Information-rich Task Allocation and Motion Planning for Heterogeneous Sensor Platforms

Brandon Luders,* Daniel Levine,† Sameera Ponda,‡ and Jonathan P. How§

Aerospace Controls Laboratory

Massachusetts Institute of Technology, Cambridge, MA

{luders,dlevine,sponda,jhow}@mit.edu

This paper introduces a novel stratified planning algorithm for teams of heterogeneous mobile sensors that maximizes information collection while minimizing resource costs. The main contribution of this work is the scalable unification of effective algorithms for decentralized informative motion planning and decentralized high-level task allocation. We present the Information-rich Rapidly-exploring Random Tree (IRRT) algorithm, which is amenable to very general and realistic mobile sensor constraint characterizations, as well as review the Consensus-Based Bundle Algorithm (CBBA), offering several enhancements to the existing algorithms to embed information collection at each phase of the planning process. The proposed framework is validated with simulation results for networks of mobile sensors performing multi-target localization missions.

I. Introduction

Mobile sensing agents often serve a crucial role in seeking out and gathering information for intelligence, surveillance, and reconnaissance (ISR) missions, both in military and civilian applications. Tasks performed by such agents might include target classification, target localization, mapping, and search and track, among others. Central to many of these tasks is the notion of collecting information to reduce uncertainty. By deploying a team of cooperative, autonomous agents, complex information collection tasks can be completed effectively and efficiently. However, there are many problems posed by planning for such teams of agents, notably, the absence of a centralized trajectory planner, limits on communication throughput/connectivity, and scalability of both with respect to the number of agents and tasks. Any planning framework for cooperative autonomous systems must be able to effectively address these concerns.

Provided that mobile sensing incurs resource costs to the operator, when evaluating agent plans, one typically seeks to both maximize the information content and minimize the cost. This motivation is central to the informative motion planning problem,[1–5] in which plans consist of trajectories for dynamically constrained sensing agents. To accommodate a variety of sensor platforms, the selected informative motion planning algorithm should be amenable to nonholonomic and/or differential dynamic constraints. Obstacles further present a challenge, in that they can both constrain the vehicle motion and occlude observations. Finally, the limitations inherent in a sensing mechanism (e.g., narrow field of view) should be accounted for when anticipating informativeness. In this paper, we present the Information-rich Rapidly-exploring Random Tree (IRRT) as an online solution method that affords such general constraint characterizations for the informative motion planning problem.

Given a fixed team of mobile sensors, solutions to the informative motion planning problem across the team may be generated in a decentralized manner. However, given a larger network of mobile sensors to select team subsets from, it is possible that another team may be better suited to extract the requisite information with lower resource expenditure. For example, a team of heterogeneous sensor agents could be

---

*Research Assistant, Department of Aeronautics and Astronautics, MIT; Member AIAA
†Research Assistant, Department of Aeronautics and Astronautics, MIT
‡Research Assistant, Department of Aeronautics and Astronautics, MIT
§Richard C. Maclaurin Professor of Aeronautics and Astronautics, MIT; Associate Fellow AIAA

American Institute of Aeronautics and Astronautics

formed online to address a particular uncertainty reduction task that would require their respective sensing actions in concert. Moreover, solving the informative motion planning problem over all agents in a network exhibits poor scalability. Towards this end, this paper examines the role of task allocation in reducing the computational requirements and increasing the efficiency of informative motion planners.

Autonomous task allocation is a significantly complex combinatorial problem (NP-hard),[6] which usually involves simultaneously allocating a set of tasks among a team of heterogeneous agents so as to optimize overall mission efficiency and reduce costs incurred. The tasks can have different locations and time-windows of validity and may require coordinated execution between several agents.[7,8] Furthermore, the planning architecture must account for vehicle limitations, agent-task compatibility requirements, and network configuration and communication requirements. For large teams of agents, centralized approaches quickly become computationally intractable and distributed architectures must be adopted.[9] In addition to alleviating the computational burden by parallelizing the planning process, distributed controllers often act on local information, making response times to changes in situational awareness significantly faster than those achievable under a purely centralized planning architecture. Distributed algorithms are typically augmented with consensus algorithms[10–12] so agents can converge on a consistent state before performing the task allocation. Although consensus algorithms guarantee convergence on information, they may take a significant amount of time and often require transmitting large amounts of data.[13] One particular algorithm of interest is the Consensus-Based Bundle Algorithm (CBBA),[9] a distributed auction protocol that provides provably good approximate solutions for multi-agent multi-task allocation problems over networks of agents, and guarantees a conflict-free solution despite possible inconsistencies in situational awareness. CBBA consists of iterations between two phases: a bundle building phase where each vehicle greedily generates an ordered bundle of tasks, and a consensus phase where conflicting assignments are identified and resolved through local communication between neighboring agents. The bidding process of CBBA runs in polynomial time, demonstrating good scalability with increasing numbers of agents and tasks, making it well suited to real-time dynamic environments.

In this paper, we analyze solution methods for fusing informative motion planning and informative task allocation. In particular we examine the performance of IRRT, with and without the stratified planning provided by CBBA. In particular, we show that:

- Stratification dramatically reduces computational requirements;

- Stratification uses deconfliction to improve performance of the multiagent informative motion planner;

- And the informative task allocation heuristic captures reduced-order capabilities of sensing agents and effectively operates as a relative capability function.

In the sequel, the multiagent information collection problem we consider is stated. Section III describes the system architecture by which we achieve stratification. Section IV outlines the informative motion planning problem, our solution algorithm (IRRT), and its properties. Section V describes the task allocation problem, a provably near-optimal solution strategy (CBBA), and its extension to the informative task allocation problem. Results comparing our informative motion planning algorithm with and without stratification are presented in Section VI. Finally, conclusions and thoughts on future work are presented in Section VII.

II.   Problem Statement

Consider a bounded, open set $X \subset \mathbb{R}^{d_x}$ of dimension d_x partitioned into an obstacle region $X_{\mathrm{obs}} \subset X$ and an obstacle-free region $X_{\mathrm{free}} = X \setminus X_{\mathrm{obs}}$. The obstacle-free region is traversable by a connected network $\mathcal{Q}$ of heterogeneous mobile sensing agents, each of which has a fully observable state $\mathbf{x}^{[q]}$, carries a set of sensors $S^{[q]}$, and has a known model of resource expenditure rates for actions it can execute. The sensing platforms are subjected to various dynamic and environmental constraints, and the individual sensors $s \in S^{[q]}$ are characterized by a set of limitations. Moreover, the obstacles comprising X_{obs} are known exactly and both constrain vehicle motion and may occlude observability of measurements.

Suppose also the existence of a known number of independent features, each of which has a partially observable state that is a realization $\mathbf{y}_i \in Y$ of a random vector[a] $\mathbf{y}_i \in Y$, where $Y \subseteq \mathbb{R}^{d_y}$ is of dimension d_y.

---

[a]Our methods, however, extend to the non-Bayesian formalism where the parameter $\mathbf{y}_i \in Y$ is an unknown, deterministic vector.

Given a prior belief $p_{\mathbf{y}_i}(\mathbf{y}_i)$ and likelihood distribution $p_{\mathbf{z}|\mathbf{y}_i}(\mathcal{Z}|\mathbf{y}_i)$ for a set of observations $\mathcal{Z}$, the posterior belief $p_{\mathbf{y}_i|\mathbf{z}}(\mathbf{y}_i|\mathcal{Z})$ may be computed using Bayes rule.

We assume that during the online estimation of $\mathbf{y}_i$, all beliefs $b_{\mathbf{y}_i}$ can be approximated by a known K_i-component Gaussian mixture model, i.e.,

$$b_{\mathbf{y}_i} \approx \sum_{k=1}^{K_i} w_{i,k}\, \mathcal{N}(\mu_{i,k}, \Lambda_{i,k}), \qquad \sum_{k=1}^{K_i} w_{i,k} = 1, \quad w_{i,k} > 0, \tag{1}$$

where $\mu_{i,k}$ and $\Lambda_{i,k}$ are the mean vector and covariance matrix, respectively, of the k-th mixture component when estimating $\mathbf{y}_i$. For such Gaussian mixture models, the mixture mean μ_i and covariance Λ_i are, respectively,

$$\mu_i = \sum_{k=1}^{K_i} w_{i,k}\, \mu_{i,k}, \tag{2}$$

$$\Lambda_i = \sum_{k=1}^{K_i} w_{i,k} \left[\Lambda_{i,k} + (\mu_i - \mu_{i,k})(\mu_i - \mu_{i,k})^T \right]. \tag{3}$$

In the remainder of this work, we specialize to the stationary target localization problem,[b] in which a partially observable state $\mathbf{y}_i \in Y \subseteq X$ represents the location of target i in physical space. Target i is said to be localized when the uncertainty in $b_{\mathbf{y}_i}$ is significantly small; here, we use the specific criterion that the A-optimality $\texttt{trace}(\Lambda_i)$ on its mixture covariance Λ_i is reduced below a mission-specified threshold $\epsilon_i > 0$ (that may vary by target).

The overall objective is that the agents in $\mathcal{Q}$ safely navigate X_{free} to collectively localize all targets $\mathbf{y}_i$ with minimal resource expenditure.

III. System Architecture

This section outlines the proposed algorithmic architecture for managing a team of unmanned, autonomous vehicles peforming a generalized series of tasks. After introducing our generalized approach to such problems, the specific architecture implemented to achieve information-rich planning for target uncertainty reduction is also presented. A key feature of our approach is the decentralization of all aspects of the planning problem, including assignment of task bundles, trajectory planning, and target estimation. The individual algorithmic components of this architecture are presented in the subsequent sections.

This paper uses a traditional, hierarchical decomposition of the full UAV mission planning problem into several sub-tasks. Such a decomposition decouples the overall UAV mission planning problem, which is generally intractable as a whole, into several fundamental planning challenges that can be addressed independently. It has assumed that some set of tasks is given for the set of agents to perform. The problem is then decomposed into (i) task assignment, assigning sequences of tasks to agents or teams of agents; (ii) trajectory design, identifying feasible trajectories for each agent to complete its tasks; and (iii) the low-level control of each vehicle to follow the given trajectory.

Figure 1 presents an overall framework for the types of mission planning problems considered in this work. Note that in this generalized architecture, there is no centralized "sensor fusion" block which maintains estimates relevant to the tasks, as new observations are made by all agents. Rather, each agent is responsible for maintaining estimates, using its own sensor package, for the tasks it has been assigned. This collectively establishes the full set of task estimation data. This information is relayed, along with agent state data, to the Task Allocation block. The Task Allocation block uses this information to assign tasks to individual agents or teams of agents[c], in order to optimize some objective. Each agent then has its own separate Path Planning algorithm, which designs trajectories in order to perform those tasks while meeting any constraints and/or safety requirements.

Figure 2 shows the proposed IRRT+CBBA architecture, designed for a team of agents to cooperatively complete a sequence of uncertainty reduction tasks in a decentralized fashion. Each agent maintains its

---

[b]The solution strategy proposed by this paper, however, is extensible to other, more general uncertainty reduction problems.
[c]Most of the concepts in this paper are easily extended to teams of agents, rather than individual agents. For clarity of presentation, this is suppressed for the remainder of this work.
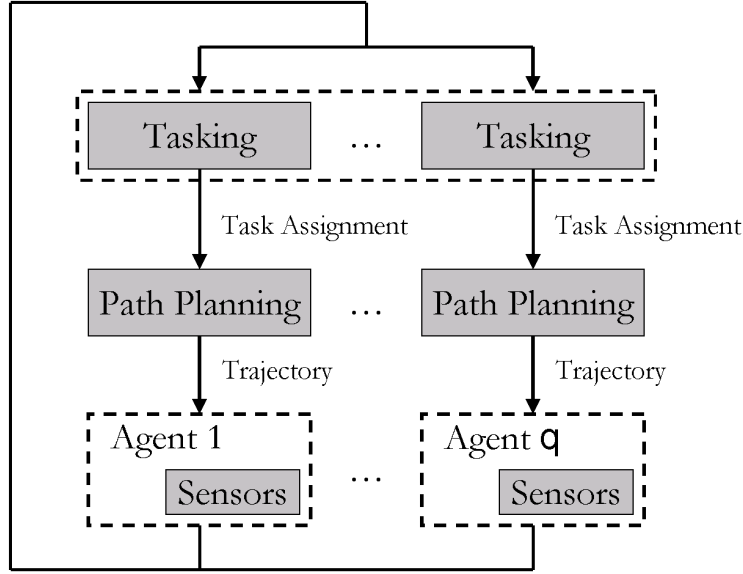
Figure 1. General system block diagram for proposed mission planning framework.

own trajectory planner, using the IRRT algorithm, as well as its own dynamics. The IRRT planner designs trajectories which favor information collection (Section IV); thus, the IRRT planner requires knowledge of each target it has been assigned to localize, and their up-to-date estimates. The vehicle maintains its own internal estimates of the targets it has been assigned, based on (in hardware) actual sensor measurements or (in software) predicted measurements based on vehicle state; these are then relayed to the other vehicles and the task allocation process. For collision avoidance, the IRRT planner must also be aware of its own vehicle state, and the state and plans of other vehicles.

The task allocation process uses the true locations of each agent and the estimated locations of each target in order to assign agents to targets (Section V-A). Task allocation is performed in this architecture using CBBA, a decentralized, consensus-based assignment algorithm in which each agent "bids" on a bundle of tasks according to their perceived effectiveness in completing them (Section V-B). Once the target assignments have been completed, they are relayed to each agent, along with the most recent PDFs for those targets (since those vehicles may not have been previously estimating those targets).

To better integrate the components of the planning problem, particularly predicted information, several extensions are presented in this paper for the task allocation framework (Section V-C). Rather than requiring allocation of every individual targets, nearby targets are clustered together, via k-means clustering. Furthermore, the CBBA algorithm should incorporate some knowledge of the information-gathering capabilities of each agent; this is implemented via the task information heuristics. These extensions are discussed in more detail in the subsequent sections.

IV.   Informative Motion Planning

A.   Subproblem Statement

The informative motion planning problem is an amalgamation of the motion planning and adaptive sampling problems. We will first briefly review the motion planning problem statement and then amend it to capture informativeness of plans.

Consider a bounded, open set $X \subset \mathbb{R}^{d_x}$ of dimension d_x partitioned into an obstacle region $X_{\mathrm{obs}} \subset X$ and an obstacle-free region $X_{\mathrm{free}} = X \setminus X_{\mathrm{obs}}$. Points in X_{obs} are said to be in collision, while those in X_{free} are said to be collision-free. Given an initial collision-free state $x_{\mathrm{init}} \in X_{\mathrm{free}}$ and a goal region $X_{\mathrm{goal}} \subset X_{\mathrm{free}}$, the feasible motion planning problem is to find a path $\sigma : [0, T] \mapsto X_{\mathrm{free}}$, i.e., that is collision-free at all points and that satisfies the specified initial state $\sigma(0) = x_{\mathrm{init}}$ and terminal state $\sigma(T) \in X_{\mathrm{goal}}$ constraints, for some $T \in \mathbb{R}^+$. When such feasible paths exists, often one is concerned with finding the minimum-cost

American Institute of Aeronautics and Astronautics

(a) Overall Architecture

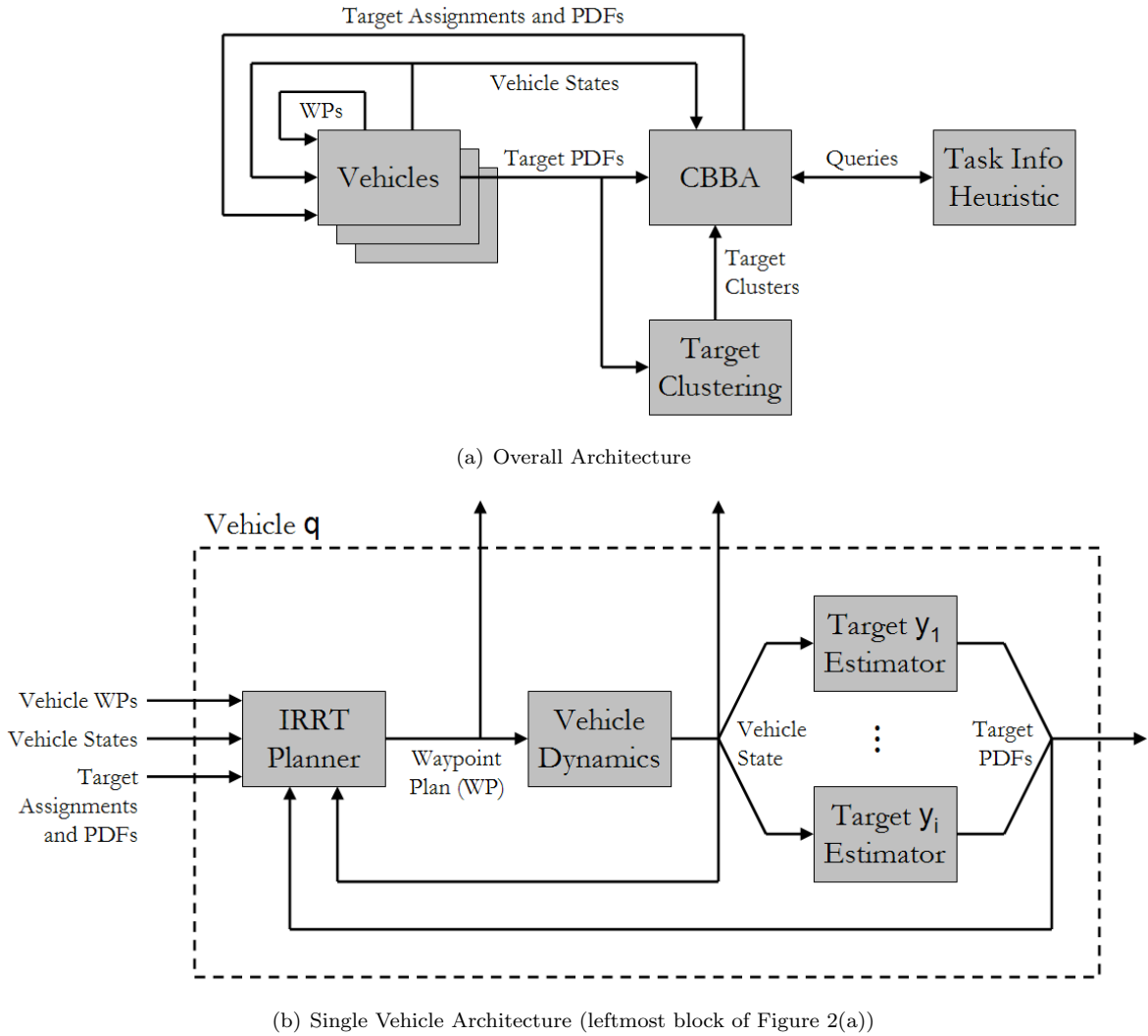(b) Single Vehicle Architecture (leftmost block of Figure 2(a))

Figure 2. **Block diagrams illustrating the overall CBBA+IRRT integrated architecture.**

feasible path, where cost may be an arbitrary function of the path with respect to the environment.[d]

The second component of informative path planning is adaptive sampling, a sequential decision problem in which a set of sensing configurations must be selected, given all previous information, to best reduce the uncertainty about an unknown quantity $\mathbf{y}_i$. By embedding information collection in the decision problem, adaptive sampling can significantly reduce the cost of obtaining an observation set that achieves a satisfactory estimate or classification. Suppose that by traversing a path σ, a set of observations $\mathcal{Z}_\sigma$ are generated by this stochastic process according to the likelihood function. We would like to assess the informativeness of σ via metrics on the posterior belief $p_{\mathbf{y}_i|\mathbf{z}}(\mathbf{y}_i|\mathcal{Z}_\sigma)$. However, one cannot anticipate the exact observation sequence that will result from traversing a selected path. The objective of informative motion planning then is to quantify the potential uncertainty reduction of the *measurement configuration* set M_σ and to embed such information collection in the planning problem, either as constraints or cost components. We specialize the objective for agent q to be safe arrival at the goal region $X_{\text{goal}}^{[q]}$ with minimum residual uncertainty on the target locations and in minimum time, which necessitates a trade-off between information colection and timely goal arrival.

---

[d]Common examples of cost functions are Euclidean distance, control effort, elapsed time, and combinations thereof. One may also, for example, penalize proximity to obstacles along a path in the cost function.

B. Motion Planning with RRTs

Before proceeding to the solution method we employ for the informative motion planning problem, we note several properties of the algorithm ours extends. For the general motion planning problem, sample-based methods, which seek to approximate connectivity in X_{free} by randomly sampling configurations and checking feasibility with respect to X_{obs}, have particularly pleasing computational properties. By construction, sample-based methods avoid issues of discretization granularity and are amenable to planning in high-dimensional state spaces. Furthermore, the performance of sample-based methods scale with the available computational resources. Finally, we note that the trajectory-wise constraint satisfaction afforded by sample-based methods leads to a significant reduction in computational complexity over that of standard optimization routines.

Within the class of sample-based methods, the Probabilistic Roadmap (PRM)[14] and Rapidly-exploring Random Tree (RRT)[15] algorithms have been prolific in the motion planning literature. The latter is especially effective for planning on nonholonomic and/or differentially constrained vehicles. Because the dynamics of sensor platforms are typically nontrivial, we build on the RRT as a baseline algorithm that is amenable to general vehicle constraints in the informative motion planning problem.

The RRT retains a tree-structured graph $\mathcal{T}$ of nodes emanating from a root node n_{root} and attempting to connect the collision-free space X_{free}. Each node $n \in \mathcal{T}$ is a tuple $n = \langle T_n, \sigma_n, v_n \rangle$, where $T_n \in \mathbb{R}^+$ is the node duration, $\sigma : [0, T_n] \mapsto X$ is the node state trajectory mapping instants in time to points in the state space X, and $v_n = \sigma_n(T_n) \in X$ is the terminal state, or *waypoint*.

1. Tree Expansion

Let $\rho : X \times X \mapsto \mathbb{R}^+$ be a distance metric comparing two states in X, $\texttt{Sample}(\cdot)$ be a function that generates random samples $\mathbf{x}_{\text{samp}} \sim \texttt{Sample}(\cdot)$ in the environment[e], and $\texttt{Nearest} : X \times \mathcal{T} \mapsto \mathcal{T}$ be a function that returns a subset of nodes $N_{\text{near}} \subset \mathcal{T}$ that are nearest to a specified state $\mathbf{x}_{\text{samp}}$ as measured by ρ. Expansion of the tree $\mathcal{T}$ proceeds by generating $\mathbf{x}_{\text{samp}} \sim \texttt{Sample}(\cdot)$, computing $N_{\text{near}} = \texttt{Nearest}(\mathbf{x}_{\text{samp}}, \mathcal{T})$, and attempting to "connect" nodes in N_{near} to $\mathbf{x}_{\text{samp}}$ using an as yet unspecified function $\texttt{Steer}$. For ease of discussion, we focus on the attempt to steer from one node $n_{\text{near}} \in N_{\text{near}}$ towards $\mathbf{x}_{\text{samp}}$, considering $v_{n_{\text{near}}}$ as the waypoint for this near node.

In open-loop RRT[15] (denoted here as OL-RRT), $\texttt{Steer}(v_{n_{\text{near}}}, \mathbf{x}_{\text{samp}}, T)$ samples input profiles $u : [0, T] \mapsto \mathcal{U}$ from an input space $\mathcal{U}$ over some finite duration $T \in \mathbb{R}^+$, simulates the resulting state sequences $\sigma_u : [0, T] \mapsto X$, and selects from them the feasible sequence $\sigma \in X_{\text{free}}$ that terminates nearest to $\mathbf{x}_{\text{samp}}$ as measured by ρ. A new node $n = \langle T, \sigma, v \rangle$ is then instantiated, with $v := \sigma(T)$, and added to the tree, i.e., $\mathcal{T} := \mathcal{T} \cup \{n\}$.

For dynamical systems with closed-loop control, open-loop RRT generates paths which may not be executable. In closed-loop RRT (CL-RRT),[16, 17] the state vector can be thought of as a concatenation of the dynamic and reference states of the vehicle. Instead of sampling open-loop input profiles, the reference states between $v_{n_{\text{near}}}$ and $\mathbf{x}_{\text{samp}}$ are connected (often by a simple guidance law), and the full closed-loop response σ of the vehicle and controller in response to the reference is generated. The termination criterion is that the portion of the *reference* state is within some distance of its counterpart at the sampled location. As before, if $\sigma \in X_{\text{free}}$, a node is instantiated and added to the tree. In addition to its executability properties, closed-loop RRT affords a notably accurate prediction of the state trajectory resulting from following a reference path,[18] making it well suited as a baseline algorithm for informative motion planning, where the measurement poses along paths must be predicted accurately.

Though we extend CL-RRT rather than OL-RRT, the two differ most fundamentally in the tree expansion phase. In the remainder of this work, we will maintain an agnostic view of RRTs and use the generic functions $\texttt{Steer} : (\mathbf{x}, \mathbf{x}'', \Delta t) \to \mathbf{x}'$ to denote the forward simulation from state $\mathbf{x}$ towards $\mathbf{x}''$ for Δt seconds resulting in state $\mathbf{x}'$ and $\texttt{Reached}(\mathbf{x}, \mathbf{x}'', t) \in \{0, 1\}$ as an indicator function denoting whether $\mathbf{x}$ and $\mathbf{x}''$ meet closeness criteria, or alternatively whether $t \in \mathbb{R}^+$ exceeds some threshold.

2. Path Selection

In describing the path selection process used in RRTs, it is useful to define several operators on nodes in the tree. Let the root operator $\texttt{root} : \mathcal{T} \mapsto \mathcal{T}$ return the root node of the tree. Due to the tree structure of the

---

[e]The proviso for admissible functions $\texttt{Sample}(\cdot)$ is that X is uniformly sampled with positive probability.

American Institute of Aeronautics and Astronautics

graph $\mathcal{T}$, the path connecting the root node $n_{\text{root}} = \texttt{root}(\mathcal{T})$ to any other node $n \in \mathcal{T}$ is unique. Therefore, let the parent operator $\texttt{pa} : \mathcal{T} \mapsto \mathcal{T}$ map a node to its parent, with $\texttt{pa}(n_{\text{root}}) := n_{\text{root}}$. By telescoping, let the ancestor operator $\texttt{anc} : \mathcal{T} \mapsto \mathcal{T}$ map a node to the set of its ancestors (i.e., all parents of parents back to n_{root}). Likewise, one can define a children operator $\texttt{chi} : \mathcal{T} \mapsto \mathcal{T}, n \to \{n' \in \mathcal{T} | n = \texttt{pa}(n')\}$ mapping a node to the set of nodes for which it is a parent. Finally, let the path operator $\mathcal{P} : \mathcal{T} \mapsto \mathcal{T}, \{n\} \to \texttt{anc}(n) \cup \{n\}$ return the union of any particular node with its ancestors. The cost function $c : \mathcal{T} \mapsto \mathbb{R}^+$ assigns positive, real-valued cost to all paths in the tree. The best path is denoted to be $\mathcal{P}_* = \mathcal{P}(n^*)$, with $n^* = \operatorname{argmin}_{n \in \mathcal{T}} c\left(\mathcal{P}(n)\right)$. Often, the cost of a path $\mathcal{P}(n)$ is composed of individual costs for each node $n' \in \mathcal{P}(n)$, as captured by the nodal path cost function $\psi : \mathcal{T} \mapsto \mathbb{R}^+$, and a cost-to-go term of reaching X_{goal} from the terminal waypoint v_n.

C. Information-Rich RRTs

In order to solve the informative motion planning problem, we employ the Information-rich Rapidly-exploring Random Tree (IRRT) algorithm,[19, 20] an extension of CL-RRT that embeds information collection as predicted from Fisher information of measurement poses along candidate paths. In the remainder of this section, we will briefly review the IRRT algorithm first presented in [19], along with its extensions to multiagent scenarios and Gaussian mixture beliefs introduced in [20].

Suppose each agent $q \in \mathcal{Q}$ is described by a tuple $q = \left\langle \mathcal{T}^{[q]}, \mathcal{P}_*^{[q]}, S^{[q]} \right\rangle$ with a tree of collision-free nodes $\mathcal{T}^{[q]}$, a selected path $\mathcal{P}_*^{[q]}$ that it executes, and a set of sensors $S^{[q]}$. We assume that each sensor has a specified measurement interarrival time, from which the observation times within each node may be anticipated. There also exists for each sensor $s \in S^{[q]}$ an indicator function $o_s : X \times Y \mapsto \{0, 1\}$ that captures whether s can generate an observation given a measurement pose $\mathbf{x} \in X$ and target state $\mathbf{y} \in Y$. Note that this observability indicator function subsumes a variety of possible sensor limitations, e.g., narrow field-of-view (FOV), limited range, and occlusions due to the presence of obstacles in X_{obs}.

1. Measurement Pose Prediction

To quantify the informativeness of paths in tree $\mathcal{T}^{[q]}$, the measurement pose sequence for each node is first anticipated. Consider a single node n, which is described by a state trajectory $\sigma_n : [0, T_n] \mapsto X$ of duration $T_n \in \mathbb{R}^+$. The measurement interarrivals time and temporal offsets (due to measurement in $\texttt{pa}(n)$) of sensors in $S^{[q]}$ are used to generate a set M_n of measurement configuration tuples. Each element $m \in M_n$ is a tuple $m = \langle t_m, \mathbf{x}_m, s_m \rangle$ composed of the intra-node measurement time $t_m \in [0, T_n]$, the measurement pose $\mathbf{x}_m = \sigma_n(t_m) \in X$, and the utilized sensor $s_m \in S^{[q]}$. In subsequent discussion, we refer to the process of anticipating a measurement pose sequence by the generic function $\texttt{MeasurementPoses}(\sigma, S)$.

2. Information Quantification

We now wish to quantify the informativeness of a node given its measurement pose sequence. Many information-theoretic measures exist for such a quantification;[21] we use Fisher information[22]

$$J_{\mathbf{z}}(\mathbf{y}_i) = \mathbb{E}_{\mathbf{z}} \left\{ \nabla_{\mathbf{y}_i} \left[\nabla_{\mathbf{y}_i} \log p_{\mathbf{y}_i, \mathbf{z}}(\mathbf{y}_i, \mathbf{z}) \right] \right\} \tag{4}$$

for its rich connection to the Cramér-Rao Lower Bound on the error covariance of unbiased estimators.[23] In general, the update for the *approximate* Fisher information $J_n(\hat{\mathbf{y}}_i)$ of target i across a node n is a function of the FIM at the parent node $J_{\texttt{pa}(n)}$, the measurement pose sequence M_n, and the current belief $b_{\mathbf{y}_i}$, i.e.,

$$J_n(\hat{\mathbf{y}}_i) = \texttt{FisherInformation}\left(J_{\texttt{pa}(n)}(\hat{\mathbf{y}}_i), \ M_n, \ b_{\mathbf{y}_i}\right). \tag{5}$$

Hereafter, we specialize to the case where beliefs have a K_i-mode Gaussian mixture model distribution and the target is assumed to be stationary. The strategy will be to maintain separate FIMs for each mode of target i and fuse them together in computing an uncertainty-based cost. Therefore, we consider $\mathbf{y}_{i,k} \sim \mathcal{N}(\mu_{i,k}, \Lambda_{i,k})$ and note that the update for the FIM across node n for target i, mode k is

$$J_n(\hat{\mathbf{y}}_{i,k}) = J_{\texttt{pa}(n)}(\hat{\mathbf{y}}_{i,k}) + \sum_{m \in M_n} o_{s_m}(\mathbf{x}_m, \hat{\mathbf{y}}_{i,k}) H^T(\mathbf{x}_m, \hat{\mathbf{y}}_{i,k}) R^{-1}(\mathbf{x}_m, \hat{\mathbf{y}}_{i,k}) H(\mathbf{x}_m, \hat{\mathbf{y}}_{i,k}), \tag{6}$$

American Institute of Aeronautics and Astronautics

where $H(\mathbf{x}_m, \hat{\mathbf{y}}_{i,k})$ is the linearized measurement matrix about the measurement pose $\mathbf{x}_m$ and estimated modal mean $\hat{\mathbf{y}}_{i,k} = \mu_{i,k}$, and $R(\mathbf{x}_m, \hat{\mathbf{y}}_{i,k})$ is likewise the measurement noise covariance matrix linearized at the same pair of states.

In the single-agent case, if we define

$$J_0(\hat{\mathbf{y}}_{i,k}) = \Lambda_{i,k}^{-1} = \mathbb{E}\left[\left(\hat{\mathbf{y}}_{i,k} - \mathbf{y}_{i,k}\right)\left(\hat{\mathbf{y}}_{i,k} - \mathbf{y}_{i,k}\right)^T\right]^{-1}, \tag{7}$$

the recursion initiates at the root node n_{root} with $J_{n_{\mathrm{root}}}(\hat{\mathbf{y}}_{i,k}) := J_0(\hat{\mathbf{y}}_{i,k})$, an exact relationship that arises from the Gaussianity of $\mathbf{y}_{i,k}$. Finally, let us denote the set of all FIMs along node n as $\mathbf{J}_n$.

Due to the additivity of (6), the multiagent IRRT extension, which accounts for the information content of paths selected by other agents in the network, is straightforward. For any agent $q \in \mathcal{Q}$, we denote its *information contribution* along path $\mathcal{P}(n)$ as

$$\Delta \mathbf{J}^{[q]}\left(\mathcal{P}(n)\right) \triangleq \mathbf{J}_n^{[q]} - \mathbf{J}_{n_{\mathrm{root}}}^{[q]}, \tag{8}$$

where operations on the FIM sets are elementwise (i.e., for each existing (i,k) pair). Therefore, factoring in the information contribution of other agents in the network is equivalent to initiating the FIM recursion at the root node as

$$\mathbf{J}_{n_{\mathrm{root}}}^{[q]} = \mathbf{J}_0 + \sum_{q' \in \mathcal{Q} \setminus \{q\}} \Delta \mathbf{J}^{[q']}\left(\mathcal{P}_*^{[q']}\right). \tag{9}$$

Fisher information is generally a matrix quantity, so we further require a scalar cost function to operate on Fisher information matrices J; many such cost functions exist.[24] For computational and geometric reasons,[25] we choose to use the A-optimality criterion $\mathtt{trace}(J^{-1})$ to assign cost to uncertain states, thereby rewarding uncertainty reduction. The *information error* with respect to $\mathbf{y}_i$ at the terminus of node n is defined as

$$\mathcal{I}_i(n) = \mathtt{trace}\left(\sum_{k=1}^{K_i} w_{i,k} J_n^{-1}(\hat{\mathbf{y}}_{i,k})\right), \qquad \sum_{k=1}^{K_i} w_{i,k} = 1, \qquad w_{i,k} > 0, \tag{10}$$

where the relative weights $w_{i,k}$ are exactly those maintained by the estimator running online. Note that the term within the $\mathtt{trace}$ operator is an approximation of the mixture covariance that does not consider the second set of terms in (1). The overall information error for node n is a convex combination of the errors for each target

$$\mathcal{I}(n) = \sum_i \gamma_i \mathcal{I}_i(n), \qquad \sum_i \gamma_i = 1, \quad \gamma_i > 0, \tag{11}$$

where the coefficients γ_i may denote the relative importance of target i in the mission.

3. Revised Algorithms

Now that the construction of the measurement sequence M_n and FIM set $\mathbf{J}_n$ along node n has been elucidated, we may expand our notion of the node tuple to $n = \langle T_n, \sigma_n, v_n, M_n, \mathbf{J}_n \rangle$. Furthermore, we revise the standard RRT algorithm descriptions to account for informativeness of paths in the tree.

The method IRRT-EXPAND($\mathcal{T}^{[q]}$) for expanding the tree is given in Algorithm 1. The function $\mathtt{Nearest}$ here subsumes several of the nearest node heuristics previously motivated.[19]

Given a tree $\mathcal{T}^{[q]}$ of candidate paths, the method IRRT-EXECUTE given in Algorithm 2 will continually expand the tree, select the best path $\mathcal{P}_*^{[q]}$, and execute it. The cost function $c : \mathcal{T}^{[q]} \mapsto \mathbb{R}^+$ it uses is of the form

$$c(\mathcal{P}^{[q]}(n)) = \rho(v_n, X_{\mathrm{goal}}^{[q]}) + \alpha \sum_{n' \in \mathcal{P}^{[q]}(n)} \psi(n') + \beta \mathcal{I}(n), \tag{12}$$

where $\rho : X \times X \mapsto \mathbb{R}^+$ is a distance metric, $\psi(\cdot)$ is a node trajectory cost, $\mathcal{I}(\cdot)$ is the information-based cost, and $\alpha \in [0,1]$ and $\beta \in [0,\infty)$ are relative weights capturing the importance of timely goal arrival and uncertainty reduction. In our case, $\psi(n') = T_{n'}$, i.e., the node path cost is simply the node trajectory duration.

American Institute of Aeronautics and Astronautics

Algorithm 1 IRRT-EXPAND($\mathcal{T}^{[q]}$)

1: $\mathbf{x}_{\text{samp}} \sim \texttt{Sample}(\cdot)$
2: $n_{\text{near}} \leftarrow \texttt{Nearest}(\mathbf{x}_{\text{samp}}, \mathcal{T}^{[q]})$
3: $\mathbf{x} \leftarrow v_{n_{\text{near}}}$
4: $t \leftarrow 0$
5: **while** $\mathbf{x} \in X_{\text{free}} \wedge \neg\texttt{Reached}(\mathbf{x}, \mathbf{x}_{\text{samp}}, t)$ **do**
6:    $\bar{\sigma}(t) \leftarrow \mathbf{x}$
7:    $\mathbf{x} \leftarrow \texttt{Steer}(\mathbf{x}, \mathbf{x}_{\text{samp}}, \Delta t)$
8:    $t \leftarrow t + \Delta t$
9: **end while**
10: **if** $\mathbf{x} \in X_{\text{free}}$ **then**
11:    $\bar{\sigma}(t) \leftarrow \mathbf{x}$
12:    $T \leftarrow t; \quad \sigma \leftarrow \bar{\sigma}; \quad v \leftarrow \sigma(T)$
13:    $M \leftarrow \texttt{MeasurementPoses}(\sigma, S^{[q]})$
14:    $\mathbf{J} \leftarrow \texttt{FisherInformation}(\mathbf{J}_{n_{\text{near}}}, M, b_{\mathbf{y}})$
15:    $n \leftarrow \langle T, \sigma, v, M, \mathbf{J}\rangle$
16:    $\mathcal{T}^{[q]} \leftarrow \mathcal{T}^{[q]} + \{n\}$
17: **end if**

Algorithm 2 IRRT-EXECUTE(q)

1: $n_{\text{init}} \leftarrow \left\langle 0, \mathbf{x}_0^{[q]}, \mathbf{x}_0^{[q]}, \emptyset, \mathbf{J}_0^{[q]}\right\rangle$
2: $\mathcal{T}^{[q]} \leftarrow \{n_{\text{init}}\}$
3: $\mathbf{x}^{[q]} \leftarrow \mathbf{x}_0^{[q]}$
4: **while** $\mathbf{x}^{[q]} \notin X_{\text{goal}}^{[q]}$ **do**
5:    Update $\mathbf{x}^{[q]}$ and target beliefs $b_{\mathbf{y}}$
6:    **while** time remaining in selection cycle **do**
7:      IRRT-EXPAND($\mathcal{T}^{[q]}$)
8:    **end while**
9:    UPDATEINFORMATION($b_{\mathbf{y}}, \texttt{root}(\tau_q)$)
10:   $n^* \leftarrow \operatorname{argmin}_{n \in \mathcal{T}^{[q]}} c(\mathcal{P}(n))$
11:   Announce $\mathcal{P}^{[q]}(n^*)$ to network, and execute it
12: **end while**

Algorithm 3 UPDATEINFORMATION($b_{\mathbf{y}}, n$)

1: $\mathbf{J}_n \leftarrow \texttt{FisherInformation}(\mathbf{J}_{\texttt{pa}(n)}, M_n, b_{\mathbf{y}})$
2: **for all** $n' \in \texttt{chi}(n)$ **do**
3:    UPDATEINFORMATION($b_{\mathbf{y}}, n'$)
4: **end for**

American Institute of Aeronautics and Astronautics

As the beliefs about the target locations are updated online, the measurement pose sequences along nodes will not change, but their informativeness will, for example, due to differences in anticipated observability or range to the target. The most current belief is cached over a planning cycle and is used to recompute the Fisher information at all nodes in the tree according to the recursive update described in Algorithm 3.

V.   Informative Task Assignment

The performance of dynamic search and track missions is typically measured in terms of the efficiency with which the agents involved reduce target estimation uncertainty. However, trajectories that achieve this uncertainty reduction are subject to a complex set of internal and external constraints, including dynamic constraints, environmental restrictions, and sensor limitations. By using the Information-rich Rapidly-exploring Random Tree (IRRT) algorithm[19] described in the previous sections a team of agents can quickly identify feasible, uncertainty-reducing paths that explicitly embed the latest target probability distributions, whilst satisfying these constraints. However, while IRRT is capable of handling multiple vehicles and targets,[20] algorithmic efficiency is lost when considering realistic large-scale ISR missions. Trajectories computed for such scenarios must embed both the vehicle routing problem (in selecting which distant targets to visit) and the constrained sensor problem (in finding a vantage point to view nearby targets), and become computationally intractable as the number of agents and targets increases. By incorporating a distributed approach that partitions the target environment into disjoint target sets, the combinatorial complexity (and thus computational burden) of the assignment problem is reduced. Furthermore, employing a task allocation algorithm ensures that targets assigned to agents or teams are deconflicted, and therefore the separate agents can complete tasks in parallel, improving the efficiency of the mission. This leads to a natural delineation of roles and responsibilities among agents, reducing the necessary communication load between them. This section describes a task allocation process that employs an information-based heuristic to allocate the targets amongst the agents, thus accounting for each agent's sensor limitations and vehicle constraints.

A.   Decentralized Task Allocation

The task allocation problem can be defined as follows. Given a heterogeneous team of N_a agents, and a list of N_t tasks with time-windows of validity, the goal of the task allocation algorithm is to find a conflict-free matching of tasks to agents that maximizes some global reward. An assignment is said to be free of conflicts if each task is assigned to no more than one agent. The "tasks" considered here are sets of clustered targets.

This task assignment problem can be written as the following integer (possibly nonlinear) program, with binary decision variables x_{ij} that are used to indicate whether or not task j is assigned to agent i:

$$\max \quad \sum_{i=1}^{N_a} \left(\sum_{j=1}^{N_t} c_{ij}(\mathbf{p}_i(\mathbf{x}_i, \boldsymbol{\tau}_i)) x_{ij} \right) \tag{13}$$

$$\text{subject to:} \quad \sum_{j=1}^{N_t} x_{ij} \leq L_t, \ \forall i \in \mathcal{I}$$

$$\sum_{i=1}^{N_a} x_{ij} \leq 1, \ \forall j \in \mathcal{J}$$

$$x_{ij} \in \{0,1\}, \ \tau_{ij} \in \{\mathbb{R}^+ \cup \emptyset\}, \ \ \forall (i,j) \in \mathcal{I} \times \mathcal{J}$$

where $x_{ij} = 1$ if agent i is assigned to task j, and $\mathbf{x}_i \triangleq \{x_{i1}, \ldots, x_{iN_t}\}$ is a vector of assignments for agent i, whose j^{th} element is x_{ij}. The index sets for i and j are defined as $\mathcal{I} \triangleq \{1, \ldots, N_a\}$ and $\mathcal{J} \triangleq \{1, \ldots, N_t\}$, representing the index sets for the agents and tasks respectively. The variable length vector $\mathbf{p}_i \triangleq \{p_{i1}, \ldots, p_{i|\mathbf{p}_i|}\}$ represents the path for agent i, an ordered sequence of tasks where the elements are the task indices, $p_{in} \in \mathcal{J}$ for $n = 1, \ldots, |\mathbf{p}_i|$, i.e. its nth element is $j \in \mathcal{J}$ if agent i conducts task j at the nth point along the path. The vector of times, $\boldsymbol{\tau}_i \triangleq \{\tau_{i1}, \ldots, \tau_{i|\mathbf{p}_i|}\}$, denotes the times that agent i proposes to execute the tasks in the path. Each agent can be assigned a maximum of L_t tasks due to limited resource constraints, and the maximum overall number of assignments achievable is given by $N_{\max} \triangleq \min\{N_t, N_a L_t\}$.

American Institute of Aeronautics and Astronautics

The current length of the path at each step is denoted by $|\mathbf{p}_i|$ and may be no longer than L_t. The global objective function is assumed to be a sum of local reward values for each agent i, while each local reward is determined as a function of the tasks assigned to the particular agent.

An important observation is that the scoring function in Equation 13 depends on the assignment vector $\mathbf{x}_i$, on the path $\mathbf{p}_i$, and on the times $\boldsymbol{\tau}_i$, which makes this integer programming problem significantly complex (NP-hard) for $L_t > 1$ due to all the inherent inter-dependencies.[6] Under these complex constraints, centralized planning methods quickly become infeasible favoring the development of distributed architectures (see [9] and references therein). In addition to alleviating the computational burden by parallelizing the planning process, distributed controllers often act on local information, making response times to changes in situational awareness significantly faster than those achievable under a purely centralized planner. As a result, distributed planning methods which eliminate the need for a central server have been explored.[26–29] Many of these methods often assume perfect communication links with infinite bandwidth, to ensure that agents have the same situational awareness before planning. In the presence of inconsistencies in situational awareness, these distributed tasking algorithms can be augmented with consensus algorithms[9–12, 30–36] to converge on a consistent state before performing the task allocation. Although consensus algorithms guarantee convergence on information, they may take a significant amount of time and often require transmitting large amounts of data.[37] In this work we use a particular decentralized task allocation algorithm called the Consensus-Based Bundle Algorithm (CBBA),[9] that is guaranteed to converge to a conflict-free solution despite possible inconsistencies in situational awareness, thereby reducing the amount of communication required between agents. The next section provides a brief description of CBBA.

B. Consensus-Based Bundle Algorithm

The Consensus-Based Bundle Algorithm (CBBA)[9] is a distributed auction protocol that provides provably good approximate solutions for multi-agent multi-task allocation problems over random network structures. CBBA consists of iterations between two phases: a bundle building phase where each agent greedily generates an ordered bundle of tasks, and a consensus phase where conflicting assignments are identified and resolved through local communication between neighboring agents. CBBA is guaranteed to converge to a conflict-free solution despite possible inconsistencies in situational awareness, and to achieve at least 50% optimality,[9] although empirically its performance is shown to be within 93% of the optimal solution.[38] The bidding process runs in polynomial time, demonstrating good scalability with increasing numbers of agents and tasks, making it well suited to real-time dynamic environments. The real-time implementation of CBBA has been demonstrated for heterogeneous teams and the algorithm has been extended to account for timing considerations associated with task execution.[7, 8, 39, 40]

C. Application to search and track missions

This work uses CBBA to allocate targets to the best suited agents. The score functions used within the CBBA task allocation framework explicitly account for the information that agents are able to obtain about their assigned targets, thus directly satisfying the objective of reducing uncertainty in the target search and tracking process while considering the complex constraints associated with realistic search and track missions. Prior to the task allocation process, the targets are grouped into clusters. These target sets or "tasks" can then be allocated to the individual agents using CBBA. A key advancement of the CBBA algorithm is a novel information-based scoring framework called the Task Information Heuristic (TIH), which embeds an approximation of the information gain in the assessed value of a target cluster to an agent or team. The following sections describe the clustering and information heuristic algorithms.

1. Target Clustering

The target clustering phase involves grouping the targets into sets using K-means clustering on the target means obtained from the latest target PDF estimates. The process for determining which clusters are appropriate is described in Algorithm 4. The algorithm runs K-means clustering iteratively, augmenting the number of clusters at each iteration, until the maximum cluster spread is below a maximum threshold S_{MAX}. The task set $\mathcal{J}$ is then created, one task for each cluster, retaining the key parameters of the clusters along with the individual target PDFs. The resulting tasks can then be allocated to the individual agents using CBBA.

American Institute of Aeronautics and Astronautics

Algorithm 4 CLUSTER-TARGETS($PDFs$)
___
  1: $N \leftarrow 0$
  2: **while** $S > S_{MAX}$ **do**
  3:     $N \leftarrow N + 1$
  4:     $C \leftarrow$ K-Means-Clustering($PDFs$, N)
  5:     $S \leftarrow \max_{c \in C}$ Cluster-Spread(c)
  6: **end while**
  7: $\mathcal{J} \leftarrow$ Create-Tasks(C, $PDFs$)
  8: **return** $\mathcal{J}$
___

2. Task Information Heuristic

The Task Information Heuristic (TIH) is embedded within the CBBA framework, and is designed to compute the cost that a particular agent receives for a particular task by following an information-rich trajectory. The TIH consists of the following two steps. First, the agent selects a starting location to enter the target cluster. For each vehicle, this is determined by computing the closest point on the outer edge of a sphere around the cluster's centroid, whose radius is given by the average cluster spread, with an additional margin to avoid starting inside any target's no-fly zone. The second step involves computing an information-rich trajectory to observe the targets within the cluster, starting from the previously computed start point. This is performed using a one-step optimization process, where the next best point is determined as the point within the reachable set that maximizes the average A-optimality of the Fisher Information Matrix over all the targets (see [41] for details on the specific algorithmic steps). The one-point optimization process is executed iteratively until the uncertainty of each target within the cluster is below some maximum threshold. The individual target Fisher Information Matrices are initialized using the inverses of the target covariance matrices obtained from the latest target PDFs, thus accounting for the actual acquired information thus far. The TIH does not explicitly consider obstacles and visibility constraints, however, it provides a good approximation of the locally optimal information-gathering trajectory. Finally, the estimated score for the task is computed as the sum of the values of the targets within the cluster minus the fuel resources consumed by executing the information-rich trajectory. The arrival time and task duration are approximated using the agent's arrival time at the selected start point, and the time required to traverse the optimized path, respectively. Using the estimated scores, task durations, and arrival times, CBBA is able to allocate the tasks to the individual agents producing target lists and expected schedules for each vehicle.

    The combination of IRRT+CBBA results in a novel multi-level algorithm which embeds information-rich trajectory planning within a task allocation framework, efficiently assigning targets and planning paths for teams of agents at the mission planning level. This real-time algorithm can leverage networks of mobile sensor agents to perform dynamic task reallocation as target estimates are updated, resulting in improved coordination and collaboration between agents while executing the mission. The CBBA task allocation algorithm employed is fully decentralized, making the system scalable to large teams of autonomous agents with diverse potential task sets and heterogeneous capabilities. The next section presents results for these combined algorithms, validating the proposed approach.
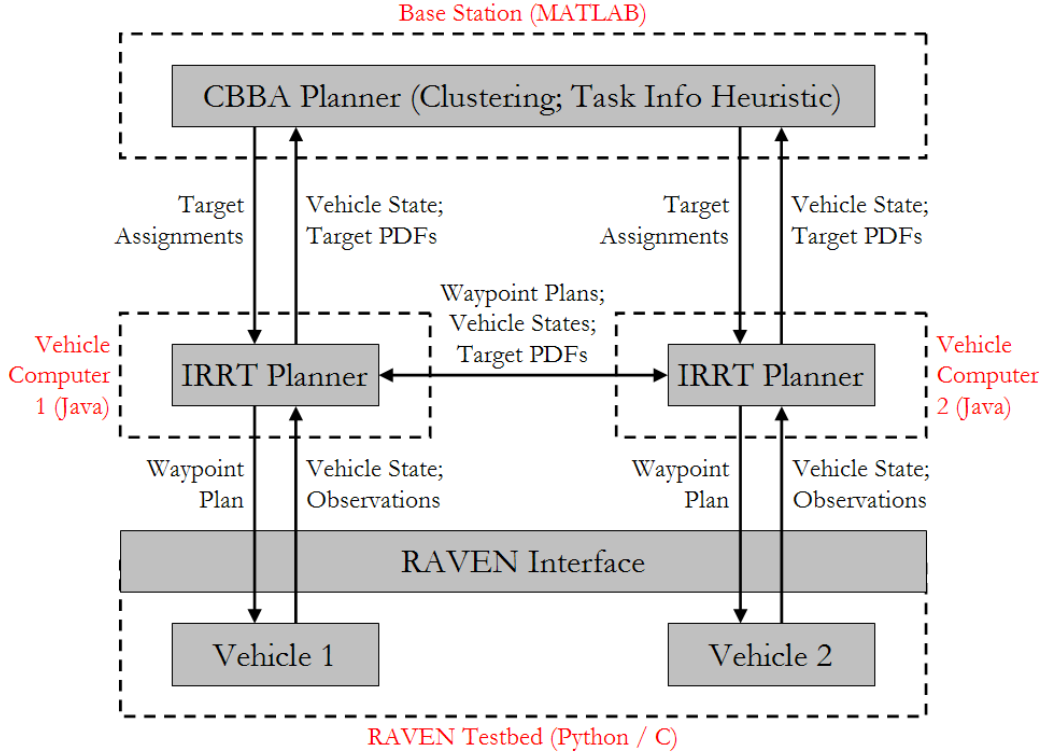
American Institute of Aeronautics and Astronautics

Figure 3. Software and hardware infrastructure used to implement the CBBA+IRRT algorithm.

VI. Results

In this section, simulation results demonstrate the effectiveness of the IRRT+CBBA algorithm in assigning tasks, such that targets are efficiently localized and identified.

A. Infrastructure

Figure 3 gives the software and hardware[f] infrastucture used to implement the algorithms in this paper. The infrastructure consists of three primary components:

- The base station, which operates the CBBA algorithm, along with clustering targets and querying the task information heuristic. It is also responsible for assessing whether tasks have been completed, as determined by the covariance of each target's uncertainty. The base station code is run in MATLAB, on an Intel Core i7 desktop with 12.0 GB of RAM. (Note that this is much more computational power than necessary to run this code.)

- The vehicle computers, which operate the IRRT planner for each individual agent. One computer is used per agent, whether real or simulated. The vehicle computer code is run in a multi-threaded Java application, on an Intel Core 2 laptop with 3.5 GB of RAM. When performing tests in simulation, a simulated vehicle is maintained in each vehicle computer. This software allows the user to specify the dynamic model, reference model, sensor model, environment, and list of targets, among other parameters.

- When hardware is included, all testing is done in the RAVEN testbed,[42] an indoor flight testbed for rapid development of control algorithms for aerial and ground vehicles. The RAVEN Interface software relays waypoints and state data between the planners and "vehicle wrappers," which send control inputs to each vehicle. Motion capture cameras provide high-fidelty state data.

---

[f]Future revisions of this paper will include hardware demonstrations of the simple example.

American Institute of Aeronautics and Astronautics

B. Example Specifications

In each simulation which follows, two agents, each with a single, heterogeneous sensor, are tasked with localizing a set of targets located in a three-dimensional environment. Each agent is modeled as a skid-steered vehicle, with the nonlinear system dynamics

$$
\begin{aligned}
x_{t+1} &= x_t + dt(1/2)(v_t^L + v_t^R)\cos\theta_t, \\
y_{t+1} &= y_t + dt(1/2)(v_t^L + v_t^R)\sin\theta_t, \\
\theta_{t+1} &= \theta_t + dt(v_t^R - v_t^L), \\
v_t^L &= \text{sat}\left(\bar{v}_t + 0.5\Delta v_t + w_t^L, 0.5\right), \\
v_t^R &= \text{sat}\left(\bar{v}_t - 0.5\Delta v_t + w_t^R, 0.5\right),
\end{aligned}
$$

where (x, y) is the vehicle position, θ is the heading, v^L and v^R are the left and right wheel speeds, respectively, sat is the saturation function, and $dt = 0.02$ sec. A variation of the pure pursuit controller[17] is applied, such that the vehicle is operated in closed-loop.[43]

One agent is always located at ground level ($z = 0.2$ m), while the other agent is always located in the air ($z = 2.0$ m), simulating fixed-wing aircraft dynamics. Both agents have a single camera sensor, with a $60°$ rectangular field of view and aimed in the vehicle's drection of motion. The ground-level agent's camera is pitched upward, while the aerial agent's camera is pitched downward; all targets appearing in the examples are located between these two altitudes.

A target is considered to have been *identified* when the trace of its covariance matrix is less than 0.01. When this occurs, the CBBA marks the task associated with this target as completed, and the target is no longer pursued by any agents.

In the simulations that follow, three different control algorithms are applied:

- **Euclidean:** The CBBA+IRRT algorithm, using the Euclidean heuristic to score tasks. Consider a possible pairing of an agent with a cluster of targets; under the Euclidean heuristic, this assignment has a score in CBBA of

$$
J = 100N_t - Fv_{\text{nom}}(t_{\text{arr}} + t_{\text{comp}}). \tag{14}
$$

  Here N_t is the number of targets in the cluster, $F = 2.5$ is a fuel coefficient, $v_{\text{nom}} = 0.4$ m/s is the nominal speed of the vehicle, t_{arr} is the estimated time for the vehicle to reach the cluster centroid, and t_{comp} is the estimated time for the vehicle to identify all targets within the cluster. This latter time is approximated as 2 seconds per target, plus the sum of the travel times from the cluster centroid to each target.

- **Information:** The CBBA+IRRT algorithm, using the Information Task Heuristic (Section V-C) to score tasks. This heuristic also assigns scores using (14); however, the times t_{arr} and t_{comp} are computed differently. The arrival time t_{arr} is the estimated time for the vehicle to enter a sphere centered on the cluster centroid; the radius increases as the distance of targets from the clutster centroid increases. The task completion time t_{comp} uses the optimization laid out in [41], terminating when the covariance threshold for identification is reached.

- **Baseline IRRT:** Only the IRRT algorithm is used. In this scenario, the base station assigned every non-identified target to every vehicle. Note that agents are not allowed to share observations; each agents maintains its own estimate of every target, and a target is not considered to be identified until some agent crosses the covariance threshold for that target.

Future revisions will consider a more complex version of IRRT, in which agents use the current paths of other agents to predict their future information gains.

Finally, note that the tree used in each IRRT planner is bounded at 1000 nodes.
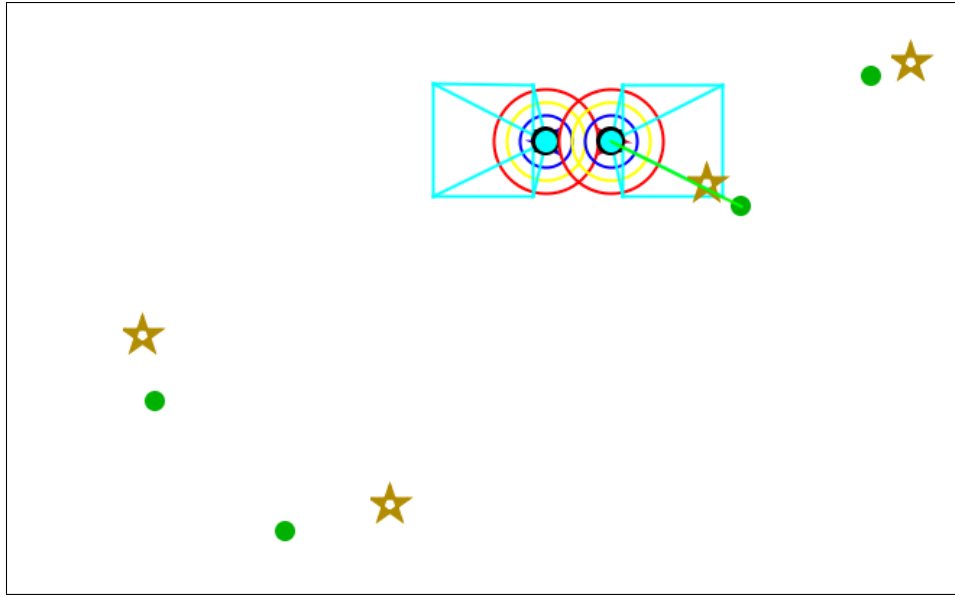
Figure 4. Environment used for the simple example. Targets are indicated by green circles; the initial target estimates provided to the agents are indicated by gold stars. Each agent is denoted by the concentric circles, with the blue lines indicating each agent's field of view. The green line emanating from the rightmost vehicle indicates a target currently in view.

C. Simple Example

The simple example demonstrates two of the primary conclusions of this paper:

- The CBBA+IRRT algorithm is able to complete missions more efficiently than using IRRT alone.

- The task information heuristic is able to assign agents to the tasks for which they are best suited.

The simple example environment is given in Figure 4. The horizontal direction is bounded by $[-2.1, 5.1]$ m, while the vertical direction is bounded by $[-2.7, 2.1]$ m. The ground-level agent (leftmost vehicle in figure, $z = 0.2$ m) starts at a position $(x, y) = (2.0, 1.0)$ m, oriented to the left, while the aerial agent (rightmost vehicle in figure, $z = 2.0$ m) starts at a position $(x, y) = (2.5, 1.0)$ m, oriented to the right. Each agent's camera is pitched $50°$ out of the level plane, with a range of 2.0 m. Each camera is designed to be more accurate at very close range (i.e., with a meter of the target).

The targets are located at 3-D positions $(-1.0, -1.0, 1.4)$, $(0.0, -2.0, 1.6)$, $(3.5, 0.5, 0.9)$, and $(4.5, 1.5, 0.7)$; this leads to a natural clustering of the two targets at lower-left and the two targets at lower-right.

First, we consider which cluster of targets will be first pursued by (or assigned to) each agent. Because of the range limitations of the cameras, each agent is best-suited to visit the cluster of targets closest to them in altitude. Thus, the desired assignment is for the aerial vehicle to visit the lower-left targets, while the ground-level vehicle visits the upper-right targets; this is the case when using the task information heuristic. However, using the Euclidean heuristic, the opposite assignment will take place, based on each agent's relative proximity to each cluster. Furthermore, using IRRT alone, both agents will pursue the upper-right cluster first due to its proximity, even though the observations they are taking then become redundant. Thus, the CBBA+IRRT using the task information heuristic makes the most efficient initial assignment of agents to target clusters.

The remaining questions are how long it takes each task assignment algorithm to make an assignment, and how long it takes each target to be identified by some agent. Table 1 gives the time per iteration, and the time need to identify each target, averaged over 10 trials. Note that by using CBBA+IRRT, the time needed to identify all targets is significantly reduced. However, the average runtime for CBBA using the information heuristic is significantly larger, without an appreciable improvement in performance. The performance gap is largely due to the initial placements and orientations of each agent, but will be explored further in future revisions.

American Institute of Aeronautics and Astronautics

Table 1. Simulation Results, Simple Example

Algorithm	Time per Iteration, s	Time to Identify (s)			
		1 Target	**2 Targets**	**3 Targets**	**4 Targets**
Baseline IRRT	0.0376	15.8	32.9	41.5	51.4
CBBA+IRRT, Euclidean	0.0454	5.5	9.8	24.5	31.6
CBBA+IRRT, Information	3.1169	21.5	25.6	32.8	36.7

Table 2. Simulation Results, Complex Example

Algorithm	# Targets	Time per Iteration, s	Time to Identify All Targets, s
Baseline IRRT	10	0.0695	70.6
CBBA+IRRT, Euclidean	10	0.1040	56.6
Baseline IRRT	20	0.0682	99.3
CBBA+IRRT, Euclidean	20	0.1040	86.1

D.   Complex Example

The simple example demonstrates two of the primary conclusions of this paper:

- The CBBA+IRRT algorithm is able to complete missions more efficiently than using IRRT alone.

- The CBBA+IRRT algorithm scales well as the problem complexity increases (here, in terms of the number of targets).

The complex example environment is given in Figure 5.

For these simulations, a randomly-generated set of obstacles and targets were generated for each trial, for a total of 5 trials; these sets are constant across all algorithms tested. Ten obstacles are included in each scenario, each restricting the motion of only the ground-level agent. Each scenario also has 10 or 20 targets, each consisting of 3 to 5 randomly weighted Gaussian models.[20] The range of each camera has been increased to 4 meters, with the pitch decreased slightly; however, the aerial agent has been given a more accurate camera than the ground-level agent.

Table 2 gives the time per iteration, and the time needed to identify all targets, averaged over the 5 trials. (Note that the task information heuristic was not considered here.) Again, by using CBBA+IRRT, the time needed to identify all targets is reduced when compared to using IRRT alone, regardless of the number of targets. Furthermore, the amount of time needed per task assignment iteration does not appreciably increase when the number of targets is doubled (the increase also scaled well when increasing from 4 targets in the previous example to 10 targets here). This may also be due in part to the increased density of targets, resulting in more incidental observations taking place.

VII.   Conclusion

In this paper, we have introduced a novel stratified planner that performs informative motion planning and task allocation in an effort to collect information from the environment in a way that minimizes sensing resource costs. The Information-rich Rapidly-exploring Random Tree (IRRT) was presented as a solution method for the informative motion planning problem, with the benefit of accommodating general constraint characterizations for the sensor platform dynamics, environment model, and sensor limitations. Because the storage and computational complexities of IRRT scale linearly with the number of target modes and number of nodes in the tree of trajectories, we have shown that stratification both dramatically reduces computational requirements and increases overall mission performance through task deconfliction. Furthermore, we have shown that in the Consensus-Based Bundle Algorithm (CBBA), a task information heuristic based on a reduced-order, gradient ascent optimization captures relative capability of the agents in a network when evaluating their ability to perform target localization tasks.
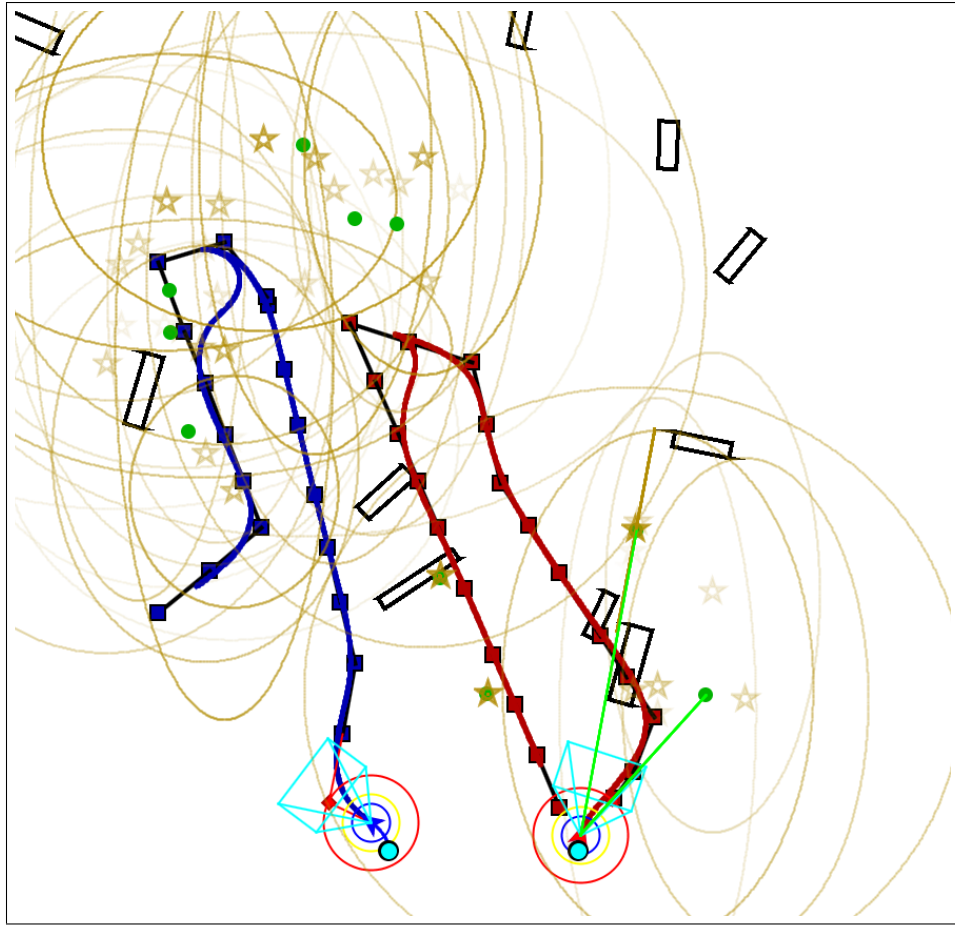
Figure 5. Sample environment used for the complex example; the locations of the obstacles (black) and target estimates (Gaussian models indicated by gold ellipses) vary with each trial. The initial waypoint sequence selected by each agent is also shown.

