An Extended Consideration of Joint Exploration and Tracking: JET*

Alexander Ivanov¹ and Mark Campbell²

Abstract—Autonomous exploration and multi-object tracking by a team of agents have traditionally been considered as two separate, yet related, problems which are usually solved in two phases: an exploration phase then a tracking phase. The exploration problem is usually viewed through an information theoretic framework where a robotic agent attempts to gather as much information about the environment or an Object of Interest (OI). Conversely, the tracking problem attempts to maintain precise location information about an OI over time. This work proposes a single framework which enables the multi-robot multi-object problem to be solved simultaneously. A hierarchical architecture is used to coordinate robotic agents in the tracking of multiple OIs while simultaneously allowing the task to remain computationally efficient. The primary contributions of this work are a probabilistic constraint on the tracked OIs’ covariances guarantees tracking performance throughout the entire mission. The automatic discovery of new OIs, a seamless transition to guaranteed tracking of discovered OIs, and the automatic balancing of exploration with the requirements of tracking.

I. INTRODUCTION

The problems of exploration and tracking are tightly coupled in many real world scenarios, including surveillance, Search and Rescue (SAR), and defense. In some SAR tasks, robots first need to locate potential victims and then subsequently track them if they are moving; examples include victims in burning forests, in the ocean, or in an alpine avalanche. Furthermore, the tracking task - that of maintaining object locations - typically takes precedence over the exploration task. While exploration and tracking have typically been solved in separate stages to simplify the joint problem’s complexity, this decoupled approach requires ad-hoc switching between stages which in turn makes guaranteeing tracking performance difficult. This work seeks to solve the joint problem of exploration and tracking under a unified framework, while guaranteeing tracking performance.

As background, research has been dedicated to the exploration problem with both single robotic agents as well as groups of homogeneous and heterogeneous agents; see [1] and the references within. Exploration applied to SAR includes probabilistic search, but does not usually consider tracking; several recent surveys have explored the state of this literature [1], [2]. This problem can also be framed as a mapping problem, as in [3]. Of particular relevance to this work, Al Khawaldah et al. consider the multi-robot exploration problem, but do not consider the detection or tracking of Objects of Interest (OIs) [4]. Similarly, Mottaghi and Vaughan use a particle filter to inform a team of robots of how to maximize the probability of detecting an OI [5], but also do not consider tracking.

A relevant variant of the exploration problem is termed the coverage problem, which implies ensuring that the largest possible area is sensed or ‘covered’. Most work on coverage is inapplicable but, Pimenta et al. generate a continuous time algorithm which seeks to guarantee both coverage and tracking [6]. Although [6] attempts to solve these problems simultaneously, no guarantees of tracking performance are given, no information theoretic sense of exploration is used, and the system is fully deterministic. Elston et.al also consider a multi-layered joint coverage and tracking problem [7]. Their work focuses on teams of ‘mother ships’ and ‘daughter ships’ which utilize a heuristic to partition a search area. Exploratory information is encoded by a heuristic, and robotic agents switch between pure tracking and pure coverage tasks.

A second related research area is the tracking problem from the viewpoint of traditional tracking metrics. Of particular interest, Ferrari et al. consider the tracking problem in a geometric manner and develop a closed form solution to the probability of detecting OIs under linearity and observability assumptions [8]. How et al. develop an RRT based planner which seeks to maximize track detections while achieving a goal [9]. Multi-agent multi-object problems have also considered questions such as data association and track assignment [1], but, in general, the tracking problem has been considered independent of the exploration or track detection problem.

Crucially, the literature rarely considers the joint exploration and planning problem. Instead, the joint prob-

¹information aii4@cornell.edu
²information mc288@cornell.edu
lem is usually solved in two stages which are assumed independent (i.e. first explore then track). This two-stage approach has several drawbacks. First, tracking accuracy may not be well maintained in the search phase, and OI tracks may be lost when the tracking phase begins. In many problems, robotic agents do not definitively finish the search task and continually find more OIs over time. This then necessitates switching between phases in a potentially ad-hoc way. Finally, the two stage approach fails to exploit the full ability of robot agents because agents may have excess control with which to continue to search for more OIs (victims) while maintaining track accuracy of already located OIs.

In this work, the joint exploration and tracking (JET) problem is presented under a probabilistic framework which enables the problem to be solved in a single stage. This allows agents to utilize their full control authority, while maintaining tracking accuracy, and seamlessly transitioning between exploration and tracking. In addition, chance constraints are utilized to guarantee tracking performance, as in many applications such as SAR, tracking is paramount. An exploration objective function is derived which generalizes others found in the literature, e.g. [8]. Finally, a hierarchical formulation is presented which enables the continuous optimal control problem to scale well, be solved efficiently, and maintain guaranteed tracking performance. Simulation results show the efficacy of the JET approach.

II. Problem Formulation

To adequately express the joint exploration and tracking problem, an optimization is posed in terms of an information theoretic performance metric and probabilistic constraints. Because robotic agents must first locate Objects of Interest (OIs), the performance metric must incorporate exploration/detection. Once OIs are detected, it is paramount to maintain tracking accuracy of all detected OIs. Thus, maintaining accuracy of known OIs is posed as a problem constraint.

