reference states, of each node. The other agents $q' \in \mathcal{Q}$ then use the known parameters of q to reconstruct through forward simulation the nodes of $\mathcal{P}_*^{[q]}$. The ability of CL-RRT to reconstruct from sparse waypoint data the complex trajectories generated by individual agents affords a considerable reduction in the necessary communication bandwidth.

The obstacle space $\mathcal{X}_{obs}(t)$ now includes both the static obstacles in the environment and the time-parameterized, dynamic obstacle associated with each agent q along $\mathcal{P}_*^{[q]}$. Though static obstacles are checked for feasibility when growing the tree $\mathcal{T}^{[q']}, \forall q' \in \mathcal{Q}$, the time-parameterized swept space of other agents is only enforced in the so-called "lazy check" step of CL-RRT (c.f. Section 3.3.2). Growing the tree through portions of the swept space of other agents can be justified in light of the tendency for agents to replan during the course of the mission; once a part of the swept space is freed, the planner may quickly select feasible paths that improve the plan cost. Once $\mathcal{P}_*^{[q]}$ has been announced, and until it is overwritten by agent q with a lower cost path, all other agents must select their respective paths to be feasible with respect to $\mathcal{P}_*^{[q]}$. Conversely, at the time of selection, the feasibility of $\mathcal{P}_*^{[q]}$ was enforced with respect to $\mathcal{P}_*^{[q']}, \forall q' \in \mathcal{Q}$.

Because of revisions to the obstacle space $\mathcal{X}_{obs}(t)$ due to replanning, a principled

Algorithm 5 Token Exchange Algorithm for Multiagent CL-RRT, Agent q

Initialize $\mathcal{X}_{obs}(t)$ with static obstacle set
$\mathcal{P}_*^{[q]\ominus} \leftarrow \emptyset$
$\mathcal{P}_*^{[q]\oplus} \leftarrow \emptyset$
for all $q' \in \{\mathcal{Q} \setminus q\}$ **do**
    $\mathcal{P}_*^{[q']} \leftarrow \emptyset$
end for
repeat
    if q is the token holder q^* **then**
        Form replan path $\mathcal{P}_*^{[q]\oplus}$ feasible w.r.t. $\mathcal{X}_{obs}(t)$
        Announce waypoints of $\mathcal{P}_*^{[q]\oplus}$ to the network
        $\mathcal{P}_*^{[q]\ominus} \leftarrow \mathcal{P}_*^{[q]\oplus}$
        $\mathcal{P}_*^{[q]\oplus} \leftarrow \emptyset$
    else
        if token holder q^* has announced a new path **then**
            $\mathcal{X}_{obs}(t) \leftarrow \mathcal{X}_{obs}(t) \setminus \mathcal{P}_*^{[q^*]}$
            $\mathcal{P}_*^{[q^*]} \leftarrow$ reconstructed plan of q^*
            $\mathcal{X}_{obs}(t) \leftarrow \mathcal{X}_{obs}(t) \cup \mathcal{P}_*^{[q^*]}$
        end if
    end if
    Form candidate replan path $\mathcal{P}_*^{[q]\oplus}$ feasible w.r.t. $\mathcal{X}_{obs}(t)$
    Compute cost reduction $\Delta C^{[q]} \leftarrow C\left(\mathcal{P}_*^{[q]\oplus}\right) - C\left(\mathcal{P}_*^{[q]\ominus}\right)$
    Send $\Delta C^{[q]}$ to token holder q^*
    if q is the token holder q^* **then**
        Determine next token holder $q^* \leftarrow \text{argmin}_{q'}\, \Delta C^{[q']}$
        Relinquish token to q^*
    end if
until the mission terminates

method is needed to decide which agent in the network is next allowed to announce its best path. The approach taken here is in line with the token-exchanging algorithm as described in, for example, [69] and reformulated in Algorithm 5. The algorithm can be summarized succinctly as follows. The agent that can best improve its own path cost via replanning receives a token, announces its best path to the network, and relinquishes the token to the agent with the next best cost improvement. Each agent need only transmit its expected cost improvement (a single number) at any time to become a candidate for token holder. It is important to note that Algorithm 5 gives a sketch of the algorithm as seen

Under the assumptions of Section 5.1, agent q will receive the entirety of the information collected from other agents $q' \in \{Q \setminus q\}$. The implication is that at time

t_0, the information concerning target is encapsulated by

$$J_0(\mathbf{x}_f) = P_f^{-1}(t_0) = \mathbb{E}\left[(\mathbf{x}_f - \hat{\mathbf{x}}_f(t_0))(\mathbf{x}_f - \hat{\mathbf{x}}_f(t_0))^T\right]^{-1}, \tag{5.1}$$

and when all agents have completed executing their respective plans, the terminal information is

$$J_{term}(\mathbf{x}_f) = J_0(\mathbf{x}_f) + \sum_{q \in Q} \Delta J\left(\mathbf{x}_f; \mathcal{P}_*^{[q]}\right), \tag{5.2}$$

where $\Delta J(\mathbf{x}_f; \mathcal{P})$ denotes the matrix increment in Fisher Information for target f as a result of the measurement sequence along the path $\mathcal{P}$. Therefore, agent q, when computing the path information in $\mathcal{T}^{[q]}$, initializes the root node with information

$$J_{root}^{[q]}(\hat{\mathbf{x}}_f) = J_0(\hat{\mathbf{x}}_f) + \sum_{q' \in \{Q \setminus q\}} \Delta J\left(\hat{\mathbf{x}}_f; \mathcal{P}_*^{[q']}\right). \tag{5.3}$$

The agent's planner then uses the tree-embedded information metrics as described in previous sections. In this manner, agents will tend to select paths which gather information with respect to other agents' paths, resulting in naturally cooperative behaviors.

A basic assumption made in the decentralized, multiagent extension to IRRT is that a single agent q can anticipate the benefit of the measurement sequences $\overline{M}^{[q']}$ for all $\forall q' \in \mathcal{Q}$. However, following replanning to form $\mathcal{P}_*^{[q]}$, no guarantee exists that the other agents will continue executing the previously announced paths. In fact, it is likely that any agent will replan based on both the evolution of target estimates and the decisions of other agents. The mitigation of *information-loss* due to an individual agent replanning its path is beyond the scope of this work. Such mitigation likely would require time-based information discounting which takes tree diversity into account, and will be considered in future work.

5.3 Simulation Results

5.3.1 Single Target Scenario

Consider a team of two Dubins agents, collectively tasked with taking measurements of an aerial target. Each agent plans its own paths while using IRRT with decentralized (but, recall, identical) estimates. Each agent has a monocular sensor, mounted on the vehicle's front and pitched up 25° from the plane, with 50° horizontal and vertical fields of view; the sensors are assumed to be identical. A target is placed at $\mathbf{x}_f = (0.0, 1.0, 2.0)^T$ m. The mission consists of planning a path for the two agents with starting positions

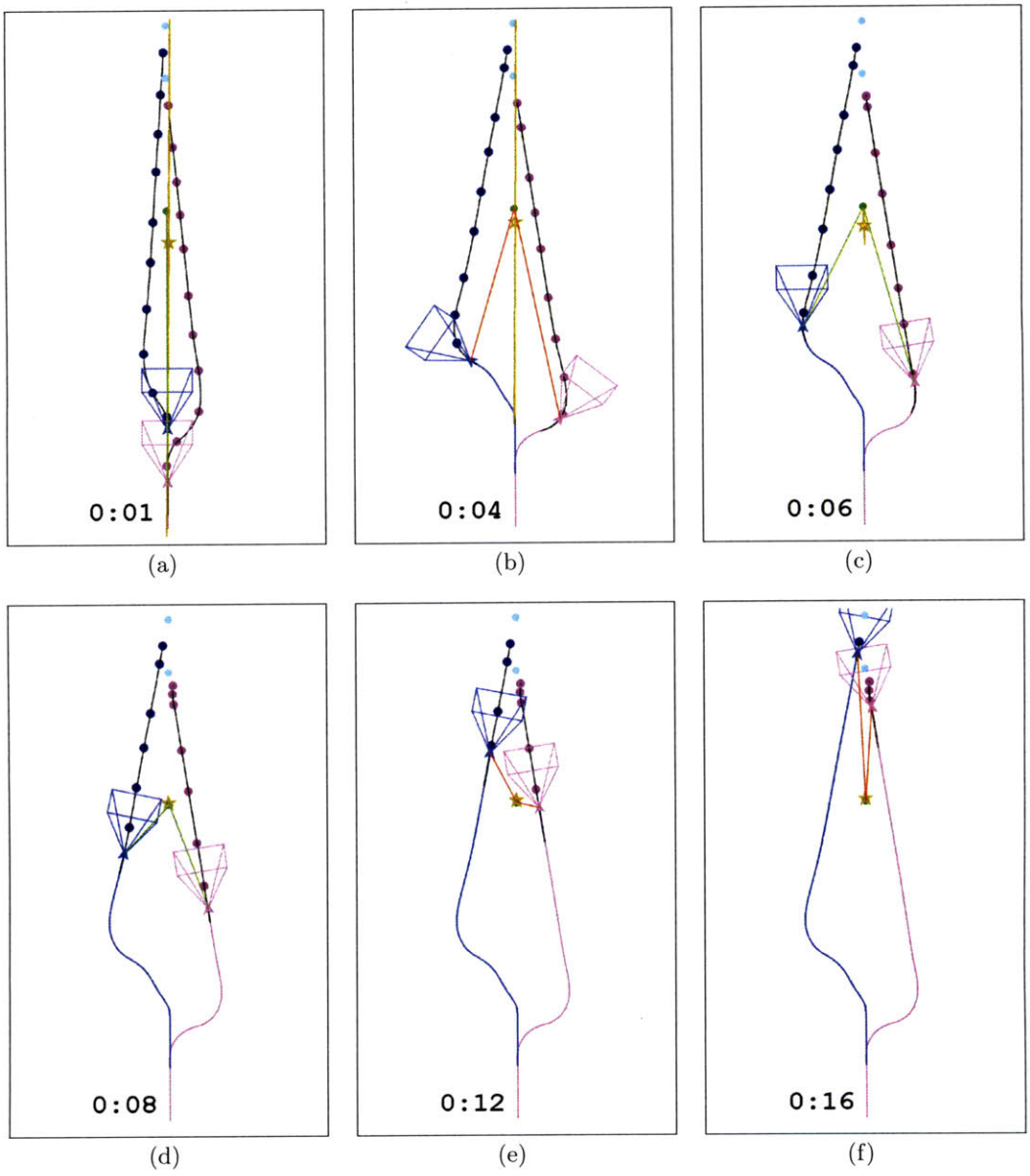
$$\mathbf{x}_0^{[1]} = \begin{bmatrix} 0.0 & -3.8 & 1.0 \end{bmatrix}^T \text{ m} \qquad \mathbf{x}_0^{[2]} = \begin{bmatrix} 0.0 & -3.0 & 1.0 \end{bmatrix}^T \text{ m},$$

and goal positions

$$\mathbf{x}_g^{[1]} = \begin{bmatrix} 0.0 & 3.0 & 1.0 \end{bmatrix}^T \text{ m} \qquad \mathbf{x}_g^{[2]} = \begin{bmatrix} 0.0 & 3.8 & 1.0 \end{bmatrix}^T \text{ m},$$

to minimize the individual agent cost functions, subject to $\alpha_{\mathcal{I}}^{[1]} = \alpha_{\mathcal{I}}^{[2]} = 1900 \text{ sm}^{-2}$.

An example trial for a such a scenario is depicted in Figure 5-1. Initially, in Figure 5-1, the target position estimate is close to the true value, but the highly eccentric uncertainty ellipse is directed along the line-of-sight from both vehicles (Figure 5-1(a)). Recall that the path planning modules are decentralized but assume a central measurement processing module for quantifying the information content of agents' paths. Based on the evolving target estimate, the vehicles individually plan paths that increase the difference in bearing between the two measurement sets subject to the other agent's announced plan. Specifically, the path selected (Figures 5-1(b-d)) is one that balances deviation from the centerline (which forms the minimal-time path for each) with time spent triangulating the target. As the joint maneuver is sufficiently information-rich, when the target leaves the line of sight of both vehicles (Figure 5-1(e)), the remaining path segments connecting each agent to the goal are followed (Figure 5-1(f)).



(a) (b) (c) (d) (e) (f)

Figure 5-1: A multiagent Dubins scenario with sensor constraints. Each agent plans paths that minimize the goal arrival time and maximize the shared information. The ability for both agents to simultaneously take measurements from disparate measurement poses, as in (c) and (d), is considered favorable by the cost function of each agent's path planning module.

5.3.2 Multitarget Scenario

A three-agent, eight-target scenario is now considered. Specifically, the performance of multiagent IRRT is compared for two planning modes. In both modes, agents report measurements to a decentralized information filter, consistent with the assumptions of Section 5.1. The modal distinction arises from the treatment of plan information contributions of agents in the network. In the *non-cooperative* mode, when an individual agent plans its path, the plan information contribution of all other agents in the network is ignored. Alternatively, in the *cooperative* mode, an individual agent fully utilizes, as in the algorithm developed in Section 5.2, the plan information contribution of the other agents in the network.

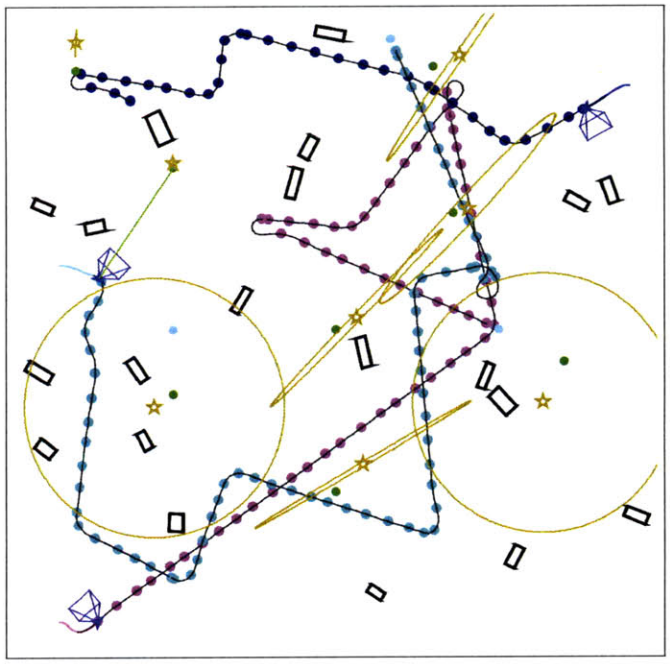
The scenario environment, which is identical for all trials, consists of an axis-aligned box in the first octant of $\mathbb{R}^3$ with dimensions $(20,\ 20,\ 6)^T$ m. The cube is populated by 20 randomly generated box obstacles whose placements are uniformly sampled within the environment, and whose lengths and widths are uniformly selected from the intervals $[0.5, 1]$ m and $[0.25, 0.5]$ m, respectively.

The true positions of the eight targets are given in Table 5.3.2. The initial estimate for each target f is random for each trial and is generated by perturbing the true poisitions according to

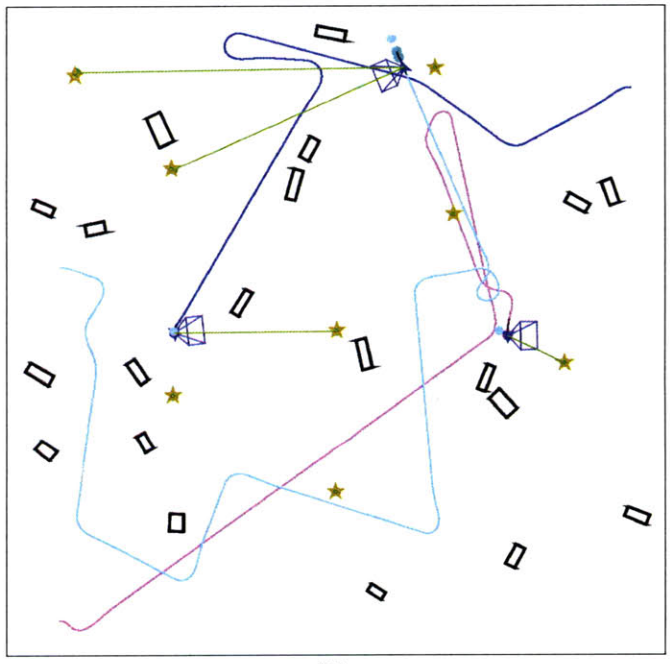
$$\hat{\mathbf{x}}_f(0) = \mathbf{x}_f + \mathbf{d}_f, \quad \mathbf{d}_f \sim \mathcal{N}\left(\mathbf{0}, \sigma_d^2 I_3\right), \quad \sigma_d = 0.5 \text{ m}. \tag{5.4}$$

All target covariances are initialized as $P_f(0) = \sigma^2 I_3, \sigma = 2.0$ m.

Each agent is a Dubins car with a diameter of 0.8 m and a monocular (bearings-only) sensor. The components of the initial and goal states for each agent are specified in Table 5.3.2. The sensor, which operates at 15 Hz and has 60° vertical and horizontal fields of view, is yawed 90° counterclockwise from the front of the car and pitched up 30° from the plane. Simulated bearing measurements are corrupted by a zero-mean, additive white Gaussian noise sequence with a standard deviation of 5°. The parameters of the cost function (3.26) are set to $\alpha = 0.5$ and $\alpha_{\mathcal{I}} = 3000$ sm^{-2} for all agents.

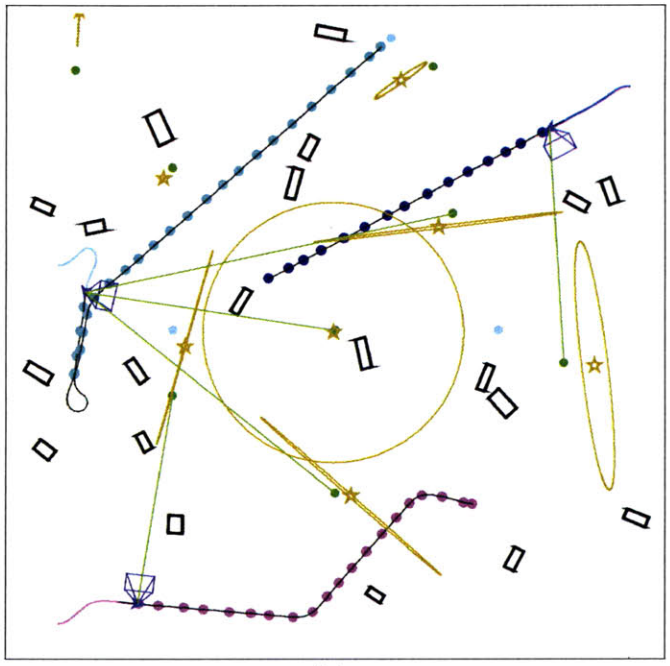


(a)

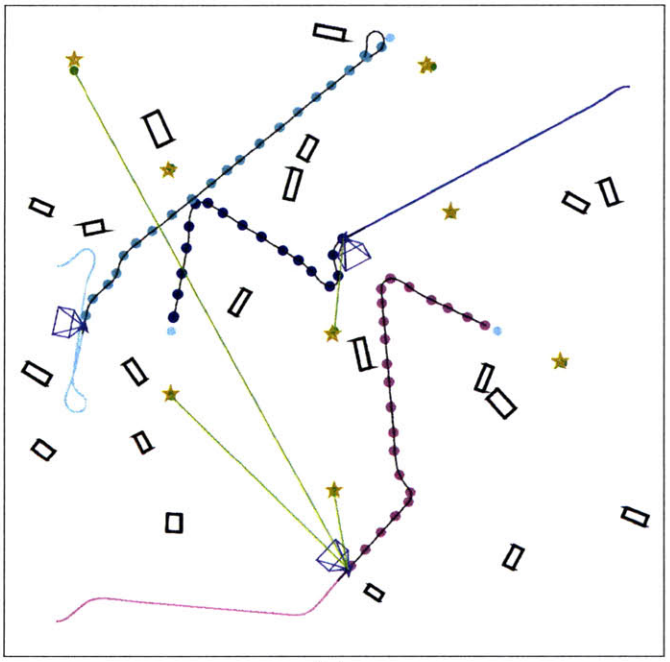


(b)

Figure 5-2: Sample non-cooperative multiagent IRRT scenario. (a) Each agent, ignoring the plan information content of the other agents, plans a circuitous path through the environment to collect information from all targets. (b) The resulting state history of the three agents at mission termination.



(a)



(b)

Figure 5-3: Sample cooperative multiagent IRRT scenario. (a) Each agent, accounting for the plan information content of the other agents, plans an efficient, information-rich path through the environment to cooperatively collect information from all targets. (b) The resulting state history of the three agents at mission termination.

Table 5.1: Multiagent scenario: target positions.

Target	x_f [m]	y_f [m]	z_f [m]
1	10.0	5.0	3.0
2	5.0	15.0	3.0
3	17.0	9.0	3.0
4	10.0	10.0	3.0
5	13.0	18.1	3.0
6	5.0	8.0	3.0
7	13.6	13.6	3.0
8	2.0	18.0	3.0

Table 5.2: Multiagent scenario: agent initial and goal states.

Agent	$\psi_0^{[q]}$ [rad]	$x_0^{[q]}$ [m]	$y_0^{[q]}$ [m]	$z_0^{[q]}$ [m]	$x_g^{[q]}$ [m]	$y_g^{[q]}$ [m]	$z_g^{[q]}$ [m]
1	0.0	1.5	1.0	1.0	15.0	10.0	1.0
2	3.1	19.0	17.5	1.0	5.0	10.0	1.0
3	0.0	1.5	12.0	1.0	11.7	19.0	1.0

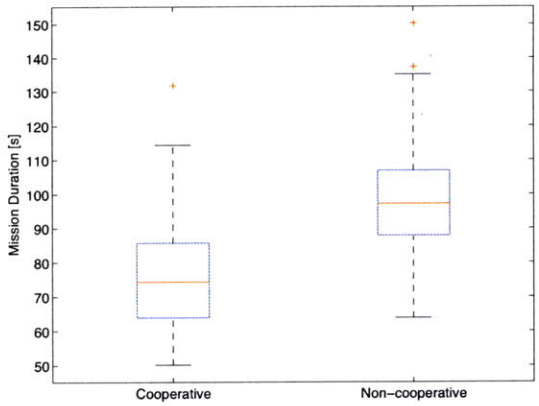
The qualitative behavior of the non-cooperative and cooperative modes is illustrated in Figures 5-2 and 5-3, respectively, for example trials. Typically, agents in the non-cooperative mode commit to path plans that are significantly longer than those selected by agents in the cooperative mode. While measurements taken by all agents reduce the uncertainty (hence, information cost) apparent to a particular agent, the inability to anticipate the plan information contribution of other agents leads each agent to selecting a (possibly circuitous) path for the purposes of collecting information from all targets.

Recall that in the IRRT algorithm, the relative weighting between the information collection and path duration is $\alpha_{\mathcal{I}}$. Thus, in order to assess the mission performance, a mission-level cost $C = d + \alpha_{\mathcal{I}}\mathcal{I}_{term}$ is specified, where d is the mission duration and $\mathcal{I}$ is the terminal A-optimality cost. A network of agents that plan in the non-cooperative mode typically gathers more information over the course of a mission, but does so at the expense of significantly longer mission durations. One would, therefore, expect the resultant mission-level cost to be higher in the non-cooperative mode. To better quantify this statement for multiagent IRRT, a randomized algorithm, the

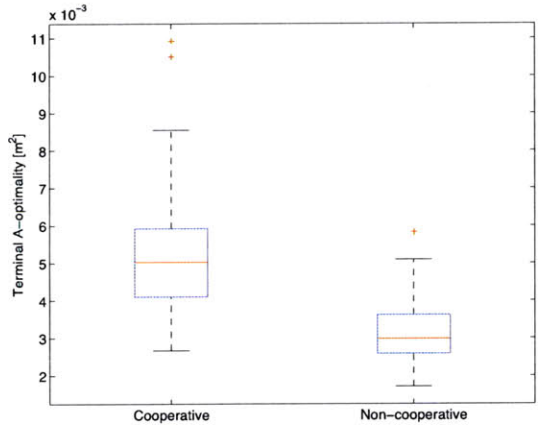
Table 5.3: Tabulated results of the batch multiagent IRRT comparison of the cooperative and non-cooperative planning modes. The total cost is computed as $c = \Delta t + \alpha_{\mathcal{I}} \mathcal{I}_{term}$, where Δt is the mission duration, $\mathcal{I}_{term}$ is the final A-optimality cost, and $\alpha_{\mathcal{I}} = 3000$ sm^{-2}.

Metric		Cooperative	Non-cooperative
Mission Duration [s]	median	74.29	97.11
	mean	75.63	98.00
	std	14.35	15.93
	IQR	[63.96, 85.63]	[87.6, 106.8]
Terminal Information Cost [10^{-3} m^2]	median	5.02	2.99
	mean	5.20	3.15
	std	1.50	0.79
	IQR	[4.11, 5.92]	[2.59, 3.62]
Total Mission Cost [s]	median	88.36	105.07
	mean	91.23	107.46
	std	12.92	14.56
	IQR	[80.77, 99.86]	[97.69, 115.90]

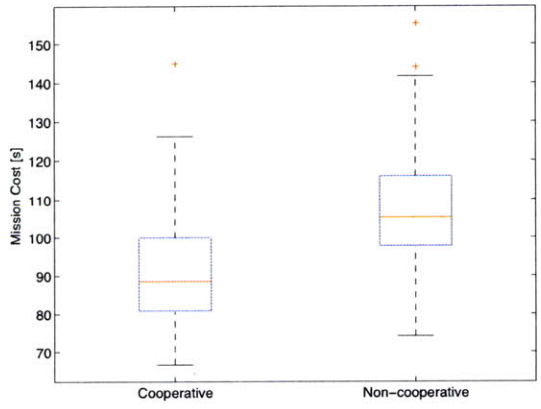
performance of the non-cooperative and cooperative modes are compared over a set of 100 trials of each. The statistical results can be found in Table 5.3 and in Figure 5-4. As expected, the cooperative mode generally outperforms the non-cooperative mode, where the severity of underperformance in the latter is a function the scenario and mission parameters, particularly the information cost weight $\alpha_{\mathcal{I}}$ in (3.26).



(a)



(b)



(c)

Figure 5-4: Boxplot of the batch multiagent IRRT scenario. (a) The mission duration for the cooperative multiagent IRRT mode is significantly shorter, on average, than that of the non-cooperative mode. (b) The extended duration of non-cooperative missions generally results in more information collected, (c) though the overall cost may yet be adversely affected.

Chapter 6

IRRT with Gaussian Mixture Models

In this chapter, we relax the Gaussian assumption on the prior distribution $p_{\mathbf{x}_f}(\mathbf{x}_f)$ of unknown feature vector $\mathbf{x}_f$ and the posterior distribution $p_{\mathbf{x}_f|Z}(\mathbf{x}_f|Z)$ of $\mathbf{x}_f$ given the measurement sequence Z. Though many particle filtering techniques have been developed to approximate general posterior distributions [55], the associated computational complexity currently bars approximations that are temporally distant. The approach pursued here is concurrent with the Gaussian mixture model framework, which seeks to approximate the posterior distribution with a "small" number of Gaussian basis functions. By "small," it is meant that the number of bases n_G is between two and twenty, in contrast to the number of particles (on the order of 10^3) usually employed in particle filtering implementations.

To simplify the notation of this chapter, we will refer to only a single target whose true position is estimated by a single agent. The parallelization to multiple targets is exactly as in Section 3.4, and the extension to multiple agents follows a nearly identical derivation to that offered in Chapter 5. Furthermore, both probability mass and density functions will be denoted by $p(\cdot)$; distinction between the two will be apparent from the context.

The chapter proceeds by reviewing the static multiple-model form of the Gaussian sum filter (Section 6.1), applying the filtering methods to the information quantifica-

tion step of IRRT (Section 6.2), and offering simulation results for select multi-modal problems (Section 6.3).