A number of extensions to this work are evident and will likely follow. In the current IRRT formulation, the beliefs are assumed to be Gaussian mixture model distributions. If the estimator is also in the form of a Gaussian sum filter, then the absence of measurements does not provide updates to the belief; however, such so-called negative information is valuable from an inference perspective. Therefore, we propose to examine the utility of negative information in the estimator and Gaussian mixture approximations thereof in the planner. The CBBA task information heuristic we have presented, while capturing the reduced-order properties germane to the task allocation problem, leaves room for improvement. In particular, we seek heuristics that are more computationally efficient and that are better able to approximate the full information collection prediction that is fundamental to IRRT.

Acknowledgments

References

[1] Kreucher, C., Kastella, K., and Hero, A., "Sensor management using an active sensing approach," *Signal Processing*, Vol. 85, March 2005, pp. 607–624.

[2] Meliou, A., Krause, A., Guestrin, C., and Hellerstein, J., "Nonmyopic informative path planning in spatio-temporal models," *Proceedings of the National Conference on Artificial Intelligence (NCAI)*, 2007.

[3] Singh, A., Krause, A., Guestrin, C., Kaiser, W., and Batalin, M., "Efficient planning of informative paths for multiple robots," *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[4] He, R., Prentice, S., and Roy, N., "Planning in Information Space for a Quadrotor Helicopter in a GPS-denied Environments," *Proc. ICRA*, Los Angeles, CA, 2008, pp. 1814–1820.

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

[6] Bertsimas, D. and Weismantel, R., *Optimization over integers*, Dynamic Ideas Belmont, MA, 2005.

[7] Ponda, S. S., Redding, J., Choi, H.-L., How, J. P., Vavrina, M. A., and Vian, J., "Decentralized Planning for Complex Missions with Dynamic Communication Constraints," *American Control Conference (ACC)*, Baltimore, MD, July 2010.

[8] Choi, H.-L., Whitten, A. K., and How, J. P., "Decentralized Task Allocation for Heterogeneous Teams with Cooperation Constraints," *American Control Conference (ACC)*, Baltimore, MD, July 2010, pp. 3057–3062.

[9] Choi, H.-L., Brunet, L., and How, J. P., "Consensus-Based Decentralized Auctions for Robust Task Allocation," *IEEE Transactions on Robotics*, Vol. 25, No. 4, August 2009, pp. 912–926.

[10] Ren, W., Beard, R. W., and Kingston, D. B., "Multi-agent Kalman consensus with relative uncertainty," *American Control Conference (ACC)*, Vol. 3, 8-10 June 2005, pp. 1865–1870.

[11] Ren, W. and Beard, R., "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, Vol. 50, No. 5, May 2005, pp. 655–661.

[12] Olfati-Saber, R. and Murray, R. M., "Consensus Problems in Networks of Agents with Switching Topology and Time-Delays," *IEEE Transactions on Automatic Control*, Vol. 49(9), 2004, pp. 1520–1533.

[13] Alighanbari, M. and How, J. P., "Decentralized Task Assignment for Unmanned Aerial Vehicles," *IEEE Conference on Decision and Control (CDC)*, 12–15 Dec. 2005, pp. 5668–5673.

[14] Kavraki, L., Svestka, P., Latombe, J., and Overmars, M., "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, 1996, pp. 566–580.

[15] LaValle, S. M., "Rapidly-Exploring Random Trees: A New Tool for Path Planning," Tech. Rep. 98-11, Iowa State University, October 1998.

[16] Frazzoli, E., Dahleh, M. A., and Feron, E., "Real-Time Motion Planning for Agile Autonomous Vehicles," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, January-February 2002, pp. 116–129.

[17] Kuwata, Y., Teo, J., Karaman, S., Fiore, G., Frazzoli, E., and How, J. P., "Motion Planning in Complex Environments using Closed-loop Prediction," *AIAA Guidance, Navigation, and Control Conference (GNC)*, Honolulu, HI, Aug 2008, (AIAA-2008-7166).

[18] Luders, B., Karaman, S., Frazzoli, E., and How, J. P., "Bounds on Track Error using Closed-Loop Rapidly-Exploring Random Trees," *American Control Conference (ACC)*, Baltimore, MD, June/July 2010, pp. 5406–5412.

[19] Levine, D., Luders, B., and How, J. P., "Information-rich Path Planning with General Constraints using Rapidly-exploring Random Trees," *AIAA Infotech@Aerospace Conference*, Atlanta, GA, April 2010, (AIAA-2010-3360).

[20] Levine, D. S., *Information-rich Path Planning under General Constraints using Rapidly-exploring Random Trees*, Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge MA, June 2010.

[21] Cover, T. M. and Thomas, J. A., *Elements of Information Theory*, Wiley, New York, 1991.

[22] Fisher, R. A., "On the Mathematical Foundations of Theoretical Statistics," *Philosophical Transactions of the Royal Society of London, Series A: Mathematical and Physical Sciences*, Vol. 222, 1922, pp. 309–368.

[23] Rao, C. R., "Information and the accuracy attainable in the estimation of statistical parameters," *Bulletin of the Calcutta Mathematical Society*, Vol. 37, 1945, pp. 81–89.

[24] Uciński, D., *Optimal Measurement Methods for Distributed Parameter System Identification*, CRC Press, 2005.

[25] Ponda, S. S., Kolacinski, R. M., and Frazzoli, E., "Trajectory Optimization for Target Localization Using Small Unmanned Aerial Vehicles," *AIAA Guidance, Navigation, and Control Conference (GNC)*, Chicago, IL, August 2009.

[26] McLain, T. W. and Beard, R. W., "Coordination Variables, Coordination Functions, and Cooperative-Timing Missions," *Journal of Guidance, Control, and Dynamics*, Vol. 28(1), 2005, pp. 150–161.

[27] Castanon, D. A. and Wu, C., "Distributed algorithms for dynamic reassignment," *IEEE Conference on Decision and Control (CDC)*, Vol. 1, 9-12 Dec. 2003, pp. 13–18.

[28] Curtis, J. and Murphey, R., "Simultaneous Area Search and Task Assignment for a Team of Cooperative Agents," *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2003 (AIAA-2003-5584).

[29] Shima, T., Rasmussen, S. J., and Chandler, P., "UAV team decision and control using efficient collaborative estimation," *American Control Conference (ACC)*, Vol. 6, 8-10 June 2005, pp. 4107–4112.

[30] Alighanbari, M. and How, J. P., "An Unbiased Kalman Consensus Algorithm," *AIAA Journal of Aerospace Computing, Information, and Communication*, Vol. 5, No. 9, Sept 2008, pp. 298–311.

[31] Moallemi, C. C. and Roy, B. V., "Consensus Propagation," *IEEE Transactions on Information Theory*, Vol. 52(11), 2006, pp. 4753–4766.

[32] Olshevsky, A. and Tsitsiklis, J. N., "Convergence Speed in Distributed Consensus and Averaging," *IEEE Conference on Decision and Control (CDC)*, 2006, pp. 3387–3392.

[33] Ren, W., Beard, R. W., and Atkins, E. M., "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, Vol. 27, No. 2, April 2007, pp. 71–82.

[34]Hatano, Y. and Mesbahi, M., "Agreement over Random Networks," *43rd IEEE Conference on Decision and Control*, 2004.

[35]Wu, C. W., "Synchronization and Convergence of Linear Dynamics in Random Directed Networks," *IEEE Transactions on Automatic Control*, Vol. 51, No. 7, 2006, pp. 1207–1210.

[36]Tahbaz-Salehi, A. and Jadbabaie, A., "On Consensus Over Random Networks," *44th Annual Allerton Conference*, 2006.

[37]Alighanbari, M. and How, J. P., "Decentralized Task Assignment for Unmanned Aerial Vehicles," *IEEE Conference on Decision and Control and European Control Conference (CDC-ECC '05)*, 2005.

[38]Bertuccelli, L. F., Choi, H.-L., Cho, P., and How, J. P., "Real-time Multi-UAV Task Assignment in Dynamic and Uncertain Environments," *AIAA Guidance, Navigation, and Control Conference (GNC)*, August 2009 (AIAA 2009-5776).

[39]Ponda, S. S., Choi, H.-L., and How, J. P., "Predictive planning for heterogeneous human-robot teams," *AIAA Infotech@Aerospace Conference*, April 2010 (AIAA-2010-3349).

[40]Johnson, L. B., Ponda, S. S., Choi, H.-L., and How, J. P., "Improving the Efficiency of a Decentralized Tasking Algorithm for UAV Teams with Asynchronous Communications," *AIAA Guidance, Navigation, and Control Conference (GNC)*, August 2010.

[41]Ponda, S. S., *Trajectory Optimization for Target Localization Using Small Unmanned Aerial Vehicles*, Master's thesis, Massachusetts Institute of Technology, 2008.

[42]Valenti, M., Bethke, B., Dale, D., Frank, A., McGrew, J., Ahrens, S., How, J. P., and Vian, J., "The MIT Indoor Multi-Vehicle Flight Testbed," *IEEE International Conference on Robotics and Automation (ICRA)*, 10-14 April 2007, pp. 2758–2759.

[43]Kuwata, Y., Teo, J., Fiore, G., Karaman, S., Frazzoli, E., and How, J. P., "Real-Time Motion Planning With Applications to Autonomous Urban Driving," *IEEE Transactions on Control Systems Technology*, Vol. 17, No. 5, September 2009, pp. 1105–1118.