Formally, the exploration portion of the joint problem is defined as optimization of the trajectory from time $t_0$ to $T$, i.e. maximize the value function:

$$ V_{info}(x(t_0), a(t_0), t_0) = \sup_{u(t), t \in (t_0, T)} J_{info}(x, a, z, u, \omega, \nu, t_0, T) $$ (1)

where $x(t)$ and $a(t)$ are the states of the set of robotic agents and OIs at time $t$. The variable $z(t)$ is a vector of sensor measurements taken at time $t$, and $u(t)$ are the controls given to the robotic agents. The variables $\omega(t)$ and $\nu(t)$ are stochastic noise affecting the state of the mobile objects (robots and OIs), and measurements respectively. Note that the dependence of the arguments on time is suppressed for compactness and denoted by the arguments $(t_0, T)$. The bold notation implies that this is an aggregated vector of all similar variables, i.e. $x$ is an aggregate of the states of all robotic agents.

The agents, OIs, and measurements are assumed to adhere to the following general dynamics equations:

$$
\begin{align*}
\dot{x}_i(t) &= f_i(x_i, u_i, \omega_i, t) & i \in \{1, ..., n\} \\
\dot{a}_j(t) &= f_j(a_j, \omega_j, t) & j \in \mathcal{T} \cup \bar{\mathcal{T}} \\
z_{i,j}(t) &= h_{i,j}(x_i, a_j, \nu_{i,j}, t) \\
u_i(t) &\in \mathcal{U}_i
\end{align*}
$$ (2)

where $f_i(\cdot)$ and $f_j(\cdot)$ are the dynamics functions of the $n$ robotic agents and $m$ OIs respectively. The measurements $z_{i,j}$ are measurements of the $j$th OI by the $i$th agent. The sets $\mathcal{T}$ and $\bar{\mathcal{T}}$ denote tracked and untracked OIs respectively; the dependence of the arguments on time is suppressed for compactness. Note that, the localization measurements for the robotic agents have not been defined. It is assumed that each robot can localize itself, and each robot state estimate, $\hat{x}_i(t)$, and covariance are available [10], [11].

Although the dynamics of the agents and OIs are continuous, realistic sensors are discrete; thus when referring to detections and sensing, discrete time instances are utilized with the convention $z_{i,j}^k = z_{i,j}(k \Delta t)$ and:

$$ z_{i,j}(t) = \begin{cases} 
0 & t \neq k \Delta t \\
0 & t \in \mathbb{N} \\
h_{i,j}(\cdot) & o.w.
\end{cases} $$

A. Objective Function

Because the framework of the problem requires agents to detect un-tracked OIs, an analysis of detection probability is in order. The probability of a single OI being detected at time $k$ is given as:

$$ P(O_j^k | 1| z_{j,1:k-1}^1, u_{1:k-1}^1) $$ (3)

where $(O_j^k = 1)$ denotes a positive detection at time instant $k$, $z_{j,1:k-1}^1$ is a vector of all sensor measurements taken of the particular un-tracked OI up until the current time instance $k - 1$, and $u_{1:k-1}^1$ is the sequence of controls given to the robotic agents up until time $k - 1$ (a zero-order-hold assumption). This expression can be decomposed using the law of total probability un-marginalize the robotic and un-tracked target states. Then, by sequentially conditioning the detection likelihood, robotic state, and target state on all other variables, Eq. (3) can be shown equivalent to:
\[
\int_{x^k} \int_{a^k_{un}} \left( p(O_j^k = 1|x^k, a_j^k) \right) \cdot p(x^k|u^{1:k-1})p(a_j^k|x_j^{1:k-1}, u^{1:k-1}) \, dx^k \, da^k_{un} \quad (4)
\]

where, \(x^k\) is the state of the robotic agents at time \(k\), and \(a_j^k\) is the state of the \(j\)th OI. Notice that the first term of \(4\) is the detector model while the second and third terms are the predictive agent and OI distributions respectively. The full derivation of \(4\) is shown in Appendix VII-A.

To understand the detection probability - Eq. \(4\)- more easily, consider that there is only one agent and its position is perfectly known (i.e. the integral with respect to \(x^k\) disappears). In addition, suppose there exists a perfect OI detector with a circular field of view of radius \(r\). In this case, Eq. \(4\) reduces to:

\[
\int_{(a_j^k|d(a_j^k, x^k) \leq r)} p(a_j^k|z_j^{1:k-1}, u^{1:k-1}) \, da_j^k \quad (5)
\]

In this case, the probability of detection simplifies to be the probability that the OI is within sensor range of the agent. The detection probability in \(5\) corresponds to, and therefore \(4\) generalizes, the sensor function in \([8]\).

Using \(4\) as an objective function maximizes the probability that a single OI is detected at the next time instant \(k\). In this scenario, \(J_{\text{info}}\) is written as:

\[
J_{\text{info}}(x, a, z, u, \omega, \nu, t_0, T) := \sum_{j \in \mathcal{T}} P(O_j^k = 1|z_j^{1:k-1}, u^{1:k-1})
\]

\[ t \in (t_0, T) \quad (6) \]

where the summation is taken over the set of untracked OIs \(\mathcal{T}\). The total number of tracked and untracked OIs, \(m\), is unknown, finite, and assumed to be \(m \leq n\). The challenge that \(m\) is unknown is addressed when discussing the hierarchical approximation of this problem in section III-B.

### B. Constraint on Tracking Performance

In traditional tracking, objects are typically tracked using a selection from a set of standard tracking estimators, referred to as Kalman Filtering (KF) techniques, which includes variants for both linear and non-linear dynamics \([10]\). In this work, tracked OI states are estimated using a KF, which models the state transition and measurement likelihoods as Gaussian distributions. As such, tracking performance can be analyzed via the covariance matrix:

\[
\mathbb{E}[(\hat{a}_j^k - a_j^k)^T (\hat{a}_j^k - a_j^k)] = \Sigma_{a_j}^k \quad \forall j \in \mathcal{T} \quad (7)
\]

where, \(\hat{a}_j^k\) is the posterior estimate of the \(j\)th tracked OI at time \(k\).

In the joint exploration and tracking problem, in order to guarantee tracking performance of already discovered OIs, a bound on the tracking error covariance, \(\Sigma_{a_j}^k\), must be defined for every OI in the tracked set \(\mathcal{T}\). Since all real-world sensors are imperfect, there is a chance that the detector does not detect a tracked OI at a particular time instance, i.e. missed or intermittent detection. The presence of intermittent measurements implies that no deterministic bound can be given for \(\Sigma_{a_j}^k\); instead, a bound on the expected value \(\mathbb{E}[\Sigma_{a_j}^k]\) is derived.

For cases of linear dynamics and intermittent measurements, the Algebraic Riccati Equation can be used to guarantee the existence of \(\mathbb{E}[\Sigma_{a_j}^k]\) \([12]\), \([13]\), \([14]\). Through these results, it can be shown that, for a certain range of probabilities of detection \(P(O_j^k = 1)\), the tracking error covariance \(\mathbb{E}[\Sigma_{a_j}^k]\) is bounded uniformly in \(t\) and there exists a finite steady state distribution for \(\Sigma_{a_j}^k\) \([12]\), \([14]\). Thus, one only needs to bound the probability of detecting an OI in order to provide a bound on its expected covariance matrix.

Consider now the following general bound on the tracking error of tracked OIs,

\[
P(O_j^k = 1|z_j^{1:k-1}, u^{1:k-1}) \geq 1 - \alpha \quad (8)
\]

where \(\alpha \in (0, 1)\) is a user set parameter. Notice that this bound utilizes the same expression as the objective function in Eq. \(3\). The constraint in \(3\) simply says that the joint robotic system has at least a probability of \(1 - \alpha\) of seeing OI \(j\) up to the discrete time instant \(k\), given the measurement and control histories. Assuming that OIs follow LTI dynamics with additive Gaussian noise, the results in \([12]\), \([14]\) can be leveraged and \(\mathbb{E}[\Sigma_{a_j}^k]\) can be computed for the chosen \(\alpha\). This calculation can be done via direct Monte-Carlo simulation in the results here, or by using the bound derived in \([12]\).

### III. A Hierarchical Approximation

The components of the proposed optimal control problem are concisely given in Eqs. \(2\), \(3\). In other words, one must maximize \(9\) subject to \(8\). The use of an exploratory metric as the objective in \(J_{\text{info}}\) implies that robotic explorers are opportunistic in their exploratory actions, but are required to maintain tracking. This problem as currently stated, is computationally intractable since the dimensionality of the control and
state spaces increase linearly with the number of robotic agents and OIs. In addition, the objective function and constraints are non-convex in general and are subject to the ‘Curse of History’ [15].

In order to alleviate these scaling issues, the joint exploration and tracking problem is solved in a hierarchical fashion. The hierarchical framework seeks to first provide optimal Next-Best-View (NBV) positions at time $T$ to each individual robotic agent. A lower level optimization then solves the continuous time optimal control problem for each agent independently in a distributed fashion. This hierarchical approach sacrifices optimality by only coordinating the terminal location of robots, but still solves the exploration and tracking problem jointly and provides probabilistic tracking guarantees. Optimal coordination throughout the time interval is not addressed, but individual local optimality for each agent is maintained.

A. Coarse dynamics for NBV goals

The NBV optimization is designed to give fast approximate exploration goals, at the time horizon $T$, to each robotic agent. These goals seek to maintain tracking performance. A coarse, discrete, linear approximation of the robotic dynamics is assumed. This approximation’s reachable set is assumed to be a subset of the true robotic dynamics, which allows for the application of bounded expected covariance due to [8].

optimal linear dynamics, the control sets $U_i$ and $U_j$ are not the same. Note that the OIs are assumed to follow LTI dynamics, which allows for the application of bounded expected covariance due to [8].

B. An approximate objective function for NBV goals

Given the coarse dynamics and measurement prediction in (9), the higher level exploration problem seeks to maximize:

$$\hat{J}_{info}(x^0, a^0, z^0, u^0, w^0, v^0) = \sum_{j \in \mathcal{T}} P(O_i^K = 1 | z^0, u^0)$$  \hspace{1cm} (10)$$

The approximate exploration objective defined by (10) maximizes the probability of seeing as many of the untracked OIs at the discrete horizon time $K$. More specifically, the optimal control, $(u^*)^r$, produced by maximizing Eq. (10), generates an optimal set of next-best viewpoints $(x^K)^*$. Finally, it is important to note that, if robotic agents have overlapping sensor fields of view at the terminal time $T$, the mutual information between agents must be considered. In general, this problem is related to Distributed Data Fusion and can be a difficult problem to solve [17]. Instead, this work proposes an additional constraint which requires that robotic agents be sufficiently separated at the terminal time $K$

$$||\hat{x}_i^K - \hat{x}_j^K|| \geq M \ \forall i \neq j$$  \hspace{1cm} (11)$$

where $M$ is a positive constant which depends on the sensor range.

C. A tighter tracking constraint

The NBV formulation, while at a coarser level, must continue to guarantee tracking for each OI in $\mathcal{T}$. Instead of satisfying Eq. (8), the NBV formulation requires that a robotic agent is assigned to each OI which is actively being tracked. The assigned agent is then required to guarantee an observation of its OI at the horizon time $T$. This constraint is formally defined as:

$$\exists i \ \forall j \in \mathcal{T} \ \text{s.t.} \ \ P(O_{i,j}^K = 1 | z_{i,j}^{1:k-1}, u_i^{1:k-1}) \geq 1 - \alpha$$  \hspace{1cm} (12)$$

Equation (12) is a tighter constraint than Eq. (8), and the assignment of agents necessitates the assumption that $m \leq n$. The higher level NBV problem can now be summarized as: maximize Eq. (10) subject to Eqs. (9)(12), which yields a set of optimal Next-Best-View points $(x^K)^*$. Notice that, as $T \rightarrow \Delta t$, the NBV problem
guarantees a probability of detecting known OIs at each time step, but greatly reduces the exploratory capability of agents, and makes exploration myopic.

D. Distributed optimization

Given the maximizer of the higher level problem, \( (x^K)^* = x^*(T) \), which implicitly assigns the robotic agents to OIs, the low level problem is considered. The constraint in Eq. (12), along with the results in [12], [13], [14], ensures that tracking performance is maintained for a short time horizon \( T \). Thus, the lower level problem no longer needs to consider tracking performance directly. Instead, the following constraint must be met:

\[
x(T) = x^*(T)
\]

Thus, the lower level path planning problem is solved independently by each robotic agent, or for the \( i \)th robot:

\[
\max_{u_i(t)} J_{LL}(x_i, a_i, z_i, u_i, \omega_i, \nu, t) \quad \text{s.t.} \quad \begin{align*}
\dot{x}_i(t) &= f_i(x_i, u_i, \omega_i, t) \\
\dot{z}_{i,j}(t) &= h_{i,j}(x_i, a_j, \nu_{i,j}, t) \\
x_i(T) &= x_i^*(T) \\
u_i(t) &\in U_i, \quad t \in (t_0, T) \quad j \in \mathcal{T} \cup \mathcal{T}
\end{align*}
\]  

(14)

E. The Joint Exploration and Tracking (JET) algorithm

A full description of a Joint Exploration and Tracking (JET) algorithm with probabilistic guarantees can now be given in Alg. 1.

Algorithm 1: Joint Exploration and Tracking: JET

1. \( x(0) = \text{InitializeStates}() \);
2. \( p(a) = \text{InitializeOIDist}(t) \);
3. \( \mathcal{T} = \emptyset \);
4. while true do
5.     UpdateOIPdf(\( \mathcal{I}(t), z(t) \)) ;
6.     if New OI detected then
7.         \( \mathcal{T} = \{ \max(T) + 1 \} \cup \mathcal{T} \)
8.     end
9.     if \( (t\%T)==0 \) or New OI detected then
10.        if \( (t\%T)==0 \) then
11.            \( t_0 = t \);
12.        end
13.        Assignment = SolveAssignment(\( \mathcal{I}(t) \)) ;
14.        \( (x^*(T+t_0)) = \text{SolveNVB}(\mathcal{I}(t)) \);
15.    end
16.    for \( i \in \{1, \ldots, n\} \) do
17.        \( (x^*_i(t), u^*_i(t)) = \text{SolvePath}(\mathcal{I}(t)) \);
18.        \( u^*_i = u^*_i(t + \Delta t/2) \);
19.    end
20.    \( t = t + \Delta t \) ;
21. end

Given the defined higher and lower level optimization problems outlined in Alg. 1, the specific modeling assumptions on the distributions of untracked and tracked OIs, robots’ states, and the form of the detection function are required to enable full implementation.

First, consider the detection function in Eq. (4). Most current literature assumes a perfect detector, which is an indicator function within a detection range or region [8]. Real sensors are seldom perfect detectors, and their accuracy may drop off as a function of range. This decrease in accuracy is especially true when using sensors, such as LIDAR and cameras, which become less accurate or resolve at a further range.

Instead of an indicator function, this work models the idealized detector as an un-normalized Gaussian Mixture (GM), which has several advantages. First, the detection probability can be cast as a function of distance, which is appropriate for LIDAR and camera sensors. Second, GMs can be composed to model complex behaviors. Finally, GMs have continuous derivatives.
Formally, the detector is modeled as a GM with a relatively small number of mixands (e.g. \( \leq 10 \)) and is quasi-concave.

\[
p(O^k = 1 | x^k, a^k) = \sum_i \gamma_i \cdot \exp \left( -\frac{1}{2} (a^k - (x^k - c_i))^T \Sigma_{1}^{-1} (a^k - (x^k - c_i)) \right)
\]

where \( \max(p(O^k = 1 | x^k, a^k = x^k)) = 1 \) (15)

**Note** that (15) is in terms of an arbitrary OI, \( a \), because both tracked and un-tracked OIs follow the same detection model.

For all tracked OIs as well as robotic agents, KF methods are used to estimate state distributions. These techniques, widely used in practice, yield tracked OI distributions that are multivariate Gaussian [10]. Robots are assumed to follow a unicycle motion model, and OIs have LTI dynamics [11].

\[
p(x^k_i | u^{1:k-1}) \propto \forall \in \{1,...,n\} \exp \left( -\frac{1}{2} (x^k_i - \bar{x}^k_i)^T \Sigma_x^{-1} (x^k_i - \bar{x}^k_i) \right) \]  (16)

\[
p(a^k_j | x^{1:k-1}, u^{1:k-1}) \propto j \in \tilde{T} \exp \left( -\frac{1}{2} (a^k_j - \mu_j)^T \Sigma_a^{-1} (a^k_j - \mu_j) \right) \]  (17)

Finally, versatility and smoothness make GMs a good candidate to represent the untracked OI distribution as well. Recent powerful tools which have been developed to approximate arbitrary distributions as GMs, [18], are of particular importance to the objective function in Eq. (6) because, if a robotic agent fails to detect an untracked OI at any time instant, a ‘negative’ measurement usually results in a posterior distribution which has no closed form. Therefore after every negative measurement, the resultant posterior distribution must be re-approximated as an analytic function in order to maintain computational efficiency and avoid having to discretize the exploration space. As a result, unknown OIs are modeled using GMs:

\[
p(a^k_j | x^{1:k-1}, u^{1:k-1}) = \sum_j \gamma_j \cdot \exp \left( -\frac{1}{2} (a^k_j - \mu_j)^T \Sigma_a^{-1} (a^k_j - \mu_j) \right) \]  (18)

Note that if tracked OIs are estimated using a Kalman filter, this implies that, as soon as a new OI is discovered and enters the tracked set \( T \), the tracked OI’s distribution is no longer modeled as a the GM in Eq. (18). Therefore, only negative measurements are taken of the un-tracked OIs.

Given these modeling assumptions, solution characteristics emerge, which are instrumental in allowing the lower level path planning problem to be solved quickly using standard non-linear optimization tools.

**Proposition 4.1:** Given equations (16), (17), and (18) if the sensor is modeled using a single mixand, then the constraint in Eq. (12) is convex in \( x_i(T) \) and quadratic in \( ||x_i(T) - a_j(T)|| \).

The proof of Proposition (4.1) is found in Appendix VII-B. The convexity of the constraint increases the optimization speed of direct transcription methods and is therefore valuable in making the JET algorithm real-time. The constraint convexity together with coarse linear dynamics provide a method to check for feasibility of the higher level problem, because the NVB problem is now a convex optimization. A similar, but weaker, result is true if the sensor is modeled by more than one mixand, assuming the detector GM is quasi-concave. This result allows for a much larger class of sensors to be accurately modeled while maintaining the convexity of the NVB problem. Both the proof and further discussion are in Appendix VII-B.

**V. SIMULATION RESULTS**

To analyze the behavior of the JET algorithm, three simulation studies are conducted. The first study shows the behavior of the low-level optimizer as a function of the excess control authority given to a robotic agent. The second study analyses the behavior of the high-level optimizer and its ability to distribute robotic agents and maintain tracking of OIs. The final study analyzes the behavior of the OI covariance matrix under the JET algorithm’s assumptions.

**A. Exploration as a function of control authority**

In this case study, a single robot is tasked with maintaining track accuracy of a single OI which has already been detected. Furthermore, there is an \( a \) priori probability of another OI being in the space, which is nearly uniform except for a peak at the bottom right corner of the map. This study seeks to exemplify the behavior of the low-level planning component of the algorithm as additional control authority is made available. In addition, the study analyzes the effects of the Open Loop Feedback (OLF) approximation on the optimality of the resultant path.

Recall that an OLF strategy does not take into account any expected measurements of the environment in the
future; as such, it is sub-optimal in general. Instead, with each new measurement of the map, the robot re-plans its path to incorporate the new information. Thus, the repeated re-planning addresses the sub-optimality of the OLF assumption. In this case study, the robotic agent moves deterministically and assumes process noise in the tracking model of the OI. The true behavior of the OI is deterministic to ensure repeatability of the experiment. In all cases, the following variables are kept constant: the prior distribution on untracked OIs, the realized tracked OI motion, and the initial conditions. The only variable is the control authority, $u_i(t)$, available to the robotic agent.

Figure 1 shows the results of this case study for three levels of control authority. The blue line is the robot's true realized path throughout the experiment, while the red line is the path taken by the tracked OI. The green lines in Fig. 1 show planned paths at particular instances in time separated by 0.5sec. Measurements of the environment are taken at 10Hz and the initial prior distribution on untracked OIs is overlayed as a heat map, with darker colors implying higher probability. Notice that, in each of the figures, the robot strays away from the expected OI path to explore the surrounding area. In all cases, the robot attempts to sense the $a priori$ peak in the bottom right and stays to the left of the true target trajectory. As the control authority increases ($a \rightarrow c$), the robot is able to explore further from the expected OI trajectory, which it must observe at intervals of 2.0sec. Subsequently, the planned green paths are longer. The planned and realized (blue) paths differ due to new measurements, which update the heat-map (not shown), but are very similar as measured in euclidean distance. This suggests that the OLF method approximates the optimal trajectory well even while ignoring future measurements.

B. Behavior and speed of the hierarchical approach

The second study seeks to qualitatively assess the ability of the JET algorithm to utilize a team of robots to both explore for unknown OIs and maintain localization of tracked OIs. The time horizon is set at $T = 2$sec and the detection/observation constraint probability for tracked OIs is set at $\alpha = .65$. To avoid clutter and confusion, Fig. 2 shows a series of snapshots of planned agent paths at three successive times ($t \in \{0, 2, 8\}$). Each figure shows five robotic agents searching a $12 \times 12$ area for, and subsequently tracking, three unknown OIs. The OIs move with stochastic dynamics. Initially, no information is known about the number or location of OIs except that they are not near the current location of the robotic agents. As such, the same un-informative prior distribution is assumed for the location of each OI's initial location. Since un-tracked OIs are independent and have not yet been detected by definition, their distribution develops in exactly the same way due to the negative measurements taken by robotic agents. The recursively updated posterior distribution of all un-tracked OIs is represented as a GM distribution, as shown by the heat-maps in Figure 2, where warmer colors represent a greater likelihood of an un-tracked OI being located at that point.

Agents are shown in red, with their initial position marked by a square and their planned final position and orientation shown as a triangle. True OI positions are shown as red dots, while estimated positions at $T$ are shown as yellow dots. Figure 2a shows the initial condition ($t = 0$sec) and all untracked OIs have no expected position. Figure 2b ($t = 2$sec) shows a single OI being tracked, while Fig. 2c ($t = 8$sec) shows all OIs being tracked.
The high level NBV optimization places the terminal positions of exploratory agents near the peaks of the unknown targets’ GM distribution (light green) while keeping the robots’ fields of view well separated. Conversely, at the time horizon $T$, assigned robots are required to be relatively near the expected terminal positions of their assigned OIs in order to satisfy the detection constraint. The low-level continuous time path planner then optimizes the robots’ trajectories, thereby improving information gathering, while satisfying the robots’ non-linear dynamics. Each figure shows that the terminal locations of purely exploratory robotic agents are well separated due to the separation constraint in Eq. (11). As a consequence, the planned exploratory trajectories tend to also be separated without requiring explicit coordination between robots at the low level.

A closer examination of the individual robotic paths (not shown) reveals a similar behavior to that seen in Fig. 1 where the robotic agent explores its environment when it has excess control authority, and tracks its assigned OI. The robotic agent’s path crosses the OI’s expected path approximately at each time horizon $T$. Figure 2c ($t = 8\text{sec}$) shows that after six time horizons, the robotic team successfully discovers all OIs and maintains tracking of each OI.

A new emergent ‘switching’ behavior from the JET algorithm can be seen in Fig. 2c. A robot, near $(-2, 2)$, which is tracking an OI near the robot’s current position, changes its role to track a newly discovered OI, near $(-2, 3)$. The robot and its new OI are encircled with dashed black lines. At the same time, a robot that was previously exploring is now tasked with tracking the ‘old’ OI (encircled with dashed white lines). Similar switching occurs when new OIs are discovered, and exploratory vs tracking roles are changed/updated.

C. Distribution of tracked OI covariance

In [13], the author discusses the discrete behavior of the distribution of covariance matrices in LTI systems with intermittent measurements. The work here uses an LTI model of OI behavior, which is noise driven in the velocity states, and therefore meets the assumptions in [12], [13]. Figure 3 shows a histogram of 10,000 Monte-Carlo, covariance norms of a single tracked OI after 20 time horizons. The OI is intermittently measured at each time horizon. This is equivalent to assuming that the coarse-dynamics tracking constraint in (12) is met for all 20 time horizons.

In this study, a probabilities of detection of 0.65 and 0.75 are analyzed. In addition, the same time horizon of $T = 2\text{sec}$, and OI process and measurement noise statistics are used as compared to the prior studies.
Fig. 3: The empirical PMF and CDF of $\Sigma$ norms at $t = 20T$.

Figure 3 shows only the smallest 95% of samples to avoid poor figure scaling. Notice that peak of the PMF, representing 64.8% and 75.4% of all cases respectively, are clustered at $< 0.1m^2$; this implies that the OI is still being well tracked. The means of the PMFs are $0.265m^2 (1 - \alpha = 0.65)$ and $0.154m^2 (1 - \alpha = 0.75)$. The CDFs show that 87.7% and 93.5% of covariance matrices respectively, have a norm less than $0.27m^2$. A norm of $0.27m^2$ indicates that the OI has a 99.7% chance of being within two sensor radii of its mean position. Summarizing all cases, if the probabilistic constraint of detection in Eq. (12) is met for each time horizon, a robotic agent maintains relatively accurate positional awareness of an OI in 87.7% and 93.5% respectively. This style of Monte-Carlo study can be used to inform a designer of the appropriate trade in time horizon length, tracking accuracy, and exploratory performance for a particular application of the JET algorithm.

VI. PRACTICAL AND REAL TIME IMPLICATIONS

The likelihood that OI tracking is lost if the constraint in Eq. (12) is met is small, yet the approximate hierarchical JET algorithm may lose tracking of an OI more frequently than the constraint implies. If the covariance of tracked OIs becomes large enough, a single robot may not be able to guarantee viewing its assigned OI with the required $(1 - \alpha)$ probability. In this case, the problem becomes infeasible. If a tracked OI’s covariance grows too large, a hybrid behavior is necessary to make the high level problem feasible once more. This hybrid behavior may require a tracking robot to switch to a pure tracking/following criterion and ignore the cost function in Eq. (14) completely until tracking accuracy is recovered. Fortunately, the convexity of the constraint in Eq. (12) allows for a fast feasibility check.

In addition, the lower level optimization is subject to limitations experienced by Non-Linear Program (NLP) solvers [19]. This work uses the MIT DRAKE toolbox along with a direct transcription method to solve the continuous time optimal control problem [20]. The initial guess provided to the NLP solver can make a large difference in computation time and even feasibility. In addition, the existence of a non-trivial gradient of the objective function in Eq. (9) is required for a timely solution. Although not common, a locally near-zero gradient may occur when a robotic agent is much faster than its assigned OI.

The average computational performance was evaluated on a 2.6GHz Intel i7-6600U processor in MATLAB using the DRAKE toolbox and SNOPT as the underlying NLP solver. The combined time of high level problem with 5 agents and a single robotic agent’s low level problem typically falls in a range of $\frac{5}{6}$sec. This excludes instances in which the NLP solver has difficulty converging due to a near-zero gradient, which can be compensated for by using appropriate problem scaling. In summary, these initial results show that the JET algorithm has potential as a real-time planner.

VII. CONCLUSIONS

This work proposes a framework which enables the multi-robot multi-object problem to be solved simultaneously. The JET algorithm allows a team of robots to search for OIs while a probabilistic constraint on the tracked OIs’ covariances guarantees tracking performance throughout the entire mission. Automatic discovery of new OIs, a seamless transition to guaranteed tracking of discovered OIs, and automatic balancing of exploration with the requirements of tracking are the primary novelties of the proposed algorithm. A novel hierarchical architecture is used to approximate the optimal control problem and coordinate robotic agents in the tracking of multiple OIs while simultaneously allowing the task to remain computationally efficient.

The JET algorithm enables each robotic agent to fully utilize its control authority to optimize exploratory behavior. At the same time, JET provides probabilistic guarantees on tracking performance, which are crucial in search and rescue scenarios. Simulation results show that the JET algorithm produces intuitive exploratory paths while maintaining tracking accuracy. In addition, the JET algorithm is able solve for continuous time optimal trajectories in a receding horizon fashion and has real-time performance potential.
This Appendix provides derivations and proofs for some of the formulas discussed in the paper.

A. Derivation of probability of detection

A derivation of the probability of detecting a single OI after a single time-step is given. Using the OLF assumption, the same expression is valid at the $K \cdot \Delta t = T$ with the exception that the process noise is scaled according to the OI’s linear dynamics:

$$p(O^k_j = 1 | z^{1:k-1}, u^{1:k-1})$$

(19)

Using the law of total probability we obtain

$$
\int_{x^k} \int_{u^k} p(O^k_j = 1, z^{1:k-1}, u^{1:k-1}) \\
\int_{x^k} \int_{u^k} p(z^{1:k-1}, u^{1:k-1})
$$

(20)

Focusing just the numerator, we can un-marginalize the location of the robot and the location of the untracked OI.

$$
\int_{x^k} \int_{u^k} p(O^k_j = 1 | z^{1:k-1}, u^{1:k-1}, x^k, a^k_j) dx^k da^k_j = \\
\int_{x^k} \int_{u^k} p(O^k_j = 1 | z^{1:k-1}, u^{1:k-1}, x^k, a^k_j) p(z^{1:k-1}, u^{1:k-1}, x^k, a^k_j) dx^k da^k_j
$$

(21)

The first multiplicative likelihood, $p(O^k_j = 1 | z^{1:k-1}, u^{1:k-1}, x^k, a^k_j)$, term is independent of previous target observations and previous control inputs. Thus Eq. (21) is equivalent to:

$$
\int_{x^k} \int_{u^k} p(O = 1 | x^k, a^k_j) * \\
p(z^{1:k-1}, u^{1:k-1}, x^k, a^k_j) dx^k da^k_j = \\
\int_{x^k} \int_{u^k} p(O = 1 | x^k, a^k_j)p(x^k | u^{1:k-1}) * \\
p(z^{1:k-1}, u^{1:k-1}, a^k_j) dx^k da^k_j
$$

(22)

Notice that the second multiplicative likelihood term, $p(x^k | u^{1:k-1})$, is the predictive robot distribution. Continuing, Eq. (22) is equivalent to:

$$
\int_{x^k} \int_{a^k_j} p(O = 1 | x^k, a^k_j)p(x^k | u^{1:k-1}) * \\
p(a^k_j | z^{1:k-1}, u^{1:k-1})p(z^{1:k-1}, u^{1:k-1}) dx^k da^k_j
$$

(23)
Noticing that the final multiplicative term is independent of the integration, and that it cancels with the denominator in Eq. (20), Eq. (4) gives the result.

**B. Proof of Prop. 4.1 and constraint properties**

Assume:

\[ x^k \sim \mathcal{N}(\bar{x}^k, \Sigma_{x^k}) \]
\[ \alpha^k \sim \mathcal{N}(\bar{\alpha}^k, \Sigma_{\alpha^k}) \]
\[ p(O = 1|\alpha^k, x^k) \sim \sum_i \xi_i \cdot \mathcal{N}(c_i, \Sigma_{\theta_i}) \]

An expression in terms of \( \bar{x}^k \) for the following set is desired:

\[ \{ \bar{x}^k | p(O = 1|z^{1:k-1}, u^{1:k-1}) \geq 1 - \alpha \} \]

From the derivation of \( p(O = 1|z^{1:k-1}, u^{1:k-1}) \), by using the properties of independent normal distributions, it can be shown that (25) is a function of \( \bar{x}^k \), and is proportional to a GM distribution. Since the tracked target is represented by a single Gaussian, the expression reduces to:

\[ \left\{ \bar{x}^k \right\} \{ \bar{x}^k \cdot \mathcal{N}(\bar{c} + \bar{\alpha}^k, \Sigma_{\text{OI}} + \Sigma_{x^k} + \Sigma_{\alpha^k}) \geq 1 - \alpha \} \]

By taking a log, and using Jensen’s inequality the following expression to inner bound for the set is valid:

\[ \sum_i \ln \left( \frac{\xi_i}{2\pi(\Sigma_{\text{OI}} + \Sigma_{x^k} + \Sigma_{\alpha^k})^{1/2}} \right) - \frac{1}{2} (\bar{x}^k - c_i - \bar{\alpha}^k)'(\Sigma_{\text{OI}} + \Sigma_{x^k} + \Sigma_{\alpha^k})^{-1}(\bar{x}^k - c_i - \bar{\alpha}^k) \geq \ln(1 - \alpha) \]

If a single Gaussian represents the sensor, the inequality reduces to:

\[ (\bar{\alpha}^k - c_0 - \bar{\alpha}^k)'\Sigma^{-1}(\bar{x}^k - c_0 - \bar{\alpha}^k) \leq -2\ln \left( \frac{(1 - \alpha)|2\pi\Sigma|^{1/2}}{\xi_0} \right) \]

where \( \Sigma \) is the sum of OI, agent, and sensor covariances. This is a convex quadratic in \( \bar{x}^k - \bar{\alpha}^k \) and Prop. 7.1 is proven.

Notice that if the right hand side of (27) is negative, the tracking constraint cannot be satisfied and the problem is infeasible.

Since \( c_i \) have so far been arbitrary in the more general case of (26), there is no formula for the set which satisfies the inequality in terms of \( \bar{x}^k \). Sufficient properties for the general GM sensor modal are now shown.

**Corollary 7.1:** Suppose the sensor is modeled as a quasi-concave GM, and that tracked and agent distributions are Gaussian. If there exists some \( \bar{x}^k \) which maximizes Eq. (12) and is feasible for \( \alpha = 0 \), then the set defined by Eq. (12) is convex and non-empty for \( \alpha \in [0, \alpha_{\text{max}}] \), \( \alpha_{\text{max}} < 1 \).

**Proof:** Corollary (7.1) is a direct consequence of the properties of quasi-convex functions.

Although this shows the convexity of the constraint in Eq. (12), this constraint is difficult to evaluate. The constraint defined in (26) is an under-bound on the true constraint set and is much simpler to compute. Notice that the function the the LHS in of Eq. (26) has similar properties to the softmax function, where the largest argument tends to dominate. Roughly speaking, a sufficiently clustered set of mixand means \( c_i \) should ensure that (26) has a solution and can be used instead of Eq. (12).

**C. The assignment problem**

In this formulation, in order to guarantee tracking performance the algorithm first assigns at least one robot to each tracked OI. To do this, a variant of the linear assignment problem can be solved. First define a mathematical graph (Fig. 4) with edges between robot nodes and OI nodes. In addition provide \( n - r \) fictitious nodes for each of the \( n - r \) robots which will remain un-assigned.
The graph only connects robots to targets which can be reached within $T$. The weight of these corresponding edges is simply the euclidean distance between them. In addition, each robot is connected to the $n - r$ dummy nodes with cost zero (some connections not shown for clarity).

The given setup is really a simplification of the true underlying assignment problem. As it has been stated above, this problem allows for a fast solution using the Hungarian algorithm or its variants. It does not take into account information gain differences from assignment. In addition, it does not capture behaviors such as "double teaming" an OI to ensure observation. Finally, it does not take into account either robot or OI uncertainties. This may be significant when considering that an OI with low uncertainty may be un-tracked some time while uncertain OIs must be tracked immediately. At the same time, the closest robot to a well known OI may gain more information by switching OIs.