6.1 Gaussian Sum Filter

The Gaussian sum filter, first proposed in [6] and later pursued in [4, 62, 63], approximates the posterior distribution $p(\mathbf{x}_k|Z_k)$ by a *Gaussian mixture*, a weighted sum of n_G Gaussian density functions. Specifically,

$$p(\mathbf{x}_k|Z_k) \approx p_{GSF}(\mathbf{x}_k|Z_k) = \sum_{i=1}^{n_G} w_{k,i}\, \mathcal{N}(\mathbf{x}_{k,i}; \overline{\mathbf{x}}_{k,i}, P_{k,i}) \tag{6.1}$$

$$\text{subject to } \sum_{i=1}^{n_g} w_{k,i} = 1 \tag{6.2}$$

$$w_{k,i} \geq 0, \forall i. \tag{6.3}$$

The individual Gaussians in (6.1) are essential basis functions parameterized by means $\overline{\mathbf{x}}_{k,i}$ and covariances $P_{k,i}$. The participation of each basis function in forming the approximate posterior $p_{GSF}(\mathbf{x}_k|Z_k)$ is governed by its weight $w_{k,i}$. The constraints (6.2)-(6.3) guarantee that $p_{GSF}(\mathbf{x}_k|Z_k)$ is a valid probability distribution (i.e., it is nonnegative and integrates to unity). Note that although (6.3) is not strictly required, extra measures must be taken to ensure the nonnegativity of the distribution function over its support [30].

Various forms of the Gaussian sum filter are distinguished by their methods for computing the constituent weights $w_{k,i}$, means $\overline{\mathbf{x}}_{k,i}$, and covariances $P_{k,i}$ online. The approach followed here is that of the static multiple-model estimator (SMME) [4], which is used in problems where the system follows one of n_G possible (nonlinear) models, though it is not known which model is correct. The SMME is comprised of a bank of n_G nonlinear filters, each matched to a particular model. Typical choices for these so-called matched filters are extended Kalman filters (EKFs) or unscented Kalman filters (UKFs).

Let i denote the index of the filter in the bank. Each filter is initialized with a

mean $\hat{\mathbf{x}}_{0,i}^{\ominus}$, covariance $P_{0,i}^{\ominus}$, and a weight $w_{0,i}$. Thereafter, when a new measurement $\mathbf{z}_k$ arrives at time step k, it is disseminated to and individually processed by each of the n_G filters, yielding locally updated estimates $\hat{\mathbf{x}}_{k,i}^{\oplus}$ and covariances $P_{k,i}^{\oplus}$. In composing the mixture model of (6.1), the mean $\overline{\mathbf{x}}_{k,i}$ and covariance $P_{k,i}$ of each Gaussian are selected as

$$\overline{\mathbf{x}}_{k,i} := \hat{\mathbf{x}}_{k,i}^{\oplus} \tag{6.4}$$

$$P_{k,i} := P_{k,i}^{\oplus}. \tag{6.5}$$

Next, for each filter, the innovation $\mathbf{i}_{k,i} \triangleq \mathbf{z}_k - \hat{\mathbf{z}}_{k,i}$ and innovation covariance $S_{k,i} \triangleq \mathbb{E}\left[\mathbf{i}_{k,i}\mathbf{i}_{k,i}^T\right]$ is computed. At this point, each filter reports $\overline{\mathbf{x}}_{k,i}$, $P_{k,i}$, $\mathbf{i}_{k,i}$, and $S_{k,i}$ to a fusion center reponsible for updating the weights $w_{k,i}$ and forming the mean $\overline{\mathbf{x}}_k$ and covariance P_k for the Gaussian mixture model.[1]

The mixture mean is simply a weighted sum of the constituent means

$$\hat{\mathbf{x}}_k = \sum_{i=1}^{n_G} w_{k,i}\overline{\mathbf{x}}_{k,i}. \tag{6.6}$$

The covariance may be computed as

$$P_k = \sum_{i=1}^{n_G} w_{k,i}\left[P_{k,i} + (\overline{\mathbf{x}}_{k,i} - \overline{\mathbf{x}}_k)(\overline{\mathbf{x}}_{k,i} - \overline{\mathbf{x}}_k)^T\right]. \tag{6.7}$$

The weights are then updated in two steps. The conditional likelihood

$$\beta_{k,i} = \mathcal{N}\left(\mathbf{i}_{k,i}; \mathbf{0}, S_{k,i}\right) \tag{6.8}$$

that filter i captures the correct Gaussian distribution is computed and used as a

[1]The use of the $\ominus$ and $\oplus$ notation is not neccessary for the mixture mean and covariance, as these quantities are not propagated between time steps and are formed directly from the post-measurement mean and covariance of the constituent models.

Algorithm 6 Static Multiple-Model Estimator (SMME)

for $i = 1, \ldots, n_G$ **do**
  Initialize $\hat{\mathbf{x}}_{1,i}^{\ominus}$, $P_{1,i}^{\ominus}$, $w_{1,i}$
end for
$Z_0 \leftarrow \emptyset$
for $k = 1, 2, \ldots$ **do**
  Approximate $p(\mathbf{x}_k|Z_{k-1})$ using (6.1)
  $Z_k \leftarrow Z_{k-1} \cup \mathbf{z}_k$
  for $i = 1, \ldots, n_G$ **do**
    Perform measurement update using $\mathbf{z}_k$, $\hat{\mathbf{x}}_{k,i}^{\ominus}$, and $P_{k,i}^{\ominus} \Rightarrow \hat{\mathbf{x}}_{k,i}^{\oplus}$, $P_{k,i}^{\oplus}$
    $\overline{\mathbf{x}}_{k,i} \leftarrow \hat{\mathbf{x}}_{k,i}^{\oplus}$
    $P_{k,i} \leftarrow P_{k,i}^{\oplus}$
    Form innovation $\mathbf{i}_{k,i}$ and innovation covariance $S_{k,i}$
  end for
  Compute mixture mean $\overline{\mathbf{x}}_k$ using (6.6)
  Compute mixture covariance P_k using (6.7)
  Update weights according to (6.8)-(6.9)
end for

sub-weighting factor in updating the filter weights $w_{k,i}$ via

$$w_{k+1,i} = \frac{w_{k,i}\beta_{k,i}}{\sum_{j=1}^{n_G} w_{k,j}\beta_{k,j}}. \tag{6.9}$$

The general procedure is given in Algorithm 6.

6.2 Application

The general SMME of Algorithm 6 is now applied to the information quantification sub-problem of IRRT when the prior or posterior distributions are approximated by Gaussian mixture models. As computational complexity at each time step is a fundamental concern for roadmap-based methods, the constituent filters of the SMME are implemented as EKFs. The EKF equations are split among a propagation phase

$$\hat{\mathbf{x}}_{k,i}^{\ominus} = \Phi_{k|k-1,i}\hat{\mathbf{x}}_{k-1,i}^{\oplus} \tag{6.10}$$

$$\hat{\mathbf{z}}_{k,i}^{\ominus} = \mathbf{h}_i\left(\hat{\mathbf{x}}_{k,i}^{\ominus}\right) \tag{6.11}$$

$$P_{k,i}^{\ominus} = \Phi_{k|k-1,i}P_{k-1,i}^{\oplus}\Phi_{k|k-1,i}^{T} + Q_{k-1,i}, \tag{6.12}$$

a linearization phase

$$H_{k,i} = \left.\frac{\partial \mathbf{h}_i}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k,i}^{\ominus}}, \tag{6.13}$$

and an update phase

$$\mathbf{i}_{k,i} = \mathbf{z}_k - \hat{\mathbf{z}}_{k,i}^{\ominus} \tag{6.14}$$

$$K_{k,i} = P_{k,i}^{\ominus} H_{k,i}^T \left[H_{k,i} P_{k,i}^{\ominus} H_{k,i}^T + R_{k,i}\right]^{-1} \tag{6.15}$$

$$\hat{\mathbf{x}}_{k,i}^{\oplus} = \hat{\mathbf{x}}_{k,i}^{\ominus} + K_{k,i}\mathbf{i}_{k,i} \tag{6.16}$$

$$P_{k,i}^{\oplus} = P_{k,i}^{\ominus} - P_{k,i}^{\ominus} H_{k,i}^T \left[H_{k,i} P_{k,i}^{\ominus} H_{k,i}^T + R_{k,i}\right]^{-1} H_{k,i} P_{k,i}^{\ominus} \tag{6.17}$$

$$S_{k,i} = H_{k,i} P_{k,i}^{\oplus} H_{k,i}^T + R_{k,i}, \tag{6.18}$$

where for the i^{th} matched model, $\Phi_{k|k-1,i}$ is the (linear) state transition matrix from time step k to $k+1$, $\mathbf{h}_i$ is the (nonlinear) measurement model, $H_{k,i}$ is the linearized measurement matrix, and $Q_{k-1,i}$ and $R_{k,i}$ are the process and sensor noise covariance matrices. The local estimate $\overline{\mathbf{x}}_{k,i} := \hat{\mathbf{x}}_{k,i}$, covariance $P_{k,i} := P_{k,i}^{\oplus}$, innovation $\mathbf{i}_{k,i}$, and innovation covariance $S_{k,i}$ are reported to a fusion center as in (6.6)-(6.9).

Consider a single agent quantifying the tree information content about a single target.[2] Assume the quantification occurs at time step k, at which time the estimate $\overline{\mathbf{x}}_{k,i}$, covariance $P_{k,i}$, and relative weight $w_{k,i}$ are known for each of the n_G filters. As the Gaussian sum filter attempts to model the prior and posterior distributions as a Gaussian mixture, the information quantification step should capture the covariance reduction in the constituent models due to sensing along paths in the tree. The basic idea is to maintain n_G FIMs in each node. Because the prior distribution is modeled as a Gaussian mixture, the FIM at root node for each mode i is initialized as

$$J_{root,i} = P_{k,i}^{-1}. \tag{6.19}$$

Thereafter, the recursive update of the FIM $J_{N,i}$ for node N and mode i follows (3.18).

The node information cost $\mathcal{I}_N$, which is required in both the cost function (3.26) and the INFORMATION nearest node heuristic (3.24), uses the total node CRLB J_N^{-1}.

[2]The extensions to multiple agents and targets are straightforward and closely follow the derivations in Sections 3.4 and 5.2, respectively.

Towards calculating J_N^{-1}, an approximate lower bound on the CRLB is pursued. Recall (6.7), repeated here as

$$P_k = \sum_{i=1}^{n_G} w_{k,i} \left[P_{k,i} + (\overline{\mathbf{x}}_{k,i} - \overline{\mathbf{x}}_k)(\overline{\mathbf{x}}_{k,i} - \overline{\mathbf{x}}_k)^T \right]$$

It is apparent that

$$\tilde{P}_{k,i} \triangleq (\overline{\mathbf{x}}_{k,i} - \overline{\mathbf{x}}_k)(\overline{\mathbf{x}}_{k,i} - \overline{\mathbf{x}}_k)^T \geq \mathbf{0}. \tag{6.20}$$

Therefore,

$$P_k \geq \sum_{i=1}^{n_G} w_{k,i} P_{k,i}. \tag{6.21}$$

By fixing the weights $w_{k,i}$ to their values at time step k when the information quantification query began, an estimate of the Cramér-Rao Lower Bound is

$$J_N^{-1} = \sum w_{k,i} J_{N,i}^{-1}. \tag{6.22}$$

Note that this approach necessarily neglects two quantities that cannot be predicted: the evolution in weights $w_{k,i}$ (according to the innovation processes) and the difference $(\overline{\mathbf{x}}_{k,i} - \overline{\mathbf{x}}_k)$ between the constituent model means $\overline{\mathbf{x}}_{k,i}$ and the mixture mean $\overline{\mathbf{x}}_k$.

In order to mitigate the effects of a diverging filter on the mixture model, heuristics for zeroing the weight must be employed. One such heuristic is suggested in [5]. Another heuristic, as in [25], compares the largest eigenvalues of the current and initial covariance matrices for each model, zeroing the model weight if the former exceeds the latter multiplied by a constant. Compactly, the heuristic is

$$w_{k,i} := 0 \quad \text{if} \quad \max_j \lambda_j(P_{k,i}^{\ominus}) > \alpha \max_j \lambda_j(P_{0,i}), \tag{6.23}$$

where the ratio α is typically chosen to be $\alpha \approx 3$. To avoid computing the eigenvalues

of the covariance matrices, a similar heuristic can be implemented as

$$w_{k,i} := 0 \quad \text{if} \quad \texttt{trace}(P_{k,i}^{\ominus}) > \alpha \, \texttt{trace}(P_{0,i}). \tag{6.24}$$

One caveat is of note in characterizing the performance of IRRT in the presence of multi-modal distributions. As mentioned in Section 3.4.1, given a target model, the information content of a node is quantified from the set of successful anticipated measurements, where success is determined by the anticipated visibility/occlusion of the point coinciding with the *model mean.* Intuitively, if a model mean is located at a particular point whose (affirmative) visibility is anticipated, then the *absence* of the target in the field of view when directed at that point should provide some "negative information" on the correctness of particular target models. However, measurements are only delivered to the filter, and, hence, model weights are only redistributed, when the actual target is visible to the sensor. To account for "negative information," one would require an additional inference algorithm that operates on the disparity between anticipated visibility and absent measurements. The design of such an algorithm is both complex and outside the scope of this thesis.

6.3 Simulation Results

As aforementioned, Gaussian mixture models are often used to approximate multi-modal prior distributions. In the following scenario, the prior distribution on the position of a single stationary target can be well approximated by two Gaussian modes whose relative contributions to the mixture model comprise a single parameter in the scenario. Moreover, one of the modes is "correct," in the sense that its mean is based on a perturbation from the true target position, and the remaining mode is fictitious. This situation arises in practice when, for example, sources report a single target to be located in mutually exclusive zones. The degree to which the mixture model captures the modal correctness will, along with the geometry of the problem, be demonstrated to significantly influence the performance of multimodal

IRRT. In fact, the multimodal IRRT performance is reminiscent of multitarget IRRT, yet an agent can often be led to waste time searching for a false target mode. Since the Gaussian sum filter, upon finally discovering the target, converges to the single correct mode, scenarios involving multiple targets or multiple agents, while providing no fundamental challenge to the planner, do not as effectively illustrate the central issues of multimodal IRRT. Therefore, simulation results for a simple single-target, single-agent scenario are now provided.

Consider an empty, axis-aligned box environment in $\mathbb{R}^3$ whose bounding vertices have coordinates $(-3.0, -4.0, 0.0)^T$ m and $(6.0, 4.0, 6.0)^T$ m. Within this environment, a Dubins car agent with initial position $\mathbf{x}(0) = (-1.0, -3.8, 1.0)^T$ m and heading $\psi(0) = \pi/2$ rad must arrive at the goal position $\mathbf{x}_g = (4.6, 2.0, 1.0)^T$ m. The car is equipped with a monocular (bearings-only) sensor, which operates at 15 Hz, has 60° vertical and horizontal fields of view, is yawed 90° counterclockwise from the front of the car, and is pitched up 30° from the plane. Simulated bearing measurements are corrupted by a zero-mean, additive white Gaussian noise sequence with a standard deviation of 8°. The parameters of the cost function 3.26 are set to $\alpha = 0.5$ and $\alpha_{\mathcal{I}} = 2500$ sm^{-2}.

The true target location is $\mathbf{x}_f = (0.5, -1.0, 3.0)^T$ m. The mean of the "correct" mode (with index, say, $i = 1$) is generated at time step $k = 0$ by randomly perturbing the true target position according to

$$\hat{\mathbf{x}}_{0,1} = \mathbf{x}_f + \mathbf{d}_f, \quad \mathbf{d}_f \sim \mathcal{N}\left(\mathbf{0}, \sigma_d^2 I_3\right), \quad \sigma_d = 0.5 \text{ m}. \tag{6.25}$$

A fictitious mode associated with a second model (with index $i = 2$) is initialized as $\hat{\mathbf{x}}_{0,2} = (-0.5, 2.0, 2.0)^T$ m. Both models are initialized with covariance matrices $P_{0,i} = I_3$ m^2.

Recall from (6.22) that the model weights w_i directly influence the information quantification step of multimodal IRRT. Therefore, an agent may select a path based on some anticipated information gain that, due to a poor initial weight distribution, is little realized. The scenario described above is simulated in Figures 6-1, 6-2, and

6-3 for the weight ratios 9:1, 5:5, and 1:9, respectively, between the correct and incorrect models. In general, the lower the weighting on the correct model, the worse the mission-level performance; this observation is often tempered by the geometry of the problem. In Figure 6-1, the initial weightings on the correct and fictitious modes are 0.9 and 0.1, respectively. The initial plan calls for a close pass on the heavily weighted mode, whereupon early measurements of the target all but eliminate the fictitious mode. In Figure 6-2, the prior weighting is equal for the correct and fictitious modes. As a result, the agent plans paths that ration measurements between the two modes. In this particular example, measurements on the true target do not occur until the agent has moved in a direction away from the goal, after which its best course of action is to make several turns and close passes around the targets. Finally, in Figure 6-3, the initial weightings on the correct and fictitious modes are 0.1 and 0.9, respectively. While the initial plans all but ignore the correct mode, focusing largely on the fictitious mode, a measurement of the true target taken at close range is enough to effectively snap to the correct mode. Conditioned on this short measurement sequence, the vehicle ammends its plan to a close sweep near the target and directed towards the goal.

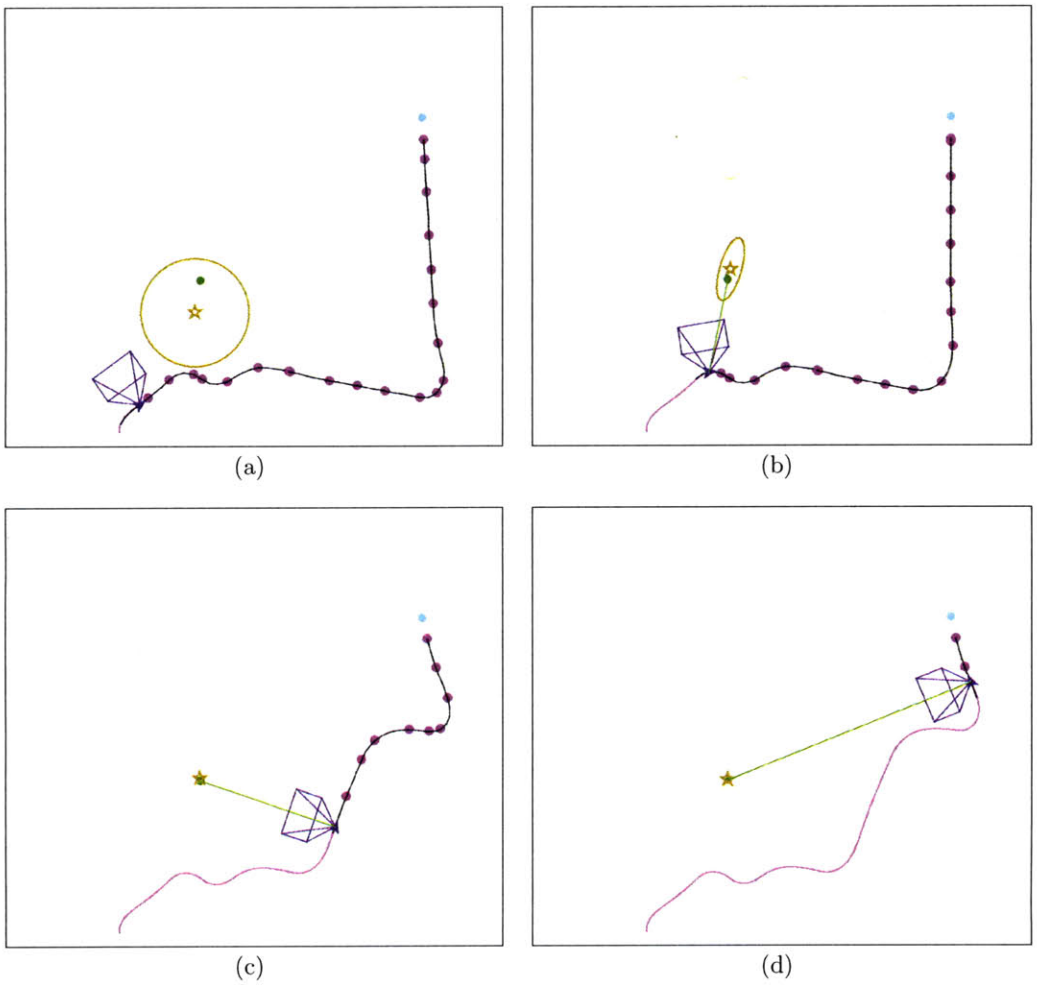


Figure 6-1: Multimodal IRRT scenario with a favorable prior. The mixture model is a composition of two modes, each with a mean (gold star) and (gold) uncertainty ellipse. The correct mode, near the true target position (green dot), is weighted at 0.9 in the prior. The fictitious mode in the top left is mostly transparent, owing to its comparatively lower weight of 0.1. (a) The initial plan calls for a close approach on the high-weight mode, followed by an orthogonalizing straight-line trajectory that only observes the low-weight mode near termination. (b) Upon observing the true target, uncertainty ellipse of each mode loses volume, yet the fictitious mode begins converging to the correct mode. (c,d) Having converged to a single-mode, the planner ammends its path so as to gather more information as the vehicle approaches the goal.

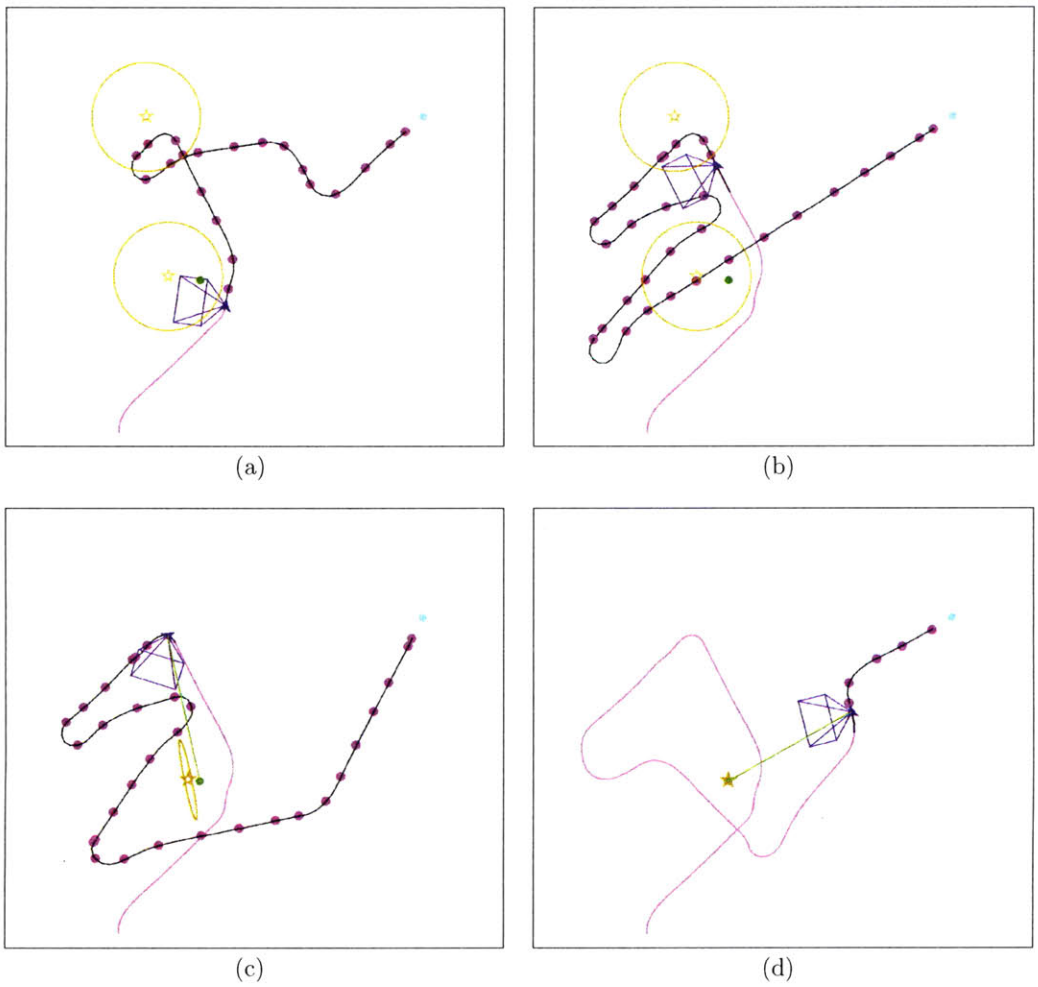


Figure 6-2: Multimodal IRRT scenario with a symmetric prior. The mixture model is a composition of two equally weighted modes. (a) The initial plan calls for one close approach and one distant triangulation maneuver for each mode, with an approximately equal amount of time spent observing each. (b) The plan is ammended to a "lawnmower" pattern that alternates between looking at either of the modes. (c) Having observed the true target, the estimate converges to a single mode. (d) The vehicle drives a path around the target so as to gather information on the way to the goal.

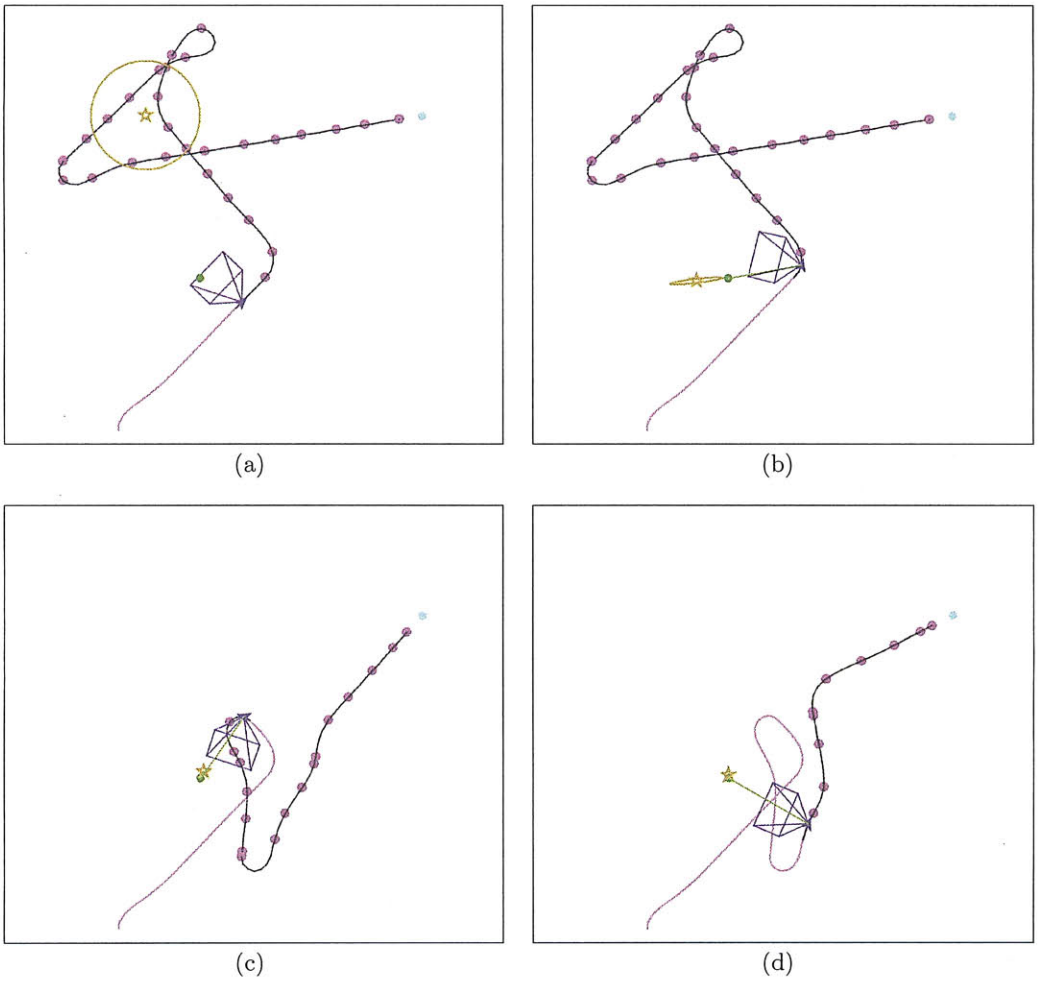


Figure 6-3: Multimodal IRRT scenario with an unfavorable prior. The correct mode, near the true target position, is weighted at 0.1 in the prior, while the fictitious mode in the top left is initialized with weight 0.9. (a) The initial plan pays little concern for the correct mode, with only sporadic visibility of the mean (transparent gold star). (b) Upon unexpectedly observing the true target at (fortunately) close range, the uncertainty ellipse shrinks significantly and the fictitious mode collapses. (c) Conditioned on the new target estimate, the vehicle ammends its path to perform a quick, parallax-building maneuver near the target, (d) after which it proceeds to the goal while keeping the target in sight.

Chapter 7

Conclusion

7.1 Summary

This thesis considers the problem of planning informative paths online and under general constraints. This problem is motivated by mobile intelligence, surveillance, and reconnaissance missions involving autonomous platforms, each with its own dynamic, sensing, and environmental constraints, performing localization on stationary targets. Previous research has used solution strategies that are not amenable to the whole of these constraints; for example, both single- and multi-step receding horizon optimization techniques lack robustness to sensing occlusions, and heuristic path shapes may be rendered infeasible by obstacles in the environment. Alternatively, problem formulations that afford very general constraint characterizations also have profound computational issues that currently render them intractable for use, for example, on vehicles with complex dynamic models.

The primary contribution of this thesis is the Information-rich Rapidly-exploring Random Tree (IRRT) algorithm, an extension of the closed-loop RRT algorithm that embeds information quantification as predicted using Fisher information matrices. In Chapter 2, various information-theoretic measures used in the information-rich planning and sensor management literature were reviewed. The most general such measures, conditional entropy, mutual information, and divergence, though capable of examining entire probability distributions, require posterior distribution approxi-

mations, which may be intractable only several steps into the future. An alternative is found in Fisher information, a measure of the information content in an observation about a particular stochastic process. Fisher information has a close relationship to the Cramér-Rao lower bound (CRLB) on the estimation error covariance for unbiased estimates. Therefore, Fisher information matrices (FIMs) may be used to characterize, independent of the (admissible) estimator used, the potential information richness of paths. For nonlinear process and sensing models subject to additive Gaussian noise, a recursive form of the FIM may be used for long-duration paths. For stationary targets, the recursion becomes simply additive.

Chapter 3 first motivates the selection of closed-loop RRT (CL-RRT) as a baseline algorithm and then details the extension to IRRT. In this way, all the beneficial properties of CL-RRT are preserved while enabling and embedding information richness at the planning level. As it is a sample-based algorithm, feasible solutions can be easily generated in real-time, and the planner effectiveness scales with the available computational resources. As in the open-loop RRT, path cost calculations and constraint evaluations are performed trajectory-wise, allowing CL-RRT to easily handle complex constraints that may cause optimization-based approaches to become computationally intractable. Furthermore, the path-tracking control loop of CL-RRT permits accurate state prediction when following a specified path, with proven error bounds on tracking performance. The IRRT extension uses this state prediction to form the measurement pose sequence, the information content of which is quantified in the Fisher information framework. The embedding of information metrics in the tree allows for nearest node heuristics and cost functions that explicitly consider information in both the tree growth and execution phases.

Chapter 4 presents a progression of simulation results that demonstrate the typical performance of IRRT in constrained scenarios. The scenarios involve either Dubins cars or quadrotor helicopters, each carrying limited field-of-view monocular (bearings-only) sensors, attempting to localize targets in the environment and to arrive at a goal state. A mission-level analysis examines the mission duration and terminal average uncertainty in the target estimates, as parameterized by the information cost weight,

and suggests the use of such parametric plots in selecting the cost weight based on the mission profile. A separate analysis of the INFORMATION nearest node heuristic reveals that the overall mission cost, a linear combination of mission duration and terminal average uncertainty, can be reduced by embedding information collection at the tree expansion level. The sensitivity of the mission duration, terminal average uncertainty, and overall cost to the probability of selecting this heuristic is examined.

Extensions to IRRT for multiagent, decentralized planning and for planning in a multimodal belief space are presented in Chapters 5 and 6, respectively. The additivity of FIMs for stationary targets is exploited in developing a method by which an individual agent accounts for the anticipated information collection of other agents in the network. A token passing algorithm that enforces order on agent replanning is also provided. Simulation results demonstrate the level of cooperation between agents in reducing the target uncertainty in both small, uncluttered environments with a single target and large, cluttered environments with multiple targets. In order to assess the benefit of cooperation among the agents, a comparison between a *cooperative* mode, in which agents share processed measurement data and anticipate team information collection, and a *non-cooperative* mode, in which only the former occurs, is performed. The network of agents in the non-cooperative mode performs worse on a mission-level cost basis, collecting more information than in the cooperative mode, but only by committing to more circuitous paths. Finally, the extension of IRRT to multimodal beliefs is presented in the framework of the Gaussian mixture model (GMM) and the static multiple model estimator (SMME). As the constituent models of the GMM are Gaussian, the information quantification sub-routine of multimodal IRRT is a generalization of that presented in 3.4.1, where the quantification occurs for each model. The resultant information metric in the cost function is based on the weighted sum of the model covariances, where the weights are taken from an SMME that runs parallel to the planner. A simulation of multimodal IRRT for a single target with two priors (one correct and one fictitious) demonstrates the influence of both the prior weight distribution and initial estimate on the overall performance. Such a problem is analogous to the multitarget problem in standard IRRT but is complicated

by the belief in a mode that is fictitious and from which no measurements arrive.

7.2 Future Work

The IRRT algorithm and two extensions thereof have been introduced in this thesis. Preliminary simulation results provided in Chapter 4 and Sections 5.3 and 6.3 have demonstrated the utility of IRRT in information-rich planning problems subject to general constraints. However, several areas of further investigation must be pursued before definitive statements about the general performance of IRRT can be made.

One area of future work involves the information quantification sub-problem. Recall that the Fisher Information calculation of Section 3.4.1 uses only the mean of the estimation process to determine the anticipated visibility of the target at future measurement poses. Specifically, the visibility was a binary-valued function of the target position estimate, measurement pose, environment model, and sensor model, but *not* the target estimate covariance. One could instead imagine, at the expense of the computational efficiency, sampling the Gaussian target model to generate visibility candidate points; the visibility metric used in the information quantification could then be the mean of the binary success value of the candidate points. The interpretation of the visibility function would, thus, shift from a binary switch to a coefficient on the individual Fisher Information matrices.

Another argument for the further embedding of the covariance in the planning process involves the look-ahead nature of the IRRT, which was expressly designed to not be limited to myopic behaviors. If a temporally long plan is selected due primarily to the anticipated information content of measurements taken near the terminus of the plan, then the possibility remains that the target estimate covariance is sufficiently large, and the information collected will be far lower than what is anticipated. This suggests that the agent should only commit to a long-term plan if it is confident that the reward will be sufficiently large. In the case of measurements which occur at temporally distant locations, a large amount of time will have been expended before the agent realizes the actual reward. It remains to be seen if performing time-based

discounting on the trajectory Fisher information in the planning phase improves the general performance; such discounting schemes could be parameterized, for example, by a matrix norm on the error covariance. Similarly, as discussed in Section 5.2, the trajectory information content of another agent in the network may be subjected to time-based discounting, where the discount rate is a function of the path diversity in a given agent's tree.

In Chapter 5, the extension of IRRT to decentralized, multi-agent scenarios was presented. The fundamental assumption therein required the information content of a measurement by any agent to be disseminated to the rest of the network via some unspecified decentralized filtering algorithm. An extension of IRRT that incorporates the communication constraints of the agents in the network and anticipates communication loss due to interference or attenuation would further the applicability of IRRT to real-world autonomous intelligence, surveillance, and reconnaissance operations.

Finally, while this thesis has assumed the environment features to be stationary targets, the extension to moving targets with linear dynamics would be straightforward and further utilize the multi-modal extension of Chapter 6, for example, in the case of a ground target approaching an intersection.

Bibliography

[1] Air Force doctrine document 2-5.2: Intelligence, surveillance, and reconnaissance operations. Technical report, United States Air Force Doctrine Center, April 1999.

[2] Defense science board task force on precision targeting. Technical report, United States Department of Defense, Office of the Under Secretary of Defense for Acquisition, Technology and Logistics, 2001.

[3] Unmanned aircraft systems roadmap, 2005-2030. Technical report, Office of the United States Secretary of Defense, August 2005.

[4] D. L. Alspach and H. W. Sorenson. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Trans. on Automatic Control*, 17(4):439–448, 1972.

[5] B. Anderson and J. Moore. *Optimal Filtering.* New York: Prentice Hall, 1979.

[6] M. Aoki. Optimal Bayesian and min-max control of a class of stochastic and adaptive dynamic systems. In *Proc. IFAC Symposium on Systems Engineering*, pages 77–84, 1965.

[7] D. B. Barber, J. Redding, T. Mclain, R. Beard, and C. Taylor. Vision-based target geo-location using a fixed-wing miniature air vehicle. *Journal of Intelligent Robotics Systems*, 47(4):361–382, 2006.

[8] J. Barraquand and J.-C. Latombe. Robot motion planning: a distributed representation approach. *Int. Journal of Robotics Research*, 10:628–649, 1991.

[9] J. M. Bernardo and A. F. M. Smith. *Bayesian Theory.* Chichester: Wiley, 1994.

[10] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte. Information based adaptive robotic exploration. In *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, 2002.

[11] H.-L. Choi. *Adaptive Sampling and Forecasting with Mobile Sensor Networks.* PhD thesis, Massachusetts Institute of Technology, 2009.

[12] T. Cover and J. Thomas. *Elements of Information Theory.* New York: Wiley-Interscience, 2 edition, 2006.

[13] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive mobile robot navigation and mapping. *Int. Journal of Robotics Research*, 18:650–668, 1999.

[14] R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London, Series A: Mathematical and Physical Sciences*, 222:309–368, 1922.

[15] E. Frazzoli, M. A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *AIAA Journal of Guidance, Control, and Dynamics*, 25(1):116–129, January-February 2002.

[16] E. W. Frew. *Trajectory Design for Target Motion Estimation Using Monocular Vision.* PhD thesis, Stanford University, 2003.

[17] E. W. Frew. Information-theoretic integration of sensing and communication for active robot networks. In *Proc. 1st Int. Conf. on Robot Communication and Coordination*, pages 1–8, Piscataway, NJ, USA, 2007. IEEE Press.

[18] C. Goerzen, Z. Kong, and B. Mettler. A survey of motion planning algorithms from the perspective of autonomous uav guidance. *Journal of Intelligent Robotic Systems*, 57:65–100, 2010.

[19] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte. Information-theoretic control of multiple sensor platforms. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan, September 2003.

[20] R. He. *Semi-conditional Planners for Efficient Planning under Uncertainty with Macro-actions.* PhD thesis, Massachusetts Institute of Technology, 2010.

[21] R. He, S. Prentice, and N. Roy. Planning in information space for a quadrotor helicopter in a gps-denied environment. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 2008.

[22] A. Hero, D. Castanon, D. Cochran, and K. Kastella. *Foundations and Applications of Sensor Management.* Berlin: Springer, 2007.

[23] A. Hero, B. Ma, O. Michel, and J. Gorman. Alpha divergence for classification, indexing and retrieval. Technical Report 328, Communications and Signal Processing Laboratory (CSPL), Department of EECS, University of Michigan, 2001.

[24] G. M. Hoffman and C. J. Tomlin. Mobile sensor network control using mutual information methods and particle filters. *IEEE Trans. on Automatic Control*, 55:32–47, 2010.

[25] J. P. How. Massachusetts Institute of Technology, 16.322 Stochastic Estimation and Control, Course Notes, April 2010.

[26] J. P. How, B. Bethke, A. Frank, D. Dale, and J. Vian. Real-time indoor autonomous vehicle test environment. *IEEE Control Systems Magazine*, 28(2):51–64, April 2008.

[27] S. Huang, N. M. Kwok, G. Dissanayake, Q. Ha, and G. Fang. Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 2005.

[28] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robotics and Automation*, 12:566–580, 1996.

[29] A. T. Klesh, P. T. Kabamba, and A. R. Girard. Path planning for cooperative time-optimal information collection. In *Proc. American Control Conference*, pages 1991–1996, 2008.

[30] W. Koch and R. Klemm. Ground target track with STAP radar. *IEEE Proc. - Radar, Sonar Navigation*, 148(3):173–185, 2001.

[31] C. Kreucher, K. Kastella, and A. Hero. Sensor management using an active sensing approach. *Signal Processing*, 85:607–624, 2005.

[32] C. M. Kreucher, A. Hero, K. Kastella, and M. Morelande. An information based approach to sensor management in large dynamic networks. *Proc. of the IEEE*, 95(5):978–999, 2007.

[33] Y. Kuwata, G. A. Fiore, J. Teo, E. Frazzoli, and J. P. How. Motion planning for urban driving using RRT. In *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, pages 1681–1686, Nice, France, September 2008.

[34] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How. Real-time motion planning with applications to autonomous urban driving. *IEEE Trans. on Control Systems Technology*, 17(5):1105–1118, September 2009.

[35] Y. Kuwata, J. Teo, S. Karaman, G. Fiore, E. Frazzoli, and J. P. How. Motion planning in complex environments using closed-loop prediction. In *Proc. AIAA Guidance, Navigation, and Control Conference*, 2008.

[36] J. Kwak and P. Scerri. Path planning for autonomous information collecting vehicles. In *Proc. Int. Conf. on Information Fusion*, 2008.

[37] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical Report 98-11, Iowa State University, October 1998.

[38] J. E. Le Cadre and C. Jauffret. Discrete-time observability and estimability analysis for bearings-only target motion analysis. *IEEE Trans. on Aerospace and Electronic Systems*, 33(1):178–201, Jan 1997.

[39] J. P. Le Cadre. Optimization of the observer motion for bearings-only target motion analysis. In *Proc. 36th IEEE Conf. on Decision and Control*, pages 3126–3131, Dec 1997.

[40] J. P. Le Cadre and H. Gauvrit. Optimization of the observer motion for bearings-only target motion analysis. In *Proc. 1st Australian Data Fusion Symposium*, pages 190–195, Nov 1996.

[41] J. Le Ny and G. J. Pappas. On trajectory optimization for active sensing in gaussian process models. In *Proc. 48th IEEE Conf. on Decision and Control*, pages 6286–6292, December 2009.

[42] P. T. Liu. An optimum approach in target tracking with bearing measurements. *Journal of Optimization Theory and Applications*, 56(2):205–214, February 1988.

[43] B. D. Luders, S. Karaman, E. Frazzoli, and J. P. How. Bounds on tracking error using closed-loop rapidly-exploring random trees. In *Proc. American Control Conference (to appear)*, Baltimore, MD, USA, June-July 2010.

[44] M. Mandic and E. Frazzoli. Efficient sensor coverage for acoustic localization. In *Proc. 46th IEEE Conf. on Decision and Control*, pages 3597–3602, Dec. 2007.

[45] S. Martínez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42(4):661–668, April 2006.

[46] Y. Oshman and P. Davidson. Optimization of observer trajectories for bearings-only target localization. *IEEE Trans. on Aerospace and Electronic Systems*, 35(3):892–902, Jul 1999.

[47] J. Ousingsawat and M. E. Campbell. Optimal cooperative reconnaissance using multiple vehicles. *Journal Of Guidance, Control, and Dynamics*, 30(1):122–132, 2007.

[48] J. Ousingsawat and M. E. Campbell. Planning for cooperative multi-vehicle reconnaissance. *AIAA Journal of Aerospace Computing, Information, and Communication 2007*, 4(2):657–675, 2007.

[49] J. M. Passerieux and D. Van Cappel. Optimal observer maneuver for bearings-only tracking. *IEEE Trans. on Aerospace and Electronic Systems*, 34(3):777–788, Jul 1998.

[50] D. Pollard. *A User's Guide to Measure Theoretic Probability.* Cambridge UK: Cambridge University Press, 2001.

[51] S. S. Ponda. Trajectory optimization for target localization using small unmanned aerial vehicles. Master's thesis, Massachusetts Institute of Technology, 2008.

[52] F. Rafi, S. Khan, K. Shafiq, and M. Shah. Autonomous target following by unmanned aerial vehicles. In *Proc. SPIE Defence and Security Symposium*, Orlando, FL, May 2006.

[53] B. Rao, H. Durrant-Whyte, and J. Sheen. A fully decentralized multi-sensor system for tracking and surveillance. *Int. Journal of Robotics Research*, 12(1):20–44, February 1993.

[54] A. Rényi. On measures of entropy and information. In *Proc. 4th Berkeley Symposium on Mathematical Statistics and Probability*, 1961.

[55] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[56] B. Ristic and A. Gunatilaka. Information driven localisation of a radiological point source. *Information Fusion*, 9:317–326, 2008.

[57] B. Ristic, M. Morelande, and A. Gunatilaka. Information driven search for point sources of gamma radiation. *Signal Processing*, 90:1225–1239, 2010.

[58] N. Roy and R. He. Efficient POMDP forward search by predicting the posterior belief distribution. Technical Report MIT-CSAIL-TR-2009-044, Massachusetts Institute of Technology, September 2009.

[59] A. D. Ryan, H. Durrant-Whyte, and J. K. Hedrick. Information-theoretic sensor motion control for distributed estimation. In *Proc. ASME Int. Mechanical Engineering Congress and Exposition*, 2007.

[60] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[61] R. Sim and N. Roy. Global A-optimality robot exploration in SLAM. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 2005.

[62] H. Sorenson. On the development of practical nonlinear filters. *Information Sciences*, 7:253–270, 1974.

[63] H. W. Sorenson and D. L. Alspach. Recursive Bayesian estimation using Gaussian sums. *Automatica*, 7:465–479, 1971.

[64] J. L. Speyer, D. G. Hull, and C. Y. Tseng. Estimation enhancement by trajectory modulation for homing missiles. *AIAA Journal Of Guidance, Control, and Dynamics*, 7(2):167–174, 1984.

[65] S. Spry, A. R. Girard, and J. K. Hedrick. Convoy protection using multiple unmanned air vehicles: Organization and coordination. In *Proc. American Control Conference*, 2005.

[66] C. Stachniss and W. Burgard. Exploring unknown environments with mobile robots using coverage maps. In *Proc. Int. Joint Confs. on Artificial Intelligence*, 2003.

[67] A. Stentz. Optimal and efficient path planning for partially-known environments. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 1994.

[68] P. Tichavsky, C. H. Muravchik, and A. Nehorai. Posterior cramer-rao bounds for discrete-time nonlinear filtering. *IEEE Trans. on Signal Processing [see also IEEE Trans. on Acoustics, Speech, and Signal Processing]*, 46(5):1386–1396, May 1998.

[69] P. Trodden and A. Richards. Robust distributed model predictive control using tubes. In *Proc. American Control Conference*, 2006.

[70] Y. Watanabe, E. N. Johnson, and A. J. Calise. Stochastic guidance design for uav vision-based control applications. In *Proc. AIAA Guidance, Navigation and Control Conference*, August 2008.