

Distributed Hierarchical Control for State Estimation With Robotic Sensor Networks

Charles Freundlich, *Student Member, IEEE*, Yan Zhang, *Student Member, IEEE*,
and Michael M. Zavlanos, *Member, IEEE*

Abstract—This paper addresses active state estimation with a team of robotic sensors. The states to be estimated are represented by spatially distributed, uncorrelated, stationary vectors. Given a prior belief on the geographic locations of the states, we cluster the states in moderately sized groups and propose a new hierarchical Dynamic Programming (DP) framework to compute optimal sensing policies for each cluster that mitigates the computational cost of planning optimal policies in the combined belief space. Then, we develop a decentralized assignment algorithm that dynamically allocates clusters to robots based on the pre-computed optimal policies at each cluster. The integrated distributed state estimation framework is optimal at the cluster level but also scales very well to large numbers of states and robot sensors. We demonstrate efficiency of the proposed method in both simulations and real-world experiments using stereoscopic vision sensors.

Index Terms—Sensor Networks, Decision/Estimation Theory, Distributed Algorithms/Control, Optimal Control

I. INTRODUCTION

ROBOTIC sensors rely on mobility to gather information. Information acquisition can be subtask in a more complex robotic mission such as SLAM, or the end goal in, e.g., geostatistical surveying, environmental sampling, or mapping missions. The goal of this paper is to determine how a team of robots should collect information so that the aggregate uncertainty in a finite collection of hidden state vectors is minimized. Specifically, given prior beliefs of the geographic location of the hidden states, we seek an optimal sequence of observations which, when fused with the prior beliefs, minimizes the estimation uncertainty. This problem is known in the mobile robotics literature as distributed state estimation. The approach presented herein is similar to recent advancements in linear-Gaussian active sensing that operate in pose-covariance space [2], [3], except that here we propose a novel belief-space discretization that admits exact value iteration and can be readily incorporated in a hierarchical multi-robot controller, enabling decentralized information acquisition of very large collections of hidden states by large teams of robots.

A typical approach to active state estimation is to employ gradient descent methods to generate sensor trajectories and sequences of associated state observations that minimize an information theoretic objective of interest, such as the trace of

the covariance matrix. This is the approach followed, e.g., in mobile target tracking [4], [5], sparse landmark localization [1], [6], [7], and active SLAM [3], [8], [9]. Recently, [8] have shown that information-theoretic objectives may fail even to be monotonic in many active sensing tasks, removing performance guarantees for greedy control.

When planning multiple observations, the robot may need to reason over the combinatorial set of future probability distributions (pdfs), efficiently represent them, and solve Bellman’s equation. This is known as the “belief representation problem.” This problem, in the context of information acquisition, has received a great deal of attention recently, both when the robot state is observable [2], [10]–[13] and when it is only partially observable [3], [14]–[21]. The most widely used approach is to grow a tree with a prior distribution of the hidden states at the root, sampling in this way several sequences of possible future observations to obtain the set of reachable belief states at the leaves. Once the tree has been constructed, one simply selects the leaf with the lowest cost and traces it back to the root to obtain the optimal policy. The horizon length can be set *a priori* [2], [10], [11], [14] or, e.g., defined implicitly by a budget [12]. Note that nonmyopic active sensing problems, such as the problem addressed in this paper, implicitly exploit *a priori* knowledge rather than exploring the environment to discover new features. Exploration can be included as a first step as in [14], and incorporating this step within our approach is a subject of further research.

Two approaches that are fundamentally different from choosing a dynamic programming horizon are (i) to represent the reachable belief space with a finite set in a clever way, typically by making an assumption about the family of distributions of the hidden states [1], [2], [7], [13], [15], [16], or (ii) to avoid this representation problem altogether by working in policy space [17]–[19]. With respect to the latter, [19] defines a *generalized policy graph*, which nonetheless relies on belief space sampling. In fact, some kind of sampling is at the core of all point-based approaches stemming from the seminal paper [22].

Non-myopic active sensing for teams of decentralized robots often leads to decentralized Partially Observable Markov Decision Process POMDPs (dec-POMDPs) [20], [21]. To our knowledge, decentralization in this context refers to the execution of the planner, as its computation is always done offline at a central location due to the non-separable nature of the value function. Recently, [23] used sequential planning for decentralized active SLAM.

Common in the vast majority of approaches discussed

Charles Freundlich, Yan Zhang, and Michael M. Zavlanos are with the Dept. of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708, USA {charles.freundlich, yan.zhang2, michael.zavlanos}@duke.edu. This work is supported in part by the NSF award CNS #1302284. A preliminary version of this work can be found in [1]

above is that mobile sensor planning relies on sampling-based strategies, e.g., forward search, for partially [3], [14], [15], [15]–[21] and fully observable [2], [13] robots, or belief space sampling in the policy domain [17]–[19]. Sampling-based approaches will typically run into scalability issues due to one or more of the following reasons: (i) sparsity of information in the environment, which forces longer planning horizons, (ii) high dimensional unknowns, which make observation sampling inefficient, or (iii) large teams of robots, which significantly increase the size of the action space. To design an algorithm that avoids these pitfalls, in this paper we introduce a hierarchical approach that decomposes the set of M states into $P \ll M$ clusters, designs optimal controllers for each cluster, and then allocates those controllers among the N robots. Specifically, for every hidden state that needs to be estimated, we define a local Dynamic Program (DP) in the joint state-space of robot positions and state uncertainties that determines robot paths and associated sequences of state observations that collectively minimize the estimation uncertainty. Then, we divide the collection of hidden states into clusters based on a prior belief of their geographic locations, and, for each cluster, we define a second DP that determines how far along the local optimal trajectories the robot should travel before transitioning to estimating the next hidden state within the cluster. Finally, a distributed assignment algorithm is used to dynamically allocate controllers to the robot team from the set of optimal control policies for every cluster. At the cluster level, the problem that we solve can be considered a generalization of the TSP in that the robot must observe high dimensional unknowns over an infinite horizon at each site. This is important to note because our method incurs the same computational complexity as the TSP, $O(2^K)$, where K is the number of hidden states in the cluster. Moreover, the greedy extension of our method to multiple robots and multiple clusters only contributes constant complexity in the size of the robot team and in the number of clusters. In this way, our approach represents an approximation algorithm to a broad class multi vehicle generalized TSPs for which no good approximation exists. While we are not able to guarantee an optimality gap, we are able to bound computational complexity by limiting the cluster size, effectively “dividing and conquering” the hard problem into a number of tractable subproblems that can be solved exactly. We are not aware of any other non myopic method in the literature that can handle as many hidden states and robot sensors. We also illustrate these claims with experiments on real robots, which are a contribution in and of themselves as the first to demonstrate experimental multi-robot active sensing using stereo vision.

The paper is organized as follows: In Section II, we formulate the distributed state estimation problem addressed in this paper. In Sections III and IV we propose a local and a cluster DP to obtain controllers that are optimal at the cluster level. Section V presents the distributed auction mechanism that can efficiently allocate clusters to the robots in real-time. In Section VI, we simulate large robot teams carrying stereo vision rigs that localize hundreds of sparse landmarks. In Section VII, we report experiments on a team of two robots localizing eight landmarks.

II. PROBLEM FORMULATION

Consider a team of N mobile robots tasked with estimating M hidden state vectors $\{\mathbf{x}_i\}_{i=1}^M \subset \mathbb{R}^n$. Let \mathcal{W} and \mathcal{U} denote the configuration and action spaces of a single robot, and let $\phi: \mathcal{W} \times \mathcal{U} \rightarrow \mathcal{W}$ denote its (possibly nonholonomic) dynamical model. In this paper, the set \mathcal{W} can be any finite discretization of the robot’s configuration space, as long as transitions in that space can be assumed to be deterministic. Assume that the robots have access to a Normal prior distribution over the hidden states, with means $\{\hat{\mathbf{x}}_{i,0}\}_{i=1}^M$ and covariances $\{\Sigma_{i,0}\}_{i=1}^M$. Let \mathbf{y}_{ij} denote the observation of \mathbf{x}_i by robot j that is corrupted by zero mean Gaussian noise, specifically, $\mathbf{y}_{ij} = \mathbf{x}_i + \boldsymbol{\nu}_{ij}$, where $\boldsymbol{\nu}_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{ij})$. We assume that we have a model of \mathbf{Q}_{ij} that depends on both the state of the robot and the hidden vector. We will thus write \mathbf{Q}_{ij} as a (possibly discontinuous) mapping $\mathbf{Q}: \mathbb{R}^n \times \mathcal{W} \rightarrow \text{Sym}_{++}(n, \mathbb{R})$, where $\text{Sym}(n, \mathbb{R})$ denotes the set of symmetric matrices and the subscript denotes the restriction of this set to the set of positive definite matrices. Collectively, the team acquires a sequence of observations $\{\mathbf{y}_{ij,k}\}$ with measurement error covariances $\{\mathbf{Q}_{ij,k}\}$ from various vantage points along controlled trajectories $\{\mathbf{p}_{j,k}\} \subset \mathcal{W}$, where k denotes a time index. Hereafter, we will sometimes write $\mathbf{Q}(\mathbf{x}_i, \mathbf{p}_{j,k})$ instead of $\mathbf{Q}_{ij,k}$ to emphasize that \mathbf{Q} is actually a function of \mathbf{x}_i and $\mathbf{p}_{j,k}$.

