

MIT Open Access Articles

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

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Levine, Daniel, Brandon Luders and Jonathan P. How. "Information-rich path planning with general constraints using rapidly-exploring random trees." AIAA Infotech@Aerospace 2010, 20-22 April 2010, Atlanta, Georgia.

Publisher: American Institute of Aeronautics and Astronautics

Persistent URL: <http://hdl.handle.net/1721.1/60027>

Version: Original manuscript: author's manuscript prior to formal peer review

Terms of use: Attribution-Noncommercial-Share Alike 3.0 Unported



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

Daniel Levine, Brandon Luders, and Jonathan P. How

Aerospace Controls Laboratory

Massachusetts Institute of Technology, Cambridge, MA

{dlevine,luders,jhow}@mit.edu

This paper 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 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. An extension of IRRT for multi-agent missions is also presented. IRRT is distinguished from previous solutions 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.

I. Introduction

Mobile intelligence, surveillance, and reconnaissance missions are predicated on information collection via sensor measurements. Yet trajectories generated to best utilize onboard sensing must also 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). Of particular interest are trajectories that efficiently reduce the uncertainty of target location estimates.

Prior research has used receding horizon, optimization-based techniques for selecting short-term control sequences^{6,18,20} or heuristically shaped, simple paths that are known to give optimal results in unconstrained settings.^{1,21} However, inclusion of general dynamic, environmental, or sensor constraints hinders solution generation in real-time for the optimization-based techniques and may prevent solution existence entirely for heuristic path shaping techniques. Target tracking solutions using the Partially Observable Markov Decision Process (POMDP) framework,²⁵ while having notably general constraint characterizations, are currently intractable for vehicle models with complex and/or non-holonomic dynamics.

This paper introduces the Information-rich Rapidly-exploring Random Tree (IRRT), an extension of the RRT algorithm¹⁴ that embeds information collection as predicted using Fisher Information Matrices.⁴ A further extension of IRRT for multi-agent missions is also presented. The primary contribution of this 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 of the trade-off between information-gathering and goal arrival. The flexibility to plan arbitrarily formed trajectories, under general cost functions and feasibility constraints, distinguishes the presented solution strategy for planning information-rich trajectories from the related research.

II. Problem Statement

The general problem to which this paper is addressed is as follows: A vehicle with nonlinear dynamics must traverse a bounded, non-convex environment from start state to goal state while minimizing path

distance and maximizing information gain of features in the environment. While the solution proposed is sufficiently abstracted and amenable to the general problem, we shall focus, for demonstrative purposes, on a specific problem. The selected vehicle is a Dubins car, carrying the selected 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 static targets in the environment, a bounded, non-convex 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 not addressed.

III. Previous Work

In the past decade, several solution strategies to effect information-rich path planning have been considered. Analytical solutions use the Fisher Information Matrix (FIM) to quantify trajectory information collection in an optimal control framework. Solutions seek to maximize, for example, a lower bound on the determinant of the FIM¹⁵ or the log det of the final FIM.¹⁹ While analytical solutions perform optimally for very simple problems, they are not immediately scalable to complicated scenarios (e.g., moving targets or dynamics of order higher than two).

An opposite extremum of the analytical solution strategy is one that neglects transient behavior and adopts a heuristic path shape shown to perform well in steady state. Examples of such shapes are circles, ellipses, and spirals. Barber et al. propose flight path optimization by finding an optimal altitude and radius for a circular trajectory above a stationary target.¹ Rafi et al. similarly analyze circular trajectories at a fixed altitude to track a constant velocity target, in the process optimizing the circle radius.²¹ 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 infeasibility are ill-addressed within the heuristic path shape framework.

Recent research has also considered trajectories constructed by performing receding-horizon control on-line. This class of solution strategies can be partitioned into discrete-space and continuous-space planners. The advent of the former involved 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 on-line. 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;^{2,3,29} these strategies can be summarized as greedy, next-best-view methods that perform steepest ascent on the information. Realizing the need for information-theoretic multi-step planners,⁹ Sim and Roy present a global planner for the SLAM framework which performs multi-step planning as a pruned graph search.²⁷

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.⁶ 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.¹⁸ However, the results therein presented are limited to simple constraint sets: an omnidirectional sensor performs 2D target estimation, vehicles are modeled as point masses, and risk zones are elliptical. Ponda²⁰ uses the A-optimality condition of the FIM as the objective function to optimize the trajectory of a fixed-wing aircraft with a perfectly gimbaled camera. The assumption of a gimbaled 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 dynamic constraints of the vehicle – 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. Recent research has 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.⁷ Using a prior map of the environ-

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:  $\bar{x}(t+k) \leftarrow$  final state of  $N_{near}$ 
4: while  $\bar{x}(t+k) \in \mathcal{X}_{free}$  and  $\bar{x}(t+k)$  has not reached  $x_{samp}$  do
5:   Use reference law to generate  $\bar{r}(t+k)$ 
6:   Use control law to generate  $\bar{u}(t+k)$ 
7:   Use prediction model to simulate  $\bar{x}(t+k+1)$ 
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  $x(t) \neq x_{goal}$  do
4:   Update the current state  $x(t)$ 
5:   Propagate the state  $x(t)$  by  $\Delta t \rightarrow \bar{x}(t+\Delta t)$ 
6:   while time remaining for this timestep do
7:     Expand tree by adding nodes (Algorithm 1)
8:   end while
9:   Use cost estimates to identify best feasible path,  $\{N_{root}, \dots, N_{target}\}$ 
10:  Apply best feasible path, if one exists
11:   $t \leftarrow t + \Delta t$ 
12: end while
```

ment 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 also explore forward-search in a POMDP framework to facilitate target tracking.²⁵ While the POMDP framework has shown promising results for simple vehicle models, POMDP solutions are currently intractable for vehicle models with complex dynamics.

IV. Information-rich RRT (IRRT)

The proposed Information-rich Rapidly-exploring Random Tree, or IRRT, is an extension of the closed-loop RRT^{5,12} that uses the Fisher Information Matrix framework for quantifying trajectory information content in 2D and 3D target tracking applications. This preserves the beneficial properties of closed-loop RRT, discussed below, while allowing the planner to be both cognizant of and proactive about information collection.

A. Closed-loop RRT for Accurate State Prediction

This section reviews the real-time closed-loop RRT (CL-RRT) algorithm, proposed by Frazzoli⁵ and later extended,^{11–13} which this paper builds upon. The fundamental operation in the standard RRT algorithm¹⁴ 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.¹⁴ Finally, as a sampling-based algorithm, the RRT planner performance scales with the available computational resources, avoiding the explosion in computational complexity often found in information-based planning approaches.

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.¹⁶ 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 the execution loop, are reviewed next; more detailed treatments of these algorithms have been considered in recent papers.^{12,16}

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,¹⁴ 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 IV.C) is identified (line

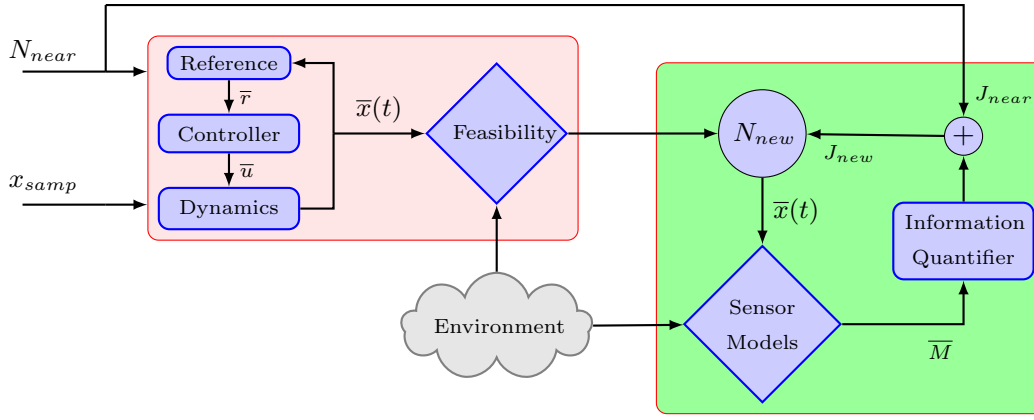


Figure 1: Block diagram of IRRT. The red shaded box denotes the typical CL-RRT functionality. The green shaded box denotes the IRRT extension to CL-RRT.

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 $\bar{u}(t)$ may be generated in the forward simulation to yield a terminal state near x_{samp} . In closed-loop RRT, a closed-loop representation of the vehicle dynamics is instead used to generate trajectories. Consider in Figure 1 the red shaded block, which represents the CL-RRT portion of IRRT. The inputs to the prediction model are x_{samp} and N_{near} , which are compared to form a simple reference trajectory $\bar{r}(t)$ (line 5), then passed through a virtual vehicle controller to form the input sequence $\bar{u}(t)$ (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 $\bar{x}(t)$ (line 7). Environmental constraints can then be queried for $\bar{x}(t)$ 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} .

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.⁵ 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.³⁰ 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 $\bar{x}(t + \Delta t)$. 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 used⁵ to select the best feasible^a path in the tree (line 9); the nature of these cost estimates is discussed further in Section IV.D. Assuming at least one such feasible path exists^b, it is selected and executed by the vehicle (line 10).

B. Path Information Quantification

From the perspective of information collection, path quality is a function of the path measurement sequence. And while CL-RRT also enjoys the benefits of smoother path planning on a stabilized vehicle model, it

^aIt is worth noting, though outside the scope of this paper, that a “lazy check” can be used to reduce the computation time spent checking the tree for feasibility.¹² 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.

^bThe 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.¹²

is the added disturbance robustness over open-loop RRT¹⁶ and the associated accurate state prediction that are particularly useful for measurement pose prediction and, therefore, for information-based planning. Consider again Figure 1; the green shaded box denotes the IRRT extension to CL-RRT. Because this state trajectory is usually simulated with high fidelity, and the result of its prediction is notably accurate, a list of predicted measurement poses $\bar{M} = \langle \bar{\mu}_1, \bar{\mu}_2, \dots, \bar{\mu}_m \rangle$ can be interpolated for each of many (possibly independent) sensors on the platform. These sensors need not have the same characteristics. Each sensor's list of predicted measurement poses is generated once per node, and thereafter has no need to be updated. Given a most recent position estimate \hat{x}_f of target f , each measurement pose $\bar{\mu}_k, k \in \{1, \dots, m\}$ can be checked against the sensor and environment models to assess visibility. The information for measurements deemed visible is quantified, as described below, and stored in the resulting node N_{new} . Visibility and information quantification of the \bar{M} elements may be recomputed with updated target estimation data.

A myriad of information-theoretic metrics exist to quantify the value of a set of measurements. Mutual information and divergence measures, the two most general information-gain metrics, are often computationally intensive. Use of either typically requires access to the posterior distribution, which is itself difficult to compute or approximate. While mutual information and divergence measures remain viable for certain sensor management problems,^{8,10,24,26} our goal is to efficiently quantify the information content of paths generated in a sample-based planner. As is often done in the analytical and receding-horizon literature,^{17,20,23,28} we refer to Fisher Information.

The Fisher Information Matrix (FIM) $J_{\mathbf{z}}(\mathbf{x})$ describes the information contained in a set of measurements \mathbf{z} about an estimation process for the vector \mathbf{x} . The inverse $J_{\mathbf{z}}(\mathbf{x})^{-1}$ of the Fisher Information Matrix is exactly the Cramér-Rao Lower Bound (CRLB), a lower bound on the achievable estimation error covariance.²² Based on this definition, we would like the CRLB to be “small,” or the FIM to be “large,” in a normed sense.³¹ A discrete system with linear state transitions and measurements, subject to additive Gaussian white sequences, has the model

$$\mathbf{x}_{k+1} = \Phi_{k+1|k} \mathbf{x}_k + \mathbf{w}_k, \quad (1)$$

$$\mathbf{z}_k = H_k \mathbf{x}_k + \mathbf{v}_k, \quad (2)$$

where $\Phi_{k+1|k}$ is the state transition matrix, H_k is the linear measurement matrix, \mathbf{w}_k is a vector of Gaussian white sequences capturing the process noise, and \mathbf{v}_k is a vector of Gaussian white sequences capturing the sensing noise. If the process and sensing noises are assumed to be zero-mean and uncorrelated, then

$$\mathbb{E} [\mathbf{w}_a \mathbf{w}_b^T] = \begin{cases} Q_k, & a = b \\ \mathbf{0}, & a \neq b \end{cases}, \quad (3)$$

$$\mathbb{E} [\mathbf{v}_a \mathbf{v}_b^T] = \begin{cases} R_k, & a = b \\ \mathbf{0}, & a \neq b \end{cases}, \quad (4)$$

$$\mathbb{E} [\mathbf{w}_a \mathbf{v}_b^T] = \mathbf{0}, \quad \forall (a, b), \quad (5)$$

where Q_k denotes the process noise covariance matrix and R_k denotes the sensor noise covariance matrix. For such systems, the recursive update equation for the FIM is

$$J_{k+1} = (Q_k + \Phi_{k+1|k} J_k^{-1} \Phi_{k+1|k}^T)^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1}. \quad (6)$$

For stationary targets ($Q_k = 0, \Phi_{k+1|k} = I, \forall k$), the recursion becomes

$$J_{k+1} = J_k + H_{k+1}^T R_{k+1}^{-1} H_{k+1}. \quad (7)$$

Therefore, the FIM in this case is additive, and the information content of a path is just the sum of the FIMs along the path edges, a considerable computational savings not found in planning methods that propagate the covariance.

The linearity assumption on the observation system can be relaxed by utilizing the linearized FIM as an approximation of the CRLB inverse. Consider 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

$$\mathbf{z}_k = \mathbf{h}(\mu_k, \mathbf{x}_f) + \mathbf{v}_k, \quad (8)$$

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.,

$$H_k(\bar{\mu}_k, \hat{\mathbf{x}}) \triangleq \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\mu_k = \bar{\mu}_k, \mathbf{x}_f = \hat{\mathbf{x}}_f(t)}. \quad (9)$$

Note that the assumption of Gaussian noise is retained, a limitation discussed further in Section VI.

The expected measurement poses $\langle \bar{\mu}_1, \dots, \bar{\mu}_m \rangle$ can be used in the framework of the FIM to quantify the information content of a particular node in the tree. Assuming a Gaussian prior, the recursion is initiated at the root node N_{root} with $J_{root}(\hat{\mathbf{x}}_f) = P_f(t)^{-1}$, where $P_f(t) = \mathbb{E}[(\mathbf{x}_f - \hat{\mathbf{x}}_f(t))(\mathbf{x}_f - \hat{\mathbf{x}}_f(t))^T]$ 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

$$J_j(\hat{\mathbf{x}}_f) = J_i(\hat{\mathbf{x}}_f) + \sum_{k=1}^m \nu(\bar{\mu}_k, \hat{\mathbf{x}}_f, \hat{\mathcal{E}}) H_k^T(\bar{\mu}_k, \hat{\mathbf{x}}_f) R_k^{-1} H_k(\bar{\mu}_k, \hat{\mathbf{x}}_f), \quad \forall f \in \mathcal{F}, \quad (10)$$

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,

$$\mathcal{I}_i(\hat{\mathbf{x}}_f) = \text{trace}(J_i^{-1}(\hat{\mathbf{x}}_f)), \quad (11)$$

which has been shown to be better suited than other FIM optimality conditions for the 3D target tracking case.²⁰ In the multi-sensor or multi-target case, convex combinations of the FIM A-optimality costs

$$\mathcal{I}_i = \sum_{f \in \mathcal{F}} w_f \mathcal{I}_i(\hat{\mathbf{x}}_f), \quad \sum_{f \in \mathcal{F}} w_f = 1 \quad (12)$$

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 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.

C. 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.⁵ In the CL-RRT algorithm (Algorithm 1, line 2), one of two heuristics are probabilistically selected, depending on whether or not a feasible path to the goal has been found. The EXPLORATION heuristic

$$i^* = \underset{i}{\text{argmin}} \hat{\tau}(x_{samp}|N_i) \quad (13)$$

uses a simple time metric $\hat{\tau}(x_{samp}|N_i)$ (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

$$i^* = \underset{i}{\text{argmin}} \hat{\tau}(x_{samp}|N_i) + \alpha_\tau \tau(N_i|N_{root}), \quad \alpha_\tau \in [0, 1) \quad (14)$$

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: $\bar{x}(t+k) \leftarrow$ final state of N_{near}
- 4: **while** $\bar{x}(t+k) \in \mathcal{X}_{free}$ **and** $\bar{x}(t+k)$ has not reached x_{samp} **do**
- 5: Use reference law to generate $\bar{r}(t+k)$
- 6: Use control law to generate $\bar{u}(t+k)$
- 7: Use prediction model to simulate $\bar{x}(t+k+1)$
- 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** \bar{M}
- 13: **Compute FIM using** (10)
- 14: Add N to \mathcal{T}
- 15: **end for**

Algorithm 4 IRRT, Execution Loop

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

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.⁵ 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 once at least one path to the goal has been found and the EXPLORATION metric otherwise.¹¹

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 information 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 discretizes the path into m' segments, each with weight λ_k and measurement pose μ_k , $k \in \{1, \dots, m'\}$. The approximate trajectory FIM $\tilde{J}_s(\hat{x}_f)$ at the sample x_{samp} for each target f as a result of taking the connecting tree path is given by

$$\tilde{J}_s(\hat{x}_f) = J_i(\hat{x}_f) + \frac{\hat{m}}{m'} \sum_{k=1}^{m'} \lambda_k \nu(\bar{\mu}_k, \hat{x}_f, \mathcal{E}) H_k^T(\bar{\mu}_k, \hat{x}_f) R_k^{-1} H_k(\bar{\mu}_k, \hat{x}_f), \quad \sum_{k=1}^{m'} \lambda_k = 1, \quad (15)$$

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

$$\tilde{\mathcal{I}}_s = \sum_{f \in \mathcal{F}} w_f \text{trace}(\tilde{J}_s^{-1}(\hat{x}_f)), \quad \sum_{f \in \mathcal{F}} w_f = 1. \quad (16)$$

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

$$i^* = \underset{i}{\text{argmin}} \frac{\tilde{\mathcal{I}}_s - \mathcal{I}_i}{\mathcal{I}_i}. \quad (17)$$

Whenever new feasible nodes N_{new} are generated for the tree, the predicted measurement poses \bar{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.

D. 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,⁵ 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 $\{N_{root}, \dots, N_{target}\}$. 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.⁵

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

$$C(N_i) = \alpha_\tau \tau(N_i|N_{root}) + \tau^*(N_i) + \alpha_{\mathcal{I}} \mathcal{I}_i, \quad (18)$$

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 maneuvering 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 Section V to be perform well empirically under a number of previously prohibitive general constraints.

E. Multi-Agent IRRT

The IRRT algorithm as previously stated can be extended in a straightforward manner for multi-agent missions. Suppose each agent q in a fully-connected network \mathcal{Q} has knowledge of the other agents’ parameters, e.g., vehicle dynamics and sensor models. Agent q grows a tree $\mathcal{T}^{[q]}$ using IRRT and periodically selects its most favorable path $\mathcal{P}_*^{[q]}$, defined as the minimal-cost feasible path in $\mathcal{T}^{[q]}$. The path cost is exactly the cost associated with the terminal node of the path (Section IV.D). The feasible set excludes regions in the vicinity of obstacles and the selected paths $\mathcal{P}_*^{[q']}$ from other agents $q' \in \{\mathcal{Q} \setminus q\}$. Agent q need only broadcast the waypoints of each node; knowing that agent’s parameters, the other agents can then reconstruct the anticipated path $\mathcal{P}_*^{[q]}$ via forward simulation.

Though each agent selects its own path in a decentralized manner, each agent utilizes full information gathered from all agents, by having each agent report its measurements to a centralized filter. Under this assumption, agent q will receive the entirety of the information collected from other agents $q' \in \{\mathcal{Q} \setminus q\}$. The implication is that at time t_0 , the information concerning target f is encapsulated by

$$J_0(\mathbf{x}_f) = P_f^{-1}(t_0) = \mathbb{E} [(\mathbf{x}_f - \hat{\mathbf{x}}_f(t_0))(\mathbf{x}_f - \hat{\mathbf{x}}_f(t_0))^T]^{-1}, \quad (19)$$

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

$$J_{term}(\mathbf{x}_f) = J_0(\mathbf{x}_f) + \sum_{q \in \mathcal{Q}} \Delta J(\mathbf{x}_f; \mathcal{P}_*^{[q]}), \quad (20)$$

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

$$J_{root}^{[q]}(\hat{\mathbf{x}}_f) = J_0(\hat{\mathbf{x}}_f) + \sum_{q' \in \{Q \setminus q\}} \Delta J(\hat{\mathbf{x}}_f; \mathcal{P}_*^{[q']}). \quad (21)$$

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 natural cooperative behaviors.

This work assumes the existence of a central filter that processes all measurements from the networked sensors. A distributed planning architecture, in which measurements are filtered locally by individual agents and estimation results are shared between agents in a consensus framework, incorporates more realistic and less restrictive assumptions. Moreover, mitigation of information-loss due to an individual agent replanning its path is not yet considered in this work. Such mitigation likely would require time-based information discounting which takes tree diversity into account, and will be considered in future work.

V. Results

The following simulation results demonstrate 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. 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 II) are considered, including non-holonomic vehicles, limited-field-of-view sensing, cluttered obstacle environments, multiple targets, three dimensions, and finally multiple vehicles. Even subject to these constraints, which render many existing approaches in the literature intractable, the IRRT generates paths with emergent information-gathering characteristics. Some basic analysis is also provided to illuminate the trade-off between information-gathering and goal arrival that is taking place.

All simulations were performed using a real-time implementation of the IRRT algorithm in Java, run 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. In these simulations, all sensor measurements are assumed to be bearings-only; a polar coordinate Extended Kalman Filter (EKF) is used to estimate the target position, with measurements processed at a rate of 15 Hz.

A. Quadrotor Scenario

Consider a quadrotor unmanned aerial vehicle (UAV) agent 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 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 (18).

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 3). For this scenario, we have selected $\alpha_\tau = 0.5$ and $\alpha_{\mathcal{I}} = 6000$ s/m². 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 ($P(t_0) = \text{diag}(8.0, 8.0, 8.0)$ m²), but as bearing measurements are taken, the longest axis of the uncertainty ellipsoid 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.

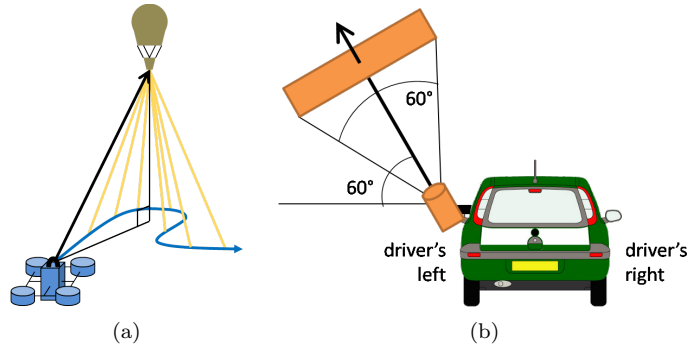


Figure 2: (a) Diagram of the quadrotor UAV scenario. As the UAV navigates at a fixed altitude below that of the aerial target, the onboard sensor has an unobstructed view of the aerial target. (b) Dubins car diagram with a pointed monocular vision sensor.

An example trial of the scenario is depicted in Figure 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 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 3(b)-3(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 (18) becomes sufficiently small that the total cost is reduced by returning to the goal; at this point, the planner begins selecting successively shorter paths to the goal (Figures 3(d)-3(e)). After 20 seconds have elapsed, the vehicle has collected sufficient information on the target and arrives at the goal (Figure 3(f)). From a simple cost tradeoff in (18) between path length and uncertainty, complex target tracking behavior for the autonomous UAV has emerged naturally.

B. 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 estimating 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 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 $\alpha_{\mathcal{I}} = 8000$ s/m². The car is assumed to move at a fixed velocity of 0.4 m/s; a variation of the pure pursuit reference law¹³ 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 scenario is given in Figure 4. The agent quickly plans a winding path that both anticipates measurements about the estimated target position and reaches the goal (Figure 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(b)-4(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(e)) in order to take additional measurements before finally turning toward the goal (Figure 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.

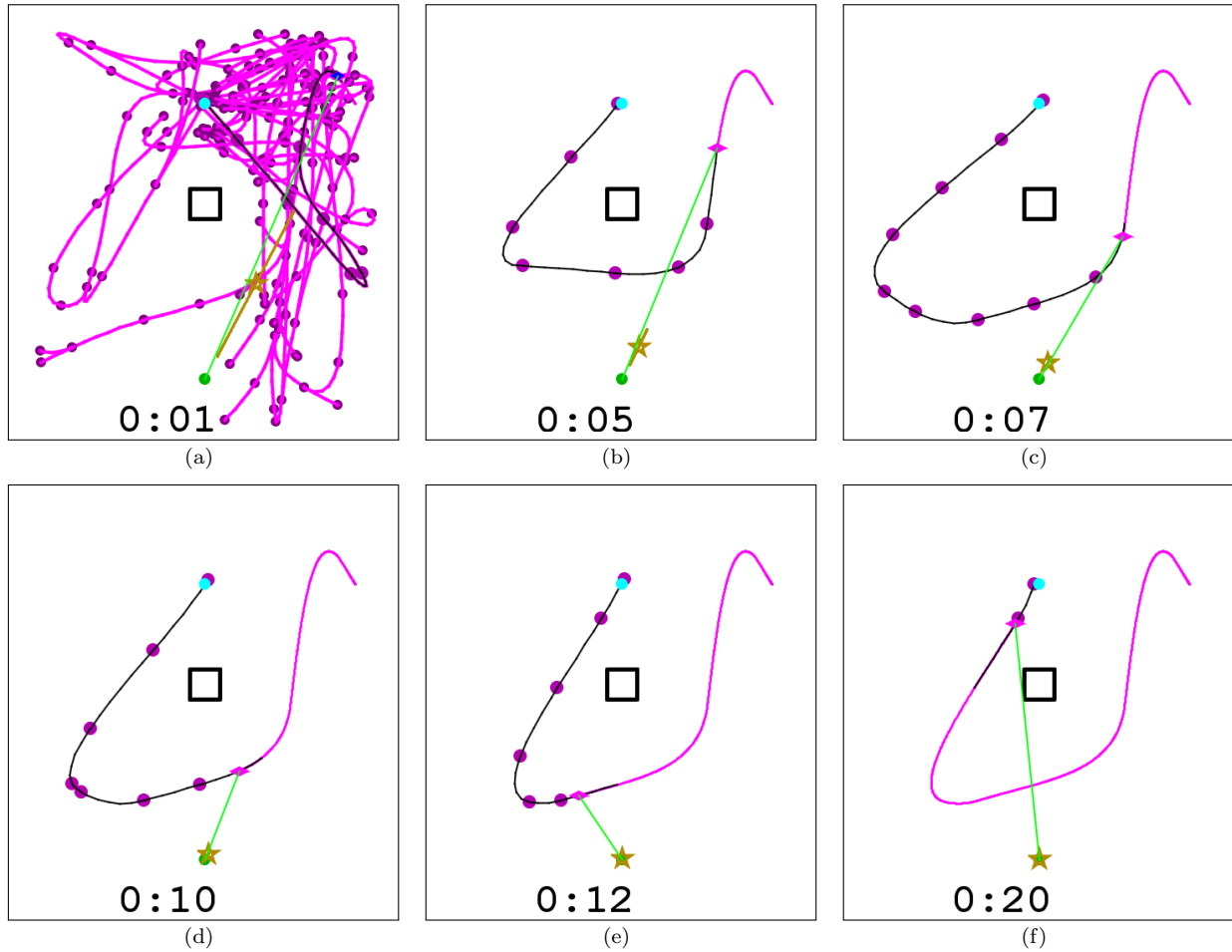


Figure 3: Snapshots of a typical trajectory for a simulated quadrotor navigating toward a goal while tracking an aerial target. The vehicle (magenta diamond, highlighted in blue in (a)) starts at top-right, and attempts to gather information about the target (green, bottom-center) and reach the goal 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.

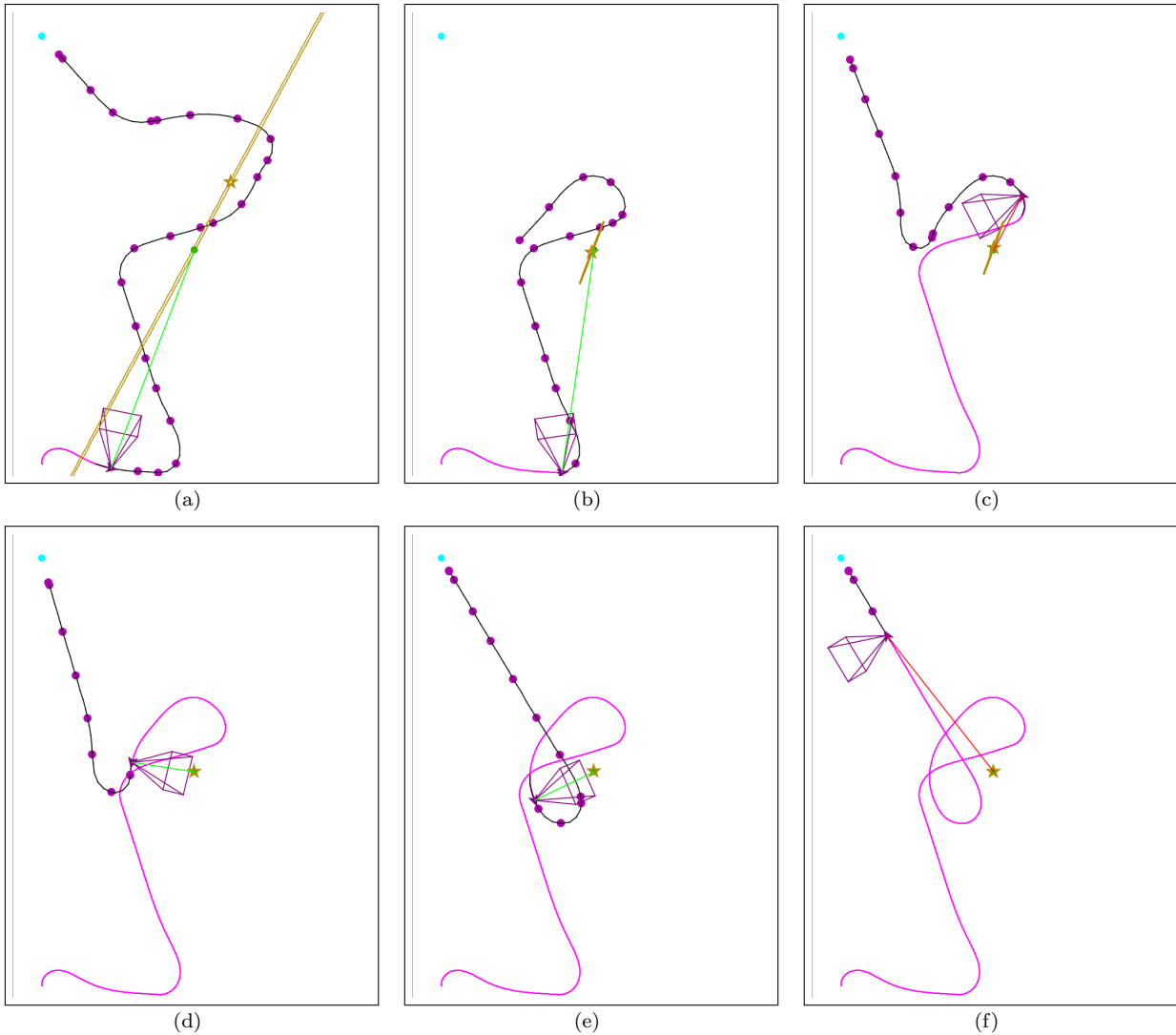


Figure 4: Snapshots of a simple Dubins car simulation with side mounted camera navigating toward a goal while tracking one aerial target. See Figure 3 for a full legend; the field of view is denoted by the magenta pyramid. The RRT tree has been suppressed for clarity.

C. Complex Dubins Scenario

Consider now the full problem statement as outlined in Section II 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 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 initially plans to move toward the goal and then loiter in its vicinity, occasionally making distant measurements of the targets (Figure 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 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 5(c)-5(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 5(f)).

D. 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 to reach the goal. In this section, we revisit the complex Dubins scenario considered in Section V.C, with particular focus on reduction in target uncertainty over time.

Figure 6 plots the value of the information A-optimality cost, (11), for the complex Dubins scenario trial shown in Figure 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 $b = \{-1, 0, 1, \dots, 5\}$. Note that as $b \rightarrow -\infty$, $\alpha_{\mathcal{I}} \rightarrow 0$, 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 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.

E. 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 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 located 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 8. The agent begins with a path directly to the goal (Figure 8(a)), but the planner then identifies a path which gives the agent sufficient time to rotate and peer into the doorway (Figure 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 8(c)) to get a much closer view of the target behind them (Figure 8(d)). The planner then seeks to return the agent to the goal, and after some wandering succeeds in doing so (Figures 8(e)-8(f)).

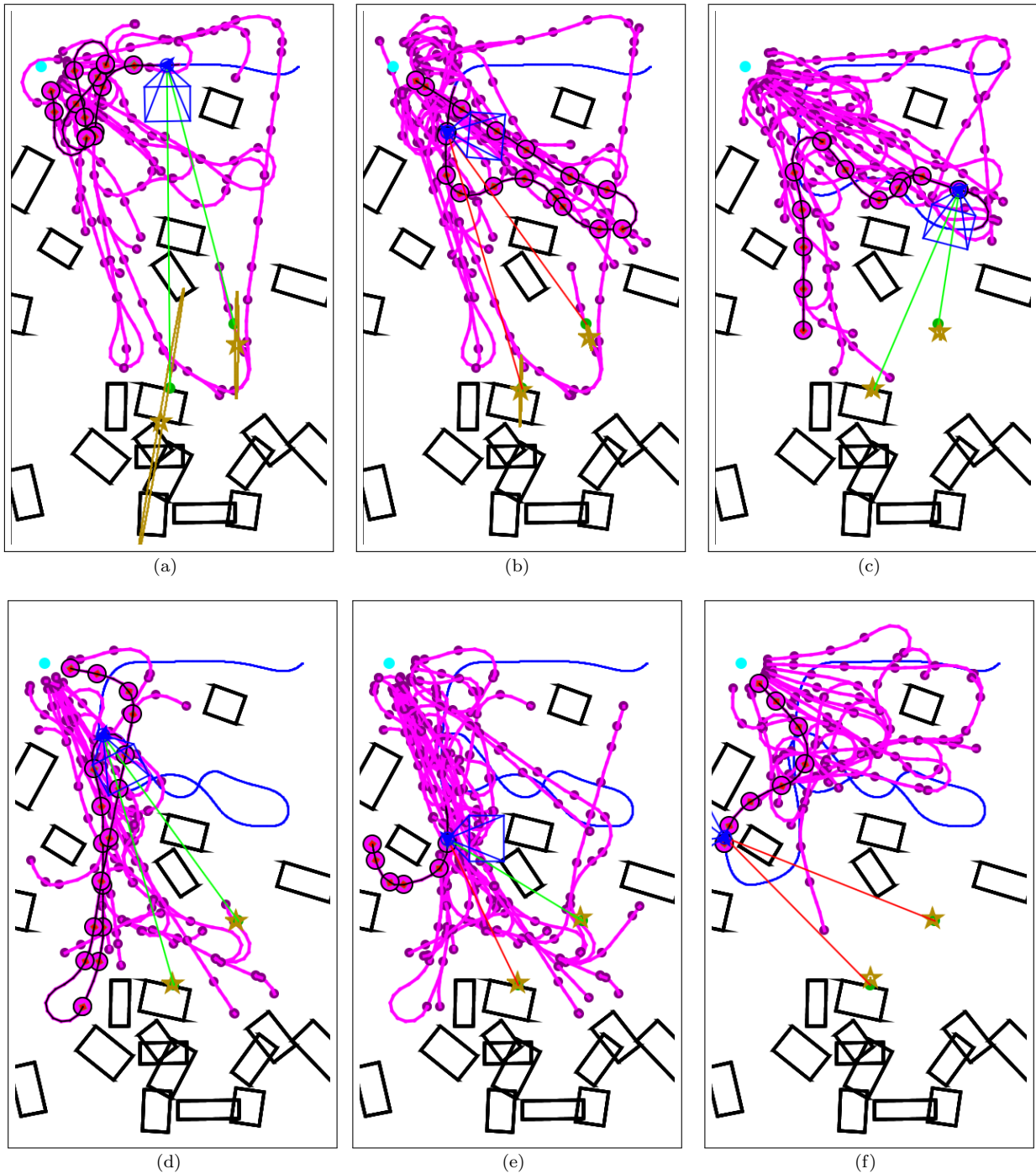


Figure 5: Snapshots of a complex simulation of a Dubins car with side mounted camera navigating toward a goal while tracking two aerial targets. See Figure 3 for a full legend; the field of view is denoted by a magenta pyramid. The RRT tree is left visible in all figures to demonstrate how the tree evolves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

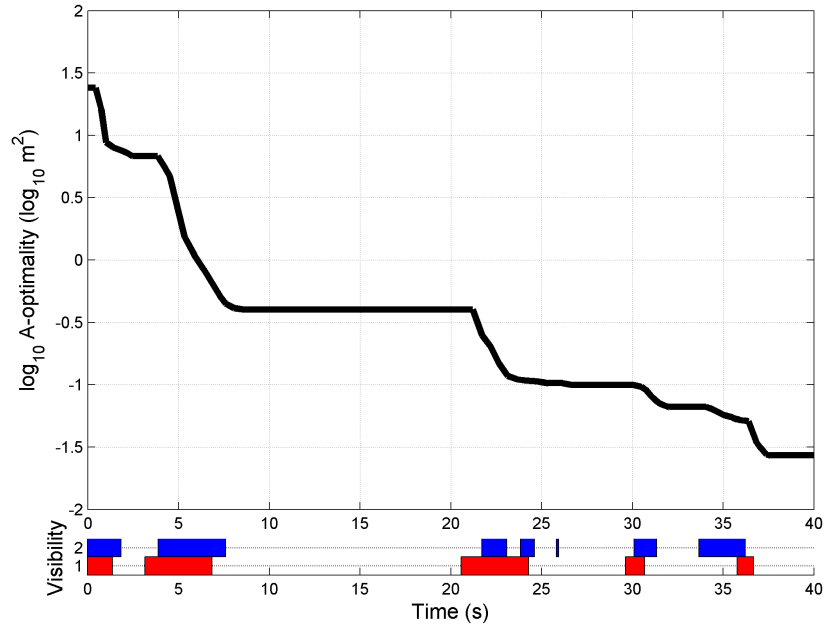


Figure 6: Comparison of the information A-optimality cost versus time for the complex Dubins scenario as shown in Figure 5. The colored bars at the bottom of the figure correspond to the time intervals during which each target is visible for measurement by the agent.

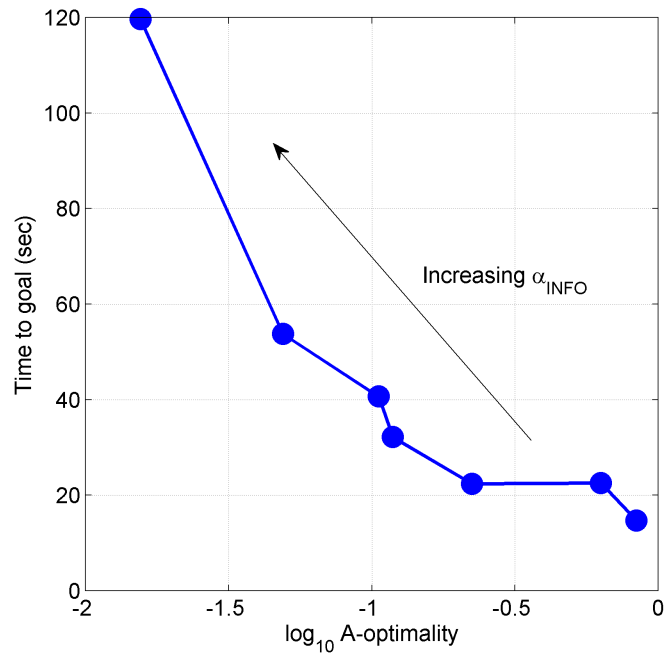


Figure 7: Comparison of average mission duration versus average terminal information A-optimality cost. Data points are parameterized by the relative information cost weighting term from (18) with values $\alpha_{\mathcal{I}} = 10^b, b = \{-1, 0, \dots, 5\}$. Each data point corresponds to one value of b , with $b = -1$ at bottom-right and $b = 5$ at top-left.

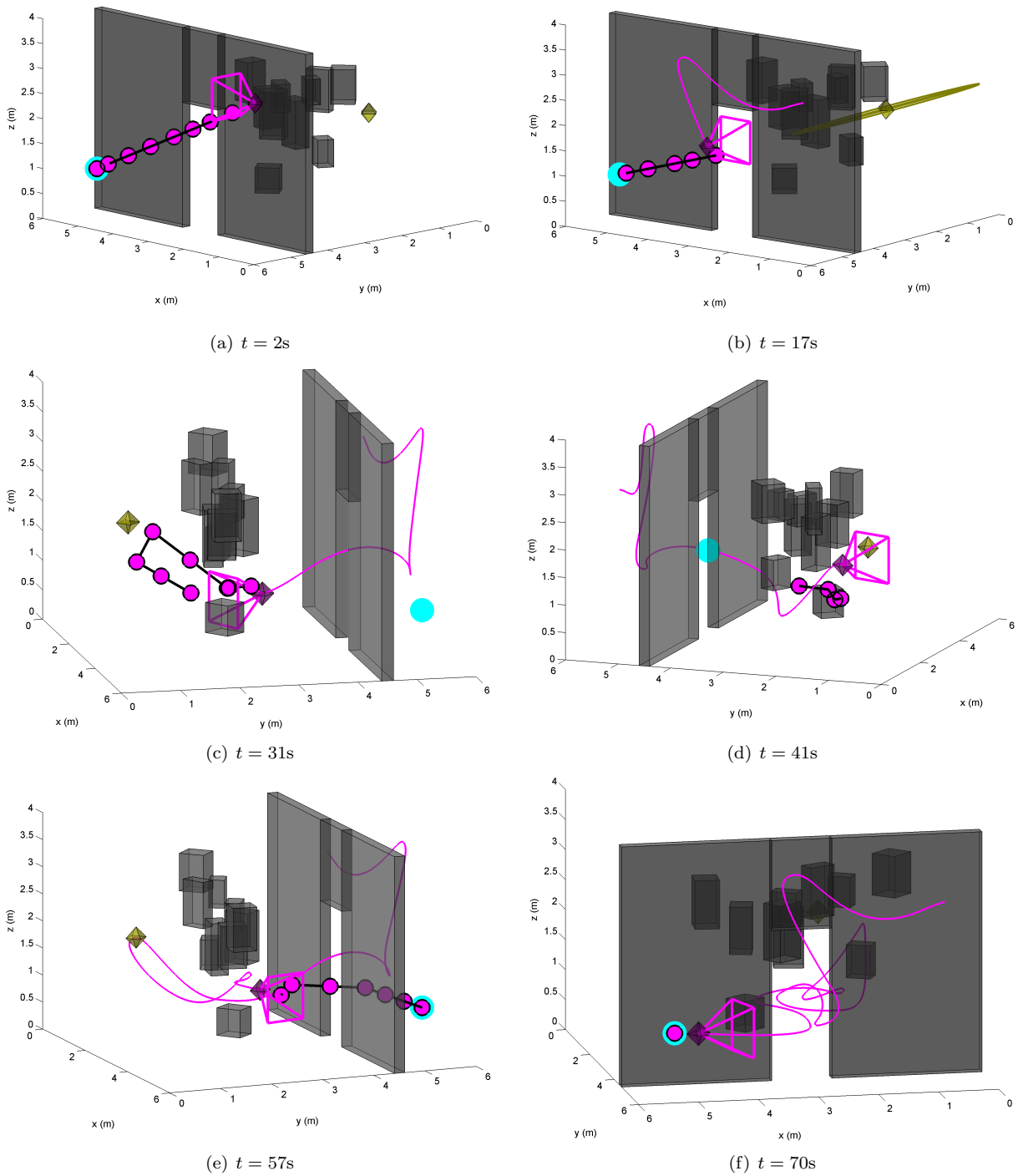


Figure 8: Snapshots of a typical trajectory for a simulated yawed quadrotor navigating toward a goal while tracking an aerial target in three dimensions. The vehicle (magenta diamond) attempts to gather information about the target (estimate and uncertainty in gold) and reach the goal waypoint (cyan) 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.

F. Multi-Agent Scenario

Finally, we consider a team of two Dubins agents, collectively tasked with taking measurements of an aerial target. Each agent plans its own paths while using centralized information estimates using IRRT, as described in Section IV.E. 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 $\alpha_{\mathcal{I}}^{[1]} = \alpha_{\mathcal{I}}^{[2]} = 1900 \text{ s/m}^2$.

An example trial for a such a scenario is depicted in Figure 9. Initially, in Figure 9, 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 9(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 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 9(e)), the remaining path segments connecting each agent to the goal are followed (Figure 9(f)).

VI. Conclusions

We have presented the IRRT algorithm for efficient information-rich trajectory planning. The primary contribution of this algorithm is target-based information maximization in arbitrary, possibly heavily constrained environments, with vehicle dynamic constraints and sensor limitations. In contrast with the class of optimal trajectory solvers, IRRT is a sample-based planning algorithm that scales with the available computational resources. A progression of simulation results has demonstrated that this algorithm can generate complex target-tracking behaviors from a simple model of the trade-off between information-gathering and goal arrival.

Several extensions to this work have been proposed. In order to plan information-rich paths in the presence of non-Gaussian additive white noise processes, which may yield multimodal posterior distributions, nonlinear sample-based estimation methods can be employed to approximate the posterior. In multi-agent scenarios, mitigation of information-loss due to replanning may be achieved via time-based discounting parametrized by, for example, path diversity. Future work will also include more extensive analysis of the IRRT’s properties and performance, including complexity and comparison to many existing heuristic planners. Finally, while this work assumes that decentralized path planning modules access a central measurement processing module, a general consensus-based framework that anticipates fusion of agent estimation results is more realistic and less restrictive.

Acknowledgements

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

References

- ¹D. B. Barber, J. Redding, T. Mclain, R. Beard, and C. Taylor. Vision-based target geo-location using a fixed-wing miniature air vehicle. *J. Intell. Robotics Syst.*, 47(4):361–382, 2006.
- ²Frédéric Bourgault, Alexei A. Makarenko, Stefan B. Williams, Ben Grocholsky, and Hugh F. Durrant-Whyte. Information based adaptive robotic exploration. In *IEEE International Conference on Intelligent RObots and Systems*, 2002.

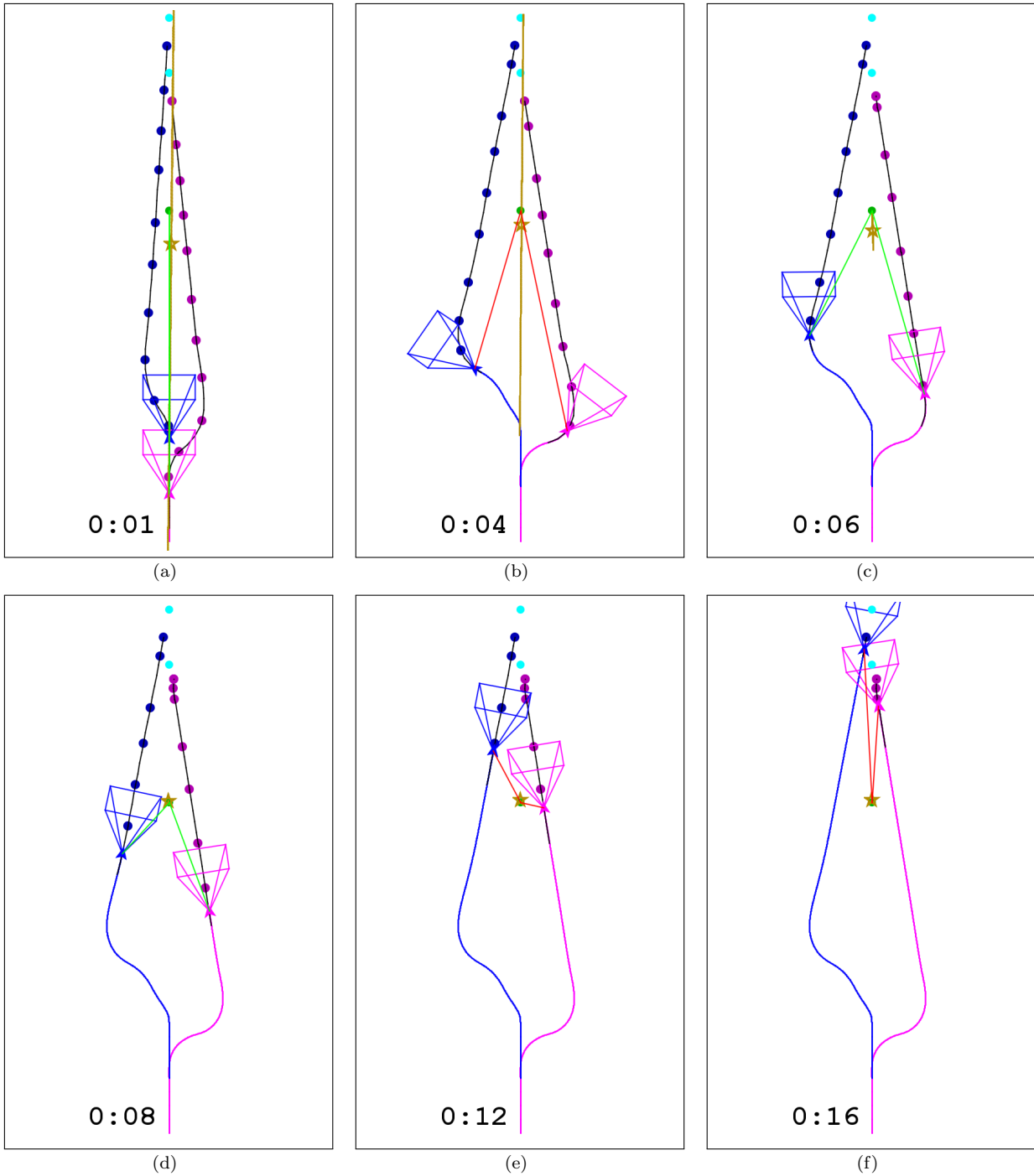


Figure 9: A multi-agent Dubins scenario with sensor constraints. Each agent plans paths that minimize the goal arrival time and maximize the shared information. The ability for both agents to simultaneously take measurements from disparate measurement poses, as in (c) and (d), is considered favorable by the cost function of each agent's path planning module.

- ³Hans Jacob S. Feder, John J. Leonard, and Christopher M. Smith. Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research*, 18:650–668, 1999.
- ⁴Ronald 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.
- ⁵E. 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.
- ⁶Eric W. Frew. *Trajectory Design for Target Motion Estimation Using Monocular Vision*. PhD thesis, Stanford University, 2003.
- ⁷Ruijie He, Sam Prentice, and Nicholas Roy. Planning in information space for a quadrotor helicopter in a gps-denied environment. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008.
- ⁸Gabriel M. Hoffman and Claire J. Tomlin. Mobile sensor network control using mutual information methods and particle filters. *Automatic Control, IEEE Transactions on*, 55:32–47, 2010.
- ⁹Shoudong Huang, N. M. Kwok, G. Dissanayake, Q.P. Ha, and Gu Fang. Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.
- ¹⁰Chris Kreucher, Keith Kastella, and Alfred O. Hero III. Sensor management using an active sensing approach. *Signal Processing*, 85:607–624, 2005.
- ¹¹Y. Kuwata, G. A. Fiore, J. Teo, E. Frazzoli, and J. P. How. Motion planning for urban driving using RRT. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 1681–1686, Nice, France, September 2008.
- ¹²Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 17(5):1105–1118, September 2009.
- ¹³Y. Kuwata, J. Teo, S. Karaman, G. Fiore, E. Frazzoli, and J. P. How. Motion planning in complex environments using closed-loop prediction. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2008.
- ¹⁴S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical Report 98-11, Iowa State University, October 1998.
- ¹⁵P. T. Liu. An optimum approach in target tracking with bearing measurements. *Journal of Optimization Theory and Applications*, 56(2):205–214, February 1988.
- ¹⁶B. D. Luders, S. Karaman, E. Frazzoli, and J. P. How. Bounds on tracking error using closed-loop rapidly-exploring random trees. In *Proceedings of the IEEE American Control Conference (to appear)*, Baltimore, MD, USA, June-July 2010.
- ¹⁷Y. Oshman and P. Davidson. Optimization of observer trajectories for bearings-only target localization. *Aerospace and Electronic Systems, IEEE Transactions on*, 35(3):892–902, Jul 1999.
- ¹⁸Jarurat Ousingsawat and Mark E. Campbell. Optimal cooperative reconnaissance using multiple vehicles. *Journal Of Guidance, Control, and Dynamics*, 30(1), 2007.
- ¹⁹J. M. Passerieux and D. Van Cappel. Optimal observer maneuver for bearings-only tracking. *Aerospace and Electronic Systems, IEEE Transactions on*, 34(3):777–788, Jul 1998.
- ²⁰Sameera S. Ponda. Trajectory optimization for target localization using small unmanned aerial vehicles. Master’s thesis, Massachusetts Institute of Technology, 2008.
- ²¹Fahd Rafi, Saad Khan, Khurram Shafiq, and Mubarak Shah. Autonomous target following by unmanned aerial vehicles. In *Proceedings of SPIE*, Orlando, FL, May 2006.
- ²²Calyampudi Radakrishna Rao. Information and the accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37:81–89, 1945.
- ²³Branko Ristic and Ajith Gunatilaka. Information driven localisation of a radiological point source. *Information Fusion*, 9:317–326, 2008.
- ²⁴Branko Ristic, Mark Morelande, and Ajith Gunatilaka. Information driven search for point sources of gamma radiation. *Signal Processing*, 90:1225–1239, 2010.
- ²⁵Nicholas Roy and Ruijie He. Efficient POMDP forward search by predicting the posterior belief distribution. Technical Report MIT-CSAIL-TR-2009-044, Massachusetts Institute of Technology, September 2009.
- ²⁶Allison D. Ryan, Hugh Durrant-Whyte, and J. Karl Hedrick. Information-theoretic sensor motion control for distributed estimation. In *Proceedings of the ASME International Mechanical Engineering Congress and Exposition*, 2007.
- ²⁷Robert Sim and Nicholas Roy. Global a-optimality robot exploration in SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.
- ²⁸J. L. Speyer, D. G. Hull, and C. Y. Tseng. Estimation enhancement by trajectory modulation for homing missiles. *Journal Of Guidance, Control, and Dynamics*, 7(2):167–174, 1984.
- ²⁹Cyrill Stachniss and Wolfram Burgard. Exploring unknown environments with mobile robots using coverage maps. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, 2003.
- ³⁰A. Stentz. Optimal and efficient path planning for partially-known environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1994.
- ³¹D. Uciński. *Optimal Measurement Methods for Distributed Parameter System Identification*. CRC Press, 2005.