Our goal is to minimize the variance in all hidden vectors as well as the distance the robots need to travel over the course of the controlled trajectories. We denote by $\psi: \mathcal{U} \rightarrow \mathbb{R}_+$ a metric that measures the distance an agent needs to travel as a result of actions in \mathcal{U} . Then, given a parameter $\rho \in [0, 1]$, we find a sequence of control inputs $\{\mathbf{u}_{j,k}\} \subset \mathcal{U}$ that solve

$$\max_{\{\mathbf{u}_{j,k}\}} \sum_{k \in \mathbb{N}} \gamma^k \left[(1-\rho) \sum_{i=1}^M \sqrt{\text{tr}(\Sigma_{i,k} - \Sigma_{i,k+1})} - \rho \sum_{j=1}^N \psi(\mathbf{u}_{j,k}) \right] \quad (1a)$$

$$\text{s.t. } \mathbf{p}_{j,k+1} = \phi(\mathbf{p}_{j,k}, \mathbf{u}_{j,k}) \quad \forall j = 1, \dots, N \quad (1b)$$

$$\Sigma_{i,k+1} = \begin{cases} \left(\Sigma_{i,k}^{-1} + \mathbf{Q}_{ij,k}^{-1} \right)^{-1} & \text{robot } j \text{ observes } \mathbf{x}_i \\ \Sigma_{i,k} & \text{else} \end{cases}, \quad (1c)$$

with initial conditions $\{\mathbf{p}_{j,0}\}_{j=1}^N$ and prior error covariance $\{\Sigma_{i,0}\}_{i=1}^M$. For each stage k , the expression inside the square brackets in (1a) is a trade off between variance reduction and distance traveled. The parameter ρ controls this tradeoff. The square root in this expression is used to compare equal units. Setting the discount factor $\gamma < 1$ ensures that the value function for the infinite horizon problem remains finite. Equation (1b) explicitly constrains the robot poses by the dynamics ϕ , while equation (1c) constraints the covariance dynamics by the Kalman Filter (KF) update for stationary hidden states.

The developments in the remainder of this paper toward solving problem (1) rely on the following assumptions:

Assumption II.1. *In this work we assume that the hidden state vectors are sparse and so are the observations. This means that even if the robot passively observes multiple hidden vectors at*

once, our plans are only optimal with respect to reducing the uncertainty of one at a time.

Assumption II.2. We assume that we have access to a noisy prior distribution $\mathcal{N}(\hat{\mathbf{x}}_{i,0}, \Sigma_{i,0})$ for each hidden vector. We assume that the hidden vector is a stationary process, thus we use the prior mean (along with the sensor configuration) to evaluate the observation uncertainty for all time $k \geq 0$. Similar to the formulation in [2], [13], this means that the dynamics of the error covariance matrices are deterministic, cf. (1). This also implies that $\mathbf{0} \preceq \Sigma_{i,k+1} \preceq \Sigma_{i,k}$ under KF dynamics.

III. LOCALIZATION OF A SINGLE TARGET

In this section, we propose a method to solve problem (1) for $N = M = 1$, thus we drop the references to i and j . We call this the *local DP*. To construct the local state-space, we approximate the space of reachable covariances by a finite set \mathcal{C} . Then, we define the local state space to be the product space $\mathcal{S} \triangleq \mathcal{W} \times \mathcal{C}$. We discuss the specifics of designing \mathcal{C} in Section III-A. A state $\mathbf{s}_k \triangleq (\mathbf{p}_k, \Sigma_k) \in \mathcal{S}$ is reachable from $\mathbf{s}_{k-1} \in \mathcal{S}$ if there exists a control input $\mathbf{u} \in \mathcal{U}$ that satisfies the joint dynamical equation $\Phi: \mathcal{S} \times \mathcal{U} \rightarrow \mathcal{S}$, given by

$$\mathbf{s}_k = \Phi(\mathbf{s}_{k-1}, \mathbf{u}) = (\phi(\mathbf{p}_{k-1}, \mathbf{u}), \Pi_{\mathcal{C}}[\Sigma_{k-1}^{-1} + \mathbf{Q}_{k-1}^{-1}]^{-1}), \quad (2)$$

where $\Pi_{\mathcal{C}}: \text{Sym}_{++}(n, \mathbb{R}) \rightarrow \mathcal{C}$. We give exact details of this projection in Section III-A and provide an example of the state-space transitions in Fig. 1 (a). Projection in (2) is necessary because we require that $\Sigma_k \in \mathcal{C}$, which is a finite subset of $\text{Sym}_{++}(n, \mathbb{R})$. The function Φ constitutes the transition function for the local DP. Then, denote by $R: \mathcal{S} \times \mathcal{U} \rightarrow \mathbb{R}$ the instantaneous reward from problem (1), given by

$$R(\mathbf{s}_k, \mathbf{u}) \triangleq (1 - \rho)(\text{tr}[\Sigma_{k-1} - \Sigma_k])^{1/2} - \rho\psi(\mathbf{u}). \quad (3)$$

In the remainder of this section, we will design a state space that is small enough for exact value leading to a desired stationary optimal policy $\mu^*: \mathcal{S} \rightarrow \mathcal{U}$. In particular, a *stationary optimal policy* is one that depends only on the current state of the system.

A. The Uncertainty State-Space and Transition Function

In this section, we discretize $\text{Sym}_{++}(n, \mathbb{R})$ to design the finite set \mathcal{C} . We emphasize that optimally sampling bounded subsets of $\text{Sym}(n, \mathbb{R})$ is an interesting and deep problem [24], and we do not provide a general framework. Our method works well for representing a specific bounded region of $\text{Sym}(n, \mathbb{R})$, which we call the *reachable covariance matrices* for the problem described herein.

We begin by the following lemmas; proofs are omitted due to space limitations.

Lem III.1. Let \mathbf{I} denote the $n \times n$ identity matrix and let $\mathbf{C} \in \text{Sym}_{++}(n, \mathbb{R})$. Then, $\mathbf{I} - (\mathbf{I} + \mathbf{C})^{-1} \in \text{Sym}_{++}(n, \mathbb{R})$.

Lem III.2 (Lemma 2.7, [25]). Let $n \in \mathbb{N}$. If $\mathbf{A}, \mathbf{B} \in \text{Sym}_{++}(n, \mathbb{R})$, then $\text{tr}(\mathbf{A}^{-1} + \mathbf{B}^{-1})^{-1} < \text{tr}\mathbf{A}$.

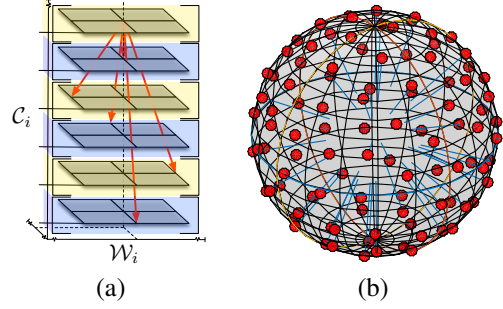


Figure 1. (a) An illustration of the state-space and transition for a local DP where \mathcal{W} is a NWSE grid and actions are to take an image and move north, south, east, west, or remain stationary. Red lines are drawn to represent each action, showing the transition of the state both in \mathcal{W} and in \mathcal{C} . The vertical axis represents the discrete nature three dimensional \mathcal{C} using colored regions, which are each represented by a single matrix in \mathcal{C} . (b) A fifty point discretization of \mathbb{R}^2 is represented by allowing 100 simulated charges confined to the 2-sphere \mathbb{S}^2 to come to equilibrium and discarding the points below an arbitrary equator. Charges are red dots, and the net electric force on each is plotted in blue.

The first implication of Lemma III.2 is that the trace of the largest instantaneous covariance bounds the maximum eigenvalue of the reachable covariance matrices. Define this bound as

$$\lambda_{\max} \leq \max_{\mathbf{p} \in \mathcal{W}} \text{tr} \mathbf{Q}(\hat{\mathbf{x}}, \mathbf{p}), \quad (4)$$

where we have written the measurement covariance as a function of the prior mean of the state estimate $\hat{\mathbf{x}} \in \mathbb{R}^n$, which is fixed during the planning phase, and the configuration of the sensor $\mathbf{p} \in \mathcal{W}$. Lemma III.2 also implies that the trace is decreasing with additional independent measurements. Therefore, the set of reachable covariances is bounded. Lemma III.2 and the resulting bound can be used to obtain a discretization of the set of possible maximum eigenvalues for the reachable covariance matrices. In particular, we define the logspace set of maximum eigenvalues

$$\mathcal{L} \triangleq \left\{ \lambda_{\max} e^{\kappa_{\mathcal{L}}(i - N_{\mathcal{L}})/N_{\mathcal{L}}} \mid i = 1, \dots, N_{\mathcal{L}} \right\} \subset \mathbb{R}_{++}, \quad (5)$$

where $N_{\mathcal{L}}$ is the cardinality of \mathcal{L} and $\kappa_{\mathcal{L}}$ is a sampling gain that controls how clustered the samples are toward zero. Note that λ_{\max} is the maximal element of \mathcal{L} . In (5), we sample in logspace as a heuristic; we have found empirically that the maximum eigenvalues of the filtered covariance matrices accumulate near zero.

To obtain a scalable discretization of the space of covariance matrices $\text{Sym}_{++}(n, \mathbb{R})$, we assume that $\lambda_{\max}(\Sigma)$ and its corresponding eigenvector are more important than any one of the other eigenvalues and eigenvectors. In particular, we assume that, in comparison to λ_{\max} , the other eigenvalues are roughly equal, thus all other eigenvalues can be parameterized by a number in the half open interval $\alpha \in (0, 1] \subset \mathbb{R}$ such that $\lambda_i \approx \alpha \lambda_{\max}$ for all $i = 2, \dots, n$. This choice alleviates the need to independently consider all possible combinations of eigenvectors corresponding to the nonprincipal eigenvalues. Define the set of ratios, which can be thought of as the set of possible inverse condition numbers, as

$$\mathcal{A} \triangleq \left\{ e^{\kappa_{\mathcal{A}}(i - N_{\mathcal{A}})/N_{\mathcal{A}}} \mid i = 1, \dots, N_{\mathcal{A}} \right\} \subset (0, 1]. \quad (6)$$

In (6), $N_{\mathcal{A}}$ is the number of eigenvalue ratios we sample and $\kappa_{\mathcal{A}}$ is a sampling gain that controls the ellipticity of the

confidence region associated with $(\lambda, \alpha) \in \mathcal{L} \times \mathcal{A}$. Again, the logspace discretization is a heuristic based on experience; KF error covariance matrices produced using robotic sensors are typically cigar-shaped, i.e., dominated by uncertainty in the direction of the principal eigenvector.

The set of possible principal eigenvectors is equivalent to the set of lines passing through the origin, known as the real projective space \mathbb{RP}^{n-1} . Let $N_{\mathcal{T}}$ be the number of samples needed to capture these possible directions for the principal eigenvalue. Because \mathbb{RP}^{n-1} can be formed by identifying antipodal points on any sphere, this problem can be approximately solved by placing $2N_{\mathcal{T}}$ point charges on a sphere of radius $\sqrt{\lambda_{\max}}$, allowing them to move until the ‘‘electrostatic forces’’ among them come to equilibrium, cutting the sphere along any equator, discarding one of the hemispheres, and saving the unit directions to each ‘‘charge’’ location on the other hemisphere. It is straightforward to build such a simulation, and for brevity we do not provide the specifics here. The result for 100 charges on the unit 2-sphere is shown in Fig. 1 (b).

To force the sampling density to be consistent, defined in terms of the surface area of the sphere used in the simulation, we create a set of sets $\{\mathcal{T}_{\lambda} \mid \lambda \in \mathcal{L}\}$; each element \mathcal{T}_{λ} is the set of unit vectors produced by the simulation using $\sqrt{\lambda}$ as the radius of the sphere. The number of elements in the largest, i.e., the set with the most elements, of these sets $\mathcal{T}_{\lambda_{\max}}$ is set to a user-specified number $N_{\mathcal{T}_{\lambda_{\max}}}$. The remaining sets $\{\mathcal{T}_{\lambda} \mid \lambda \in \mathcal{L}\}$ correspond to the other $N_{\mathcal{L}} - 1$ possible principal eigenvalues and, since λ_{\max} is the maximal element of \mathcal{L} , the sets $\{\mathcal{T}_{\lambda} \mid \lambda \neq \lambda_{\max}\}$ must have fewer elements than $\mathcal{T}_{\lambda_{\max}}$ so that the sampling density is the same. In particular, for some $\lambda \in \mathcal{L}$, the set \mathcal{T}_{λ} has $\left\lceil \frac{\lambda}{\lambda_{\max}} N_{\mathcal{T}_{\lambda_{\max}}} \right\rceil$ elements, where the ceiling function is used to ensure that the number of elements in \mathcal{T}_{λ} is positive.

Using the sets \mathcal{L}, \mathcal{A} , and $\{\mathcal{T}_{\lambda} \mid \lambda \in \mathcal{L}\}$, we can create a discretization of $\text{Sym}_{++}(n, \mathbb{R})$. The result, interpreted geometrically, is $N_{\mathcal{L}}$ concentric sets of cigar-shaped confidence ellipsoids with a variety of major diameters, defined by $\lambda \in \mathcal{L}$, ellipticities $\alpha \in \mathcal{A}$, and orientations $\mathbf{u} \in \mathcal{T}_{\lambda}$. The full covariance space can thus be described with a map $f: \mathbb{R}^{2+n} \rightarrow \text{Sym}(n, \mathbb{R})$ given by

$$f(\lambda, \alpha, \mathbf{u}) = \lambda \begin{bmatrix} \mathbf{u} & * \\ \mathbf{0} & \alpha \mathbf{I}_{n-1} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \alpha \mathbf{I}_{n-1} \end{bmatrix} \begin{bmatrix} \mathbf{u} & * \\ \mathbf{0} & \alpha \mathbf{I}_{n-1} \end{bmatrix}^{\top}, \quad (7)$$

where $*$ is any basis completion for \mathbb{R}^n , and \mathbf{I}_{n-1} is the identity matrix. The function f essentially builds a covariance matrix from the parameters supplied by \mathcal{A} and $\{(\lambda, \mathcal{T}_{\lambda}) \mid \lambda \in \mathcal{L}\}$. We define the covariance space as

$$\mathcal{C} \triangleq \{\mathbf{0}\} \cup f \left(\mathcal{A} \times \bigcup_{\lambda \in \mathcal{L}} (\{\lambda\} \times \mathcal{T}_{\lambda}) \right). \quad (8)$$

The $\mathbf{0}$ covariance is an artificial state that we include in \mathcal{C} to denote that no more uncertainty remains in the variable being estimated, i.e., estimation is complete to the user-specified tolerance, defined as $\frac{1}{2} \min \{\lambda \in \mathcal{L}\}$.

The projection operator $\Pi_{\mathcal{C}}$ guarantees that the fusion of the current covariance state Σ_k and the new measurement covariance \mathbf{Q}_k is a member of \mathcal{C} . In particular, for some

$\Sigma \in \text{Sym}_{++}(n, \mathbb{R})$, $\Pi_{\mathcal{C}}$ first computes the principal eigenvalue λ_{\max} and its corresponding normalized eigenvector \mathbf{u}_{\max} . Then, it rounds λ_{\max} to the closest element in $\mathcal{L} \cup \{0\}$. Call this map $\Pi_{\mathcal{L}}: \text{Sym}(n, \mathbb{R}) \rightarrow \mathcal{L} \cup \{0\}$. If $\Pi_{\mathcal{L}}(\Sigma)$ is nonzero, there will be some $\lambda' \in \mathcal{L}$ that is closest to λ_{\max} , and $\Pi_{\mathcal{C}}$ then finds the element $\mathbf{u}' \in \mathcal{T}_{\lambda'}$ that forms the largest magnitude inner product with \mathbf{u}_{\max} . Call this map $\Pi_{\mathcal{T}_{\lambda'}}: \text{Sym}(n, \mathbb{R}) \rightarrow \mathbb{S}^{n-1}$, where $\mathbb{S}^{n-1} = \{\mathbf{u} \in \mathbb{R}^n \mid \langle \mathbf{u}, \mathbf{u} \rangle = 1\}$. In particular, $\mathbf{u}' = \Pi_{\mathcal{T}_{\lambda'}}(\Sigma) \triangleq \max_{\mathbf{u} \in \mathcal{T}_{\lambda'}} |\langle \mathbf{u}, \mathbf{u}_{\max} \rangle|$. Finally, $\Pi_{\mathcal{C}}$ computes the ratio of $\lambda_{\min}(\Sigma)$ with $\lambda_{\max}(\Sigma)$ and finds the closest element $\alpha' \in \mathcal{A}$ to that ratio. Call this map $\Pi_{\mathcal{A}}: \text{Sym}_{++}(n, \mathbb{R}) \rightarrow (0, 1]$. By this construction, it holds that $(\alpha', \lambda', \mathbf{u}') \in \mathcal{A} \times \bigcup_{\lambda \in \mathcal{L}} (\lambda \times \mathcal{T}_{\lambda})$, so that its image of this triplet under f from (7) is guaranteed to be a matrix in \mathcal{C} . In particular, the projection map is given by $\Pi_{\mathcal{C}}(\Sigma) = f \left(\Pi_{\mathcal{L}}(\Sigma), \Pi_{\mathcal{A}}(\Sigma), \Pi_{\mathcal{T}_{\Pi_{\mathcal{L}}(\Sigma)}}(\Sigma) \right)$,

IV. LOCALIZATION OF MULTIPLE TARGETS

Assume the collection of all hidden states discussed in Section II is divided into clusters. Temporarily let M denote the number of hidden states in a particular cluster. Denote by $S_i = \mathcal{W}_i \times \mathcal{C}_i$ the state space local to the i -th hidden vector in the cluster. Each S_i has $N_{S_i} = N_{\mathcal{W}_i} (1 + N_{\mathcal{A}_i} \sum_{\lambda \in \mathcal{L}_i} N_{\mathcal{T}_{\lambda}})$ individual states.

Let $\mathcal{E}_i \subset S_i$ denote the set of initial states that the robot can visit when it first arrives at \mathcal{W}_i to observe \mathbf{x}_i . We assume that the first observation of \mathbf{x}_i will occur at the boundary of the convex hull of local pose space $\partial \mathcal{W}_i$. Therefore, we define the *entry points* to the i local state space as $\mathcal{E}_i \triangleq \{(\mathbf{p}, \Sigma_{i,0}) \mid \mathbf{p} \in \partial \mathcal{W}_i\}$. Let us index the states in the set \mathcal{E}_i using integers $j \in \{1, \dots, |\mathcal{E}_i|\}$ such that $j \xrightarrow{1 \rightarrow 10} \mathbf{s}_j \in \mathcal{E}_i$. Since in the neighborhood of every hidden state, the robots follow the local optimal policy determined in Section III, a robot that begins observing the i -th hidden state at the entry point $\mathbf{s}_j \in \mathcal{E}_i$ and has spent k steps at that particular state has a known global location and local covariance matrix. To keep track, let $\Phi_i: S_i \times \mathcal{U}_i \rightarrow S_i$ and $\mu_i^*: S_i \rightarrow \mathcal{U}_i$ denote the i -th local transition function from (2) and optimal policy. Then, define the k -times recursive local optimal transition function as $\Phi_i^{*k}: \mathcal{E}_i \rightarrow S_i$, given by

$$\begin{aligned} \Phi_i^{*0}(\mathbf{s}_j) &= \mathbf{s}_j \\ \Phi_i^{*1}(\mathbf{s}_j) &= \Phi_i(\mathbf{s}_j, \mu_i^*(\mathbf{s}_j)) \\ &\vdots \\ \Phi_i^{*k}(\mathbf{s}_j) &= \Phi_i \left(\Phi_i^{*(k-1)}(\mathbf{s}_j), \mu_i^* \left(\Phi_i^{*(k-1)}(\mathbf{s}_j) \right) \right). \end{aligned} \quad (9)$$

The function Φ_i^{*k} denotes the local state in S_i in which the robot will land when starting observing \mathbf{x}_i at from entry point $\mathbf{s}_j \in \mathcal{E}_i$ and after following the local optimal policy μ_i^* for k time steps.

In the cluster DP, there are M available actions, one for every hidden state in the cluster. In particular, for a robot observing \mathbf{x}_i , action \mathbf{u}_i is simply to continue observing the same state along the local optimal policy. Note that, since the optimal policy is stationary, there is always a local optimal action to take. The following proposition shows that any local optimal trajectory reaches an absorbing state so that the iteration $k \mapsto k + 1$ terminates.

Proposition IV.1. $\forall i \in \{1, \dots, M\}$ and $\mathbf{s}_j \in \mathcal{E}_i$, there exists $K_i \in \mathbb{N}$ such that $\Phi_i^{*K_i}(\mathbf{s}_j) = \Phi_i^{*k}(\mathbf{s}_j) \forall k \geq K_i$.

Proof: Since each \mathcal{S}_i is finite and the optimal policy $\mu_i^* : \mathcal{S}_i \rightarrow \mathcal{U}_i$ is fixed, failure to converge is possible only if the optimal policy drives the robot in a cycle. This is a contradiction to the optimality of μ_i^* . To see why, consider an optimal local trajectory such as $\{\dots, \mathbf{s}, \mathbf{s}', \dots, \mathbf{s}, \dots\} \subset \mathcal{S}_i$. If $\text{tr}\Sigma < \text{tr}\Sigma'$, then the transition from \mathbf{s} to \mathbf{s}' contradicts Lemma III.2. Similarly, if $\text{tr}\Sigma > \text{tr}\Sigma'$, then the transition from \mathbf{s}' to \mathbf{s} is a contradiction. If $\text{tr}\Sigma = \text{tr}\Sigma'$, then the robot must have moved in a loop $\{\dots, \mathbf{s}, \mathbf{s}', \dots, \mathbf{s}, \dots\}$ without changing the uncertainty, i.e., energy was consumed for no gain in reward, a contradiction to the optimality of μ_i^* . ■

As a result of Proposition IV.1, we do not store local optimal trajectories longer than $\max_{i \in \{1, \dots, M\}} K_i$.

We can now define the state-space for the cluster DP. The cluster state must contain the index of the current hidden vector being estimated $i \in \{1, \dots, M\}$, the entry point $j \in \{1, \dots, |\mathcal{E}_i|\}$, and the amount of time spent observing the current state $k \in \{0, \dots, K_i\}$. The cluster state must also contain the visitation history a $\mathbf{v} \in \{0, 1\}^M$ to prevent rewards from being gained by collecting the same information twice. The cluster state-space is thus

$$\mathcal{S}_C \triangleq \{0, 1\}^M \times \{1, \dots, M\} \times \{1, \dots, \max_i |\mathcal{E}_i|\} \times \{0, \dots, \max_i K_i\}, \quad (10)$$

where the max functions are needed to account for the largest local state spaces in the cluster.

Let $\mathcal{U}_C = \{1, \dots, M\}$ denote the set of control inputs in the cluster DP. Let also $\Phi_C : \mathcal{S}_C \times \mathcal{U}_C \rightarrow \mathcal{S}_C$ denote the cluster transition function. When the robot is in state $(\mathbf{v}, i, j, k) \in \mathcal{S}_C$, then $\mathbf{u}_i \in \mathcal{U}_C$ transitions the robot to cluster state $(\mathbf{v}, i, j, k+1) \in \mathcal{S}_C$. Action \mathbf{u}_i has the same reward with the corresponding transition in the local DP. The remaining actions $\{\mathbf{u}_\ell \mid \ell \neq i\}$ set the visitation history of the i -th hidden vector to zero and transition the robot to the ℓ -th local space, specifically by selecting the closest entry point in \mathcal{E}_ℓ to the robot's current location. In particular,

$$\Phi_C((\mathbf{v}, i, j, k), \mathbf{u}_\ell) = \begin{cases} (\mathbf{v}, i, j, k+1) & \text{if } \ell = i \\ (\mathbf{v}', \ell, j', 0) & \text{if } \ell \neq i \end{cases},$$

where $\mathbf{v}'_i = 0$ and

$$j' = \operatorname{argmin}_{j''} \|\mathbf{s}_{j''} - \Phi_i^{*k}(\mathbf{s}_j)\| \mid \mathbf{s}_{j''} \in \mathcal{E}_\ell, \mathbf{s}_j \in \mathcal{E}_i. \quad (11)$$

The reward for continuing the local optimal policy is the same as the reward in the local DP, and the reward for transitioning to a new hidden state in the cluster is the negative distance to the next entry point j' defined in (11). The only other difference with the local reward is that the robot cannot gain positive reward for taking action \mathbf{u}_i when at any state $(\mathbf{v}, i, \cdot, \cdot)$ such that $\mathbf{v}_i = 0$. In particular, define the reward function in the cluster DP $R_C : \mathcal{S}_C \times \mathcal{U}_C \rightarrow \mathbb{R}$ as

$$R_C((\mathbf{v}, i, j, k), \mathbf{u}_\ell) = \begin{cases} R\left(\Phi_i^{*k}(\mathbf{s}_j), \mu_i^*\left(\Phi_i^{*k}(\mathbf{s}_j)\right)\right) & \text{if } \mathbf{v}_i = 1, \ell = i \\ -\rho\psi(\mathbf{u}_\ell) & \text{else} \end{cases}.$$

Remark IV.2. One could directly apply the method developed in Section III to the task of sensing multiple targets at once. The state-space in this case would be of the form $(\cup_{i=1}^M \mathcal{W}_i) \times \mathcal{C}_1 \times \dots \times \mathcal{C}_M$, which has

$$\left(\sum_{i=1}^M N_{\mathcal{W}_i}\right) \prod_{i=1}^M \underbrace{\left(1 + N_{\mathcal{A}_i} \sum_{\lambda \in \mathcal{L}_i} N_{\mathcal{T}_\lambda}\right)}_{N_{\mathcal{C}_i}} \quad (12)$$

states, an intractably large number of states for any sufficiently rich belief space \mathcal{C}_i . This is the reason why existing approaches have typically employ online policy search or point-based solvers. Our proposed hierarchical approach solves a simple local DP once to find a local optimal policy for every state, mitigating the exponential complexity to the cluster DP, where the state space has size $O(2^M)$. Solving the cluster DP has similar in complexity to TSP solvers although it solves a substantially more complicated problem.

V. OPTIMAL PLANNING AND RESOURCE ALLOCATION FOR MULTIPLE ROBOTS

Given the single-robot optimal control policies developed in Sections III and IV, in this section we develop a distributed framework to synthesize them in a multi-robot system that can efficiently estimate large groups of hidden vectors. To develop the proposed framework, let $\mathcal{T} = \{1, \dots, M\}$ denote the index set of all available hidden states, and for every hidden state $t \in \mathcal{T}$ define the discrete set $\mathcal{Z}^t \triangleq \{\mathbf{z} \in \mathbb{R}^d \mid (\mathbf{z}, \theta) \in \mathcal{W}_t\}$ containing those states of the local workspace \mathcal{W}_t that correspond to robot positions in $d = 2$ or 3 dimensions only (excluding other configuration information contained in θ). Let $\mathcal{Z} = \cup_{t \in \mathcal{T}} \mathcal{Z}^t$ denote the set of all possible robot positions. Consider further a partition $\{\mathcal{T}_p\}_{p=1}^P$ of the hidden state set \mathcal{T} into $P \leq M$ clusters so that two hidden states belong in the same cluster if they are sufficiently close to each other based on the initial belief of their locations. Let $\mathcal{Z}_p \triangleq \{\mathbf{z} \in \mathcal{Z}^t \mid t \in \mathcal{T}_p\}$ denote the set of all robot positions in cluster \mathcal{T}_p . Using the methods developed in Sections III and IV, we can determine an optimal sensing policy for every cluster of hidden states \mathcal{T}_p , that is a function from the state-space of robot positions and hidden state uncertainties to the set of robot actions. Then, given an entry point from where a robot can begin sensing cluster \mathcal{T}_p this policy can be combined with the system dynamics (1b) and (1c) to generate an optimal robot trajectory within that cluster. In particular, we define the set of *entry points* of cluster \mathcal{T}_p to be the set of states in \mathcal{Z}_p lying on the boundary of its convex hull, denoted by $\partial\mathcal{Z}_p$. We also define an optimal trajectory in cluster \mathcal{T}_p starting from a point $\mathbf{p}_{j_p} \in \partial\mathcal{Z}_p$ by $\xi_{p,j_p} : [0, L_{p,j_p}] \rightarrow \mathcal{Z}_p$, where $L_{p,j_p} > 0$ denotes the total length of that trajectory from the entry point until the cluster is completed and j_p is an index of the j -th entry point in $\partial\mathcal{Z}_p$.

Consider now N mobile robots and let $\{\mathbf{z}_i^k\}_{i=1}^N \subset \mathbb{R}^d$ denote their locations at time $k \in \mathbb{N}$. While sensing cluster \mathcal{T}_p a robot moves along the optimal trajectory ξ_{p,j_p} . When the exploration of \mathcal{T}_p has been completed, the robot needs to transition to a different cluster. For this, it needs to identify

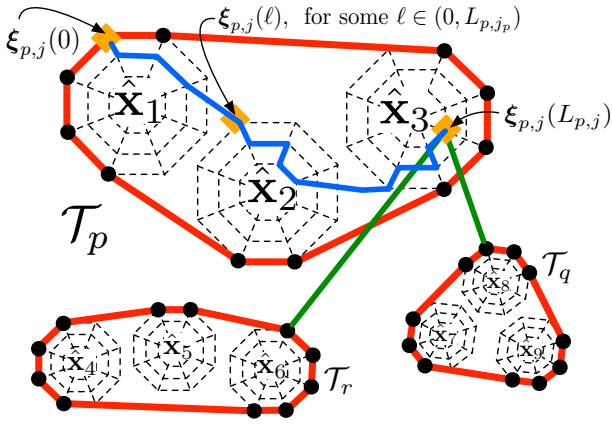


Figure 2. An illustration of two potential sensing paths for a single robot for $d = 2$ in a landmark localization task. The blue segment is the range of $\xi_{p,j}$, where the top left entry point is arbitrarily assigned the index j . The robot's progress at three instants, corresponding to $\ell = 0$, $\ell = L_{p,j}$, and some intermediate value $\ell \in (0, L_{p,j})$, are shown as orange diamonds. The choices of routes to two next clusters \mathcal{T}_q and \mathcal{T}_r are shown as green segments emanating from $\xi_{p,j}(L_{p,j})$.

an entry point in the new cluster and travel to that point. Therefore, the set of all possible paths that a robot can follow while sensing different clusters of hidden states can be represented by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = (\cup_{p=1}^P \partial\mathcal{Z}_p)$ denotes the set of vertices (embedded in \mathbb{R}^d) containing the depot and all entry points of the P hidden state clusters and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of directed edges so that for any entry points $\mathbf{p}_{j_p} \in \partial\mathcal{Z}_{j_p}$ and $\mathbf{p}_{j_q} \in \partial\mathcal{Z}_{j_q}$, the edge $(\mathbf{p}_{j_p}, \mathbf{p}_{j_q}) \in \mathcal{E}$ if $p \neq q$ and $\mathbf{p}_{j_q} = \operatorname{argmin}_{\mathbf{p} \in \partial\mathcal{Z}_q} \|\xi_{p,j_p}(L_{p,j_p}) - \mathbf{p}\|$. In other words, the directed edges in \mathcal{E} connect every entry point in \mathcal{V} to the closest entry points of different clusters. With every edge $(\mathbf{p}_{j_p}, \mathbf{p}_{j_q}) \in \mathcal{E}$ in \mathcal{G} we associate a distance that the robot needs to travel in order to get to entry point $\mathbf{p}_{j_q} \in \partial\mathcal{Z}_{j_q}$ having started from point $\mathbf{p}_{j_p} \in \partial\mathcal{Z}_{j_p}$. This distance consists of the distance L_{p,j_p} that the robot needs to travel within cluster \mathcal{T}_p plus the distance $\|\xi_{p,j_p}(L_{p,j_p}) - \mathbf{p}_{j_q}\|$ that the robot needs to travel to reach entry point $\mathbf{p}_{j_q} \in \partial\mathcal{Z}_{j_q}$ once it has completed cluster \mathcal{T}_p , i.e., $d(\mathbf{p}_{j_p}, \mathbf{p}_{j_q}) = L_{p,j_p} + \|\xi_{p,j_p}(L_{p,j_p}) - \mathbf{p}_{j_q}\|$. We assume that between clusters, robots travel on straight-line paths. An illustration of the graph \mathcal{G} of possible motion paths for the running example of sparse landmark localization in two dimensions, containing the cluster entry points and trajectories to two possible next clusters, is shown in Figure 2. To avoid having two robots select the same cluster and fail to resolve this conflict, we assume that the communication range of the robots is larger than the largest diameter of any cluster.

In what follows, we develop a distributed framework to allow the team of N robots to dynamically allocate the P clusters amongst themselves, as they plan trajectories on the graph \mathcal{G} to visit their assigned hidden state clusters. Specifically, we assume that at every time k , every robot i can be in one of three modes $m_i^k \in \{\text{'busy'}, \text{'transit'}, \text{'done'}\}$. We say that a robot is 'busy' if it is currently sensing a cluster, it is in 'transit' if it is traveling between clusters, and it is 'done' if it has completed sensing its last cluster. While in mode 'busy' robot i moves along the trajectory ξ_{p,j_p} according to

$$\mathbf{z}_i^{k+1} = \text{busy}(\ell^k, p, j_p) \triangleq \xi_{p,j_p}(\ell^k + \delta\ell), \quad (13)$$

where $\ell^k = \xi_{p,j_p}^{-1}(\mathbf{z}_i^k) \in [0, L_{p,j_p}]$ denotes the distance that robot i has already travelled in cluster \mathcal{T}_p and $\delta\ell > 0$ is a small, user-defined, positive distance increment that the robot travels between times k and $k+1$. While in mode 'transit', robot i moves according to

$$\mathbf{z}_i^{k+1} = \text{transit}(\mathbf{z}_i^k, q) \triangleq \mathbf{z}_i^k + \delta\ell \frac{\mathbf{p}_{j_q} - \mathbf{z}_i^k}{\|\mathbf{p}_{j_q} - \mathbf{z}_i^k\|}, \quad (14)$$

where q denotes the index of the cluster that the robot is traveling to, $\mathbf{p}_{j_q} \in \partial\mathcal{Z}_q$ is the selected closest entry point in that cluster defined as $\mathbf{p}_{j_q} = \operatorname{argmin}_{\mathbf{p} \in \partial\mathcal{Z}_q} \|\mathbf{z}_i^k - \mathbf{p}\|$, and $\delta\ell > 0$ is defined as in (13). Assuming that robot i is in transit to cluster \mathcal{T}_q after having completed cluster \mathcal{T}_p , the line segment defined by the end points \mathbf{z}_i^k and \mathbf{p}_{j_q} is completely contained in the line segment defined by the points $\xi_{p,j_p}(L_{p,j_p})$ and \mathbf{p}_{j_q} . Therefore, (14) drives the robot along the path corresponding to the edge $(\mathbf{p}_{j_p}, \mathbf{p}_{j_q}) \in \mathcal{E}$. However, the controller (14) also allows robot i to diverge from the predefined paths in the graph \mathcal{G} by selecting an alternative closest cluster while in transit mode, for reasons that we discuss in Section V-A, e.g., if another robot places a higher bid. In this case, the motion of robot i temporarily leaves the predefined motion paths in \mathcal{G} and it re-enters \mathcal{G} once it has reached cluster \mathcal{T}_q . In the remainder of this section, we will refer to the current cluster being sensed by the i -th robot as $c_{i,\text{curr}}$ and the next cluster to be sensed by that robot as $c_{i,\text{next}}$. Then, our goal is to find a set of N distinct paths in \mathcal{G} whose union visits every cluster exactly once and has a minimum combined travel cost. We achieve this goal by a distributed auction mechanism that we discuss next.

A. Distributed Auction Mechanism

In this section we propose a distributed auction method to dynamically and sequentially allocate clusters to robots as they move to localize the whole scene. A proof of the convergence of distributed auctions can be found in [26].

Specifically, let $s = 1, 2, \dots$ denote a sequence of time instants when the robots communicate with each other, that is in general different from the times k when the robots move, and let $\mathcal{N}_i^s \triangleq \{j \mid \|\mathbf{z}_i^s - \mathbf{z}_j^s\| < \Delta\}$ denote the set of neighbors of robot i at time s , where $\Delta > 0$ denotes a given communication range. Moreover, assume that every robot i carries two lists: the list of 'free' clusters $\mathcal{I}_{i,f}^s$ and the list of 'taken' clusters $\mathcal{I}_{i,t}^s$, so that $\mathcal{I}_{i,f}^s \cup \mathcal{I}_{i,t}^s = \{1, \dots, P\}$ and $\mathcal{I}_{i,f}^s \cap \mathcal{I}_{i,t}^s = \emptyset$ for all time s . Initially, $\mathcal{I}_{i,f}^0 = \{1, \dots, P\}$ and $\mathcal{I}_{i,t}^0 = \emptyset$. The list of 'free' clusters contains clusters that are available to robot i , meaning that robot i can select from those clusters a cluster to visit next. On the other hand, the list of 'taken' clusters contains clusters that have been selected by other robots and are, therefore, not available to robot i . During operation, robot i coordinates with its neighbors $j \in \mathcal{N}_i^s$ to update its list of 'taken' and 'free' clusters by $\mathcal{I}_{i,t}^{s+1} = \cup_{j \in \mathcal{N}_i^s} \mathcal{I}_{j,t}^s$ and $\mathcal{I}_{i,f}^{s+1} = \{1, \dots, P\} \setminus \mathcal{I}_{i,t}^{s+1}$, respectively. In other words, with every communication round, robot i removes from its list of free clusters those clusters that are considered taken by other robots.

Given the list of 'free' clusters $\mathcal{I}_{i,f}^s$ at time s , robot i can select any cluster from that list to be the next cluster $c_{i,\text{next}}$ to

visit. To minimize the total distance travelled by the robots, we propose a greedy approach where robots select a cluster that is the closest to their current location. In particular, we define

$$c_{i,\text{next}}^{s+1} = \begin{cases} \operatorname{argmin}_{c \in \mathcal{I}_{i,f}^s} d_{p,j_p}^{\text{busy}}(\ell^s, c) & \text{if } m_i^s = \text{'busy'} \\ \operatorname{argmin}_{c \in \mathcal{I}_{i,f}^s} d^{\text{trans}}(\mathbf{z}_i^s, c) & \text{if } m_i^s = \text{'transit'} \end{cases} \quad (15)$$

$$d_{p,j_p}^{\text{busy}}(\ell^s, c) \triangleq (L_{p,j_p} - \ell^s) + \min_{\mathbf{p} \in \partial \mathcal{Z}_c} \|\boldsymbol{\xi}_{p,j_p}(L_{p,j_p}) - \mathbf{p}\| \quad (16)$$

$$\text{and } d^{\text{trans}}(\mathbf{z}_i^s, c) \triangleq \min_{\mathbf{p} \in \partial \mathcal{Z}_c} \|\mathbf{z}_i^s - \mathbf{p}\|, \quad (17)$$

In (16) and (17), $d_{p,j_p}^{\text{busy}}(\ell^s, c)$ and $d^{\text{trans}}(\mathbf{z}_i^s, c)$ are the distances that robot i needs to travel in order to reach a new cluster c from its current location \mathbf{z}_i^s while in modes ‘busy’ and ‘transit’, respectively, and $\ell^s = \boldsymbol{\xi}_{p,j_p}^{-1}(\mathbf{z}_i^s) \in [0, L_{p,j_p}]$, as in (13). When robot i selects a new cluster $c_{i,\text{next}}^s$, then it also updates its lists of ‘free’ and ‘taken’ clusters by removing $c_{i,\text{next}}^s$ from $\mathcal{I}_{i,f}^s$ and adding it to $\mathcal{I}_{i,t}^s$.

Every time $c_{i,\text{next}}$ is updated, robot i also places a bid that indicates how important the selection of the new cluster is. The bids are inversely proportional to the distance robot i needs to travel to reach the new cluster, so that nearby clusters have higher value. Specifically, bids are placed according to

$$b_i^{s+1} = \begin{cases} \max_{c \in \mathcal{I}_{i,f}^s} \left(1 + d_{p,j_p}^{\text{busy}}(\ell^s, c)\right)^{-1} & \text{if } m_i^s = \text{'busy'}, \\ \max_{c \in \mathcal{I}_{i,f}^s} \left(1 + d^{\text{trans}}(\mathbf{z}_i^s, c)\right)^{-1} & \text{if } m_i^s = \text{'transit'} \end{cases}. \quad (18)$$

If at some point in time there exist neighbors $j \in \mathcal{N}_i^s$ of robot i so that $c_{j,\text{next}}^s = c_{i,\text{next}}^s$, then these robots set up a local auction and compare their bids to resolve the underlying conflict. If $b_i^s > b_j^s$ for all $j \in \mathcal{N}_i^s$ for which $c_{j,\text{next}}^s = c_{i,\text{next}}^s$, then robot i wins the auction and maintains the same next cluster and bid, i.e., $c_{i,\text{next}}^{s+1} = c_{i,\text{next}}^s$ and $b_i^{s+1} = b_i^s$. The robots that lose the auction update their set of ‘free’ clusters by removing cluster $c_{j,\text{next}}^s$, i.e., $\mathcal{I}_{j,f}^s = \mathcal{I}_{j,f}^s \setminus \{c_{j,\text{next}}^s\}$, and select a new next cluster and bid according to (15) and (18). If $\mathcal{I}_{i,f}^s = \emptyset$, i.e., if there are no other available clusters for robot i , we set $c_{i,\text{next}}^{s+1} = \text{'depot'}$, effectively controlling the robot to return to a depot after it has completed its current (final) task.

Fig 3 illustrates the integrated, hybrid, controller. The labels α and β mark events that need to be synchronized across the navigation and coordination control blocks. In particular, transitions labeled by the letter α are triggered by the navigation block and generate synchronous transitions in the coordination block aimed to produce new bids or update the next cluster that the robot needs to visit. Similarly, transitions labeled by the letter β are triggered by the coordination block when new bids are computed or the next cluster that the robot needs to visit is updated and they generate synchronous transitions in the navigation block aimed to guide the robot its new assigned cluster. Note that while the k and s time indices used for the navigation and coordination blocks, respectively, can in general be different, the transitions labeled by the letters α and β can generate transitions in these blocks that can be off-clock. For example, a transition at time k in the navigation block labeled by α will generate a transition in the coordination block at a time instant $k \neq s$.

VI. NUMERICAL SIMULATIONS

In this section, we present simulations of the proposed distributed state estimation algorithm. In our simulations, we focus on the problem of sparse landmark localization. As a sensor model, we use a stereo camera. We refer the reader to [5], [6] for a discussion of the covariance function $\mathbf{Q}(\hat{\mathbf{x}}_i, \mathbf{p})$ for stereo vision. We assume that each camera in the simulated rig has 1024×1024 resolution and a 70° field of view. The characteristic length in stereo vision is the baseline. Therefore, in these simulations, *all units are measured in stereo baselines unless otherwise stated*. For a mobile stereo rig, the baseline could be as large as 1 meter or as small as a few centimeters.

To simplify the exposition, we consider fully actuated kinematic robots. The local pose spaces and prior error covariances are identical for each landmark, i.e., $\mathcal{W}_i = \mathcal{W}_j$ and $\boldsymbol{\Sigma}_{i,0} = \boldsymbol{\Sigma}_{j,0} = \beta \mathbf{I}_3$ for all $i, j \in \{1, \dots, M\}$, where we recall that the largest eigenvalue in the dimension of the state-space that represents the possible principal eigenvalues is denoted as β .

A. Active sensing of a single target

The local pose space \mathcal{W} for the single target case is made up of concentric spherical shells with randomly distributed viewpoints on each. There were 177 total views in \mathcal{W} at six equally spaced radii between 20 and 40 units from the landmark. To discretize the covariance space \mathcal{C} , we set $1 = \max\{\lambda \in \mathcal{L}\} \triangleq \beta$. In total, the number of possible principal eigenvalues was $N_{\mathcal{L}} = 6$. We also use $N_{\mathcal{A}} = 3$ condition numbers. We set the number of principal eigenvectors for covariances with β as the principal eigenvalue to $N_{\mathcal{T}_\lambda} = 98$. This means \mathcal{T}_β has 98 unit vectors. The number of samples at other principal eigenvectors $\{\lambda \in \mathcal{L} \mid \lambda \neq \beta\}$ was $\left\lceil \frac{\lambda}{\beta} N_{\mathcal{T}_\beta} \right\rceil$. For the logspace discretizations, we set $\kappa_{\mathcal{L}} = 9$, and $\kappa_{\mathcal{A}} = 3$. Stereo vision is used as the sensing model; see Appendix A. The discount factor γ was 0.99. The value of the uncertainty reduction gain ρ is very important, since it controls the tradeoff between traveling cost and uncertainty reduction due to more images. Fig. 4 shows the sensitivity of the total reward gained and the two opposing objectives that comprise it as they depend on ρ . Fig. 5 shows example an optimal trajectory in both pose and covariance space with $\rho = 0.999$.

In these simulations, we approximate error distributions as state-dependent white noise using the equations given in Appendix A. White noise is uncorrelated in time, thus, under standard KF assumptions, two observations from the same vantage point will reduce variance by Lemma III.2. This is why, in the figures, there are more intermediate covariance ellipsoids than steps in the robot trajectories; it is sometimes optimal to remain stationary and take multiple observations. In real scenarios, noise is seldom well-approximated as white, and duplicate observations will exacerbate bias. We account for such hidden biases in these simulations by using a real model of stereo vision that is subject to the nonlinear effects of quantization on the image plane; see Appendix A. Interestingly, the presence of such noise reveals a secondary benefit of sensor mobility: mobile sensors avoid by default duplicate observations that would plague a static sensor. Although our

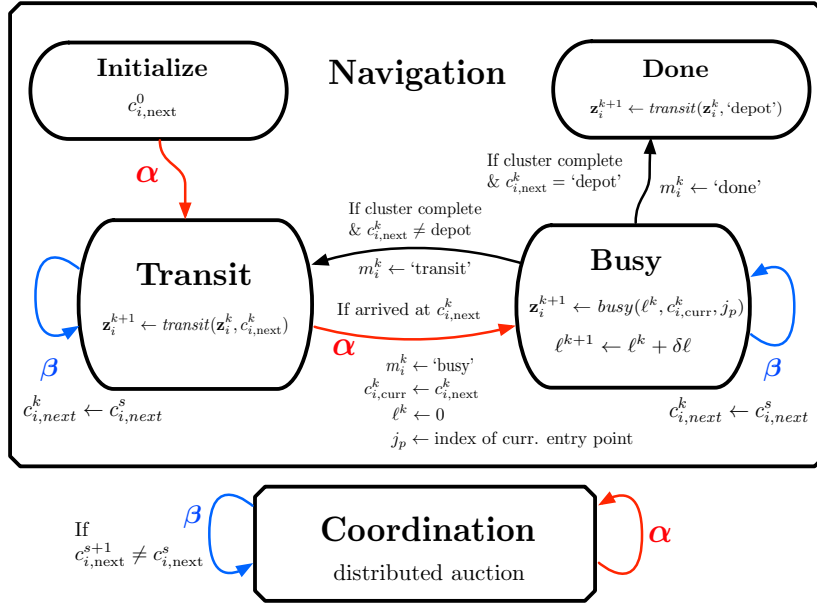


Figure 3. A diagram representing the controller for the i -th robot. The two blocks, Navigation and Coordination, synchronize whenever the updates denoted by α and β are triggered. Functions *transit* and *busy* are defined in (14) and (13).

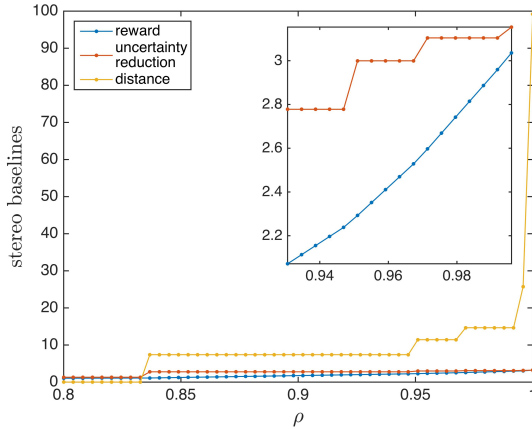


Figure 4. Showing the effect of the uncertainty gain parameter ρ on the total reward, uncertainty reduction, and total distance traveled by the sensor in the single landmark localization simulations. Lines are drawn to guide the eye

Table I
COMPARISON TO HEURISTICS FOR SINGLE LANDMARK

Method	optimal	closer	static	circle	random
Reward (m)	3.15	2.75	1.31	1.66	2.74

controller does not explicitly encourage motion for this reason, a basic heuristic adjustment that does would be simple to implement by, e.g., removing the action “remain stationary.”

Table I compares the total reward gained by a robot starting from the same initial condition under the optimal policy with $\rho = 0.999$ found by our (optimal) local controller with some heuristic policies. The heuristics we chose for comparison were a policy that drives closer to the hidden state (closer), one that circles around the source (circle), one that remains still (static), and one that acts randomly (random). The fact that that ‘random’ performs about as well as ‘closer’ suggests

that a sophisticated controller is needed in order to beat ‘random.’ Our ‘optimal’ policy obtains 115% of the reward of the ‘closer’ and ‘random’ policies. Also note that the ‘static’ and ‘circle’ policies perform very poorly in comparison to the others. This is because they stay at a constant depth with respect to the landmark. In triangulation with stereo vision, there is significant bias in the viewing direction, causing poor localization performance given multiple observations at constant depth.

B. Active sensing of Target Clusters

We present simulations of the proposed distributed estimation method for a single robot observing a cluster of sparse landmarks. For the landmarks, we model a famous group of sculptures called the Queens of France and Famous Women, which can be found in the Luxembourg Garden, Paris, France. The Luxembourg Garden is actually home to hundreds of sculptures, and the Queens of France and Famous Women is a cluster that surrounds a large pool (octagon in Fig. 6) that is adjacent to the Luxembourg Palace (large rectangle in Fig. 6). We chose a subset of eight queens as individual sparse landmarks to comprise a cluster of statues. The task of the robot in this scenario is to exactly localize each statue to create a precise spatial map of the sculpture garden. For these simulations, we used a discount factor and gain of 0.999. We use the same hemispherical pose state-spaces as in Section VI-A for each local DP.

We also compare the trajectories generated by our cluster DP with a state-of-the-art algorithm [2]. The method in [2], called ϵ - δ Reduced Value Iteration (ϵ - δ RVI), grows a tree in belief space that computes an optimal sensor trajectory, i.e., it applies the most widely used approach to the same problem that we solve under similar assumptions. In our implementation of ϵ - δ RVI, planning horizons greater than 12 caused our simulation, run on Macbook AirTM with 4 GB of RAM,

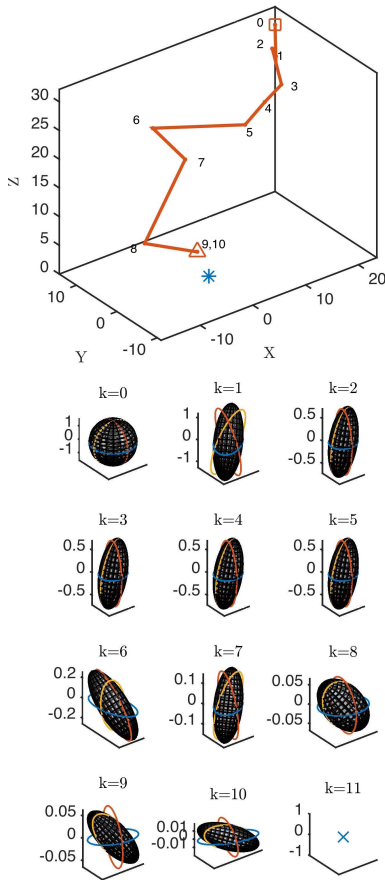


Figure 5. Showing an optimal trajectory through both pose (top panel) and covariance (bottom six panels) space for $\rho = 0.999$. The indices in the top panel correspond to the time indices in the bottom panel. Covariance states are shown as 50% confidence regions. Robot starts at \square and ends at \triangle . All units in stereo baselines and axis directions equally scaled

to run out of memory. Because of the sparsity of the scene, 12 step lookahead was sometimes not enough to plan future landmarks to visit. Therefore, if the robot following ϵ - δ RVVI finishes observing a landmark to the threshold, it greedily selects a new landmark. The top panel in Fig. 6 displays four snapshots of two different trajectories: one produced by the cluster DP optimal policy and one produced using ϵ - δ RVVI. The bottom panel of Fig. 6 shows the result of the KF output of 100 simulations of observations taken along the trajectories in the top panel. The empirical standard deviations for each error vector are also drawn on the figure. Both methods perform similarly, with our method requiring slightly less observations. The important distinction, however, is that for every new initial condition, any tree-based planner, including ϵ - δ RVVI, needs to be run again, whereas we can reuse our optimal policy for any initial condition. This is important in the next section, as it allows allocation of clusters dynamically in the multi-robot team from a variety of initial conditions along the cluster boundaries. In other words, our method allows us to compute an *optimal policy*, rather than a single *optimal trajectory*.

C. Multiple Robots and Multiple Clusters

We present simulations of the proposed distributed estimation framework for multi-robot multi-target active localization

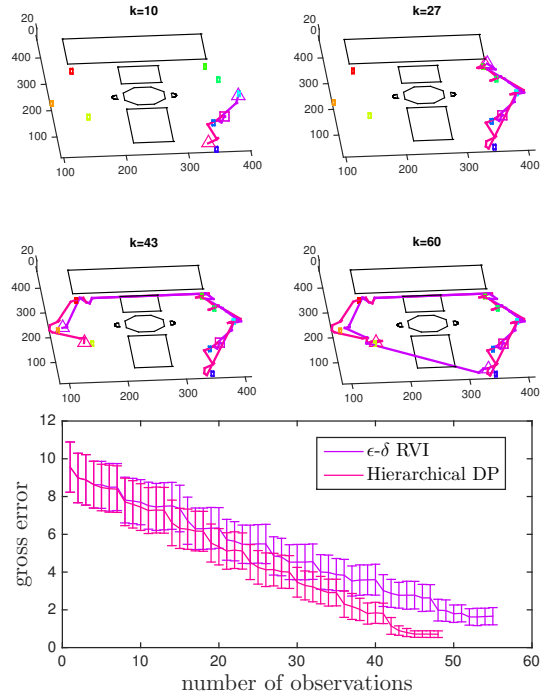


Figure 6. Top panel: Showing trajectories generated by our hierarchical controller and by the method in [2]. The trajectories start at the \square and end at the \triangle . The number k indicates the number of observations taken. Bottom Panel: plotting the sum of errors in all landmarks versus the number of observations required found over 100 simulations of observations taken along the trajectories in the top panel. The midlines represent the empirical mean and error bars represent the standard deviation of the sum of error vectors over the 100 simulations. All units in meters

in Fig. 7. For these simulations, we uniformly randomly generated targets in a rectangle in \mathbb{R}^2 . The communication range of the robots was set to 1500 m. In this simulation, first we divide the targets into two sets of roughly equal size based on the prior location estimates $\{\hat{\mathbf{x}}_i\}_{i=1}^M$. Then, we iteratively split the sets based on relative distance until the largest cluster has less than nine targets. In our testing, we have found that a simple clustering strategy can be effectively generated using the prior over the target locations. We leave the clustering strategy as a design choice in this algorithm that should be made once the location prior is available. For example, in our example in Section VI-A, the sculpture garden can be naturally clustered with *a priori* available tourist maps.

VII. EXPERIMENTS

In this section, we present experiments using a team of two ground robots (iRobot Create), r_1 and r_2 , that localize a set of stationary targets. The robots each carry a stereo rig mounted atop a servo that can rotate the rig $\pm 180^\circ$. Each rig uses Point Grey Flea3TM cameras with resolution 1280×1024 . To simulate long distance localization, all images are downsampled by rate 24 so that the effective resolution is 54×43 . Each robot is equipped with an on-board computer with 8GB RAM, an Intel Core i5-3450S processor and a 802.11n wireless network card for communication. All implementation is done in C++ and run on Robot Operating System (ROS). We calibrate the intrinsic and the extrinsic parameters of the stereo rig offline using the Bouget MatlabTM camera

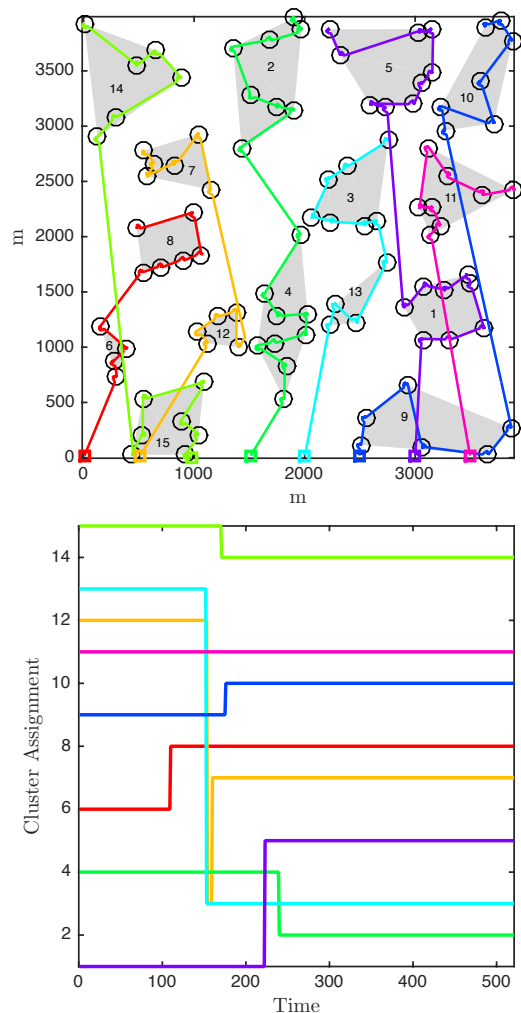


Figure 7. Cooperative active sensing for 100 sparse landmarks and 15 clusters using 8 robots. Top panel: Targets are denoted by circles, and clusters are shaded grey. Squares represent the initial locations of the robots at the bottom of the panel. Bottom panel: cluster assignment result from the online distributed auction for each of the eight robots. Colors correspond to robot trajectories in the top panel. Note that at time 150, the orange robot loses an auction for cluster 3 to the cyan robot, and selects a different cluster 7.

calibration toolbox [27]. For self-localization of the robots, our laboratory is equipped with an OptiTrack™ array of infrared cameras that act as a motion capture system. The OptiTrack™ system tracks reflective markers that are rigidly attached to the robots. Each robot can thus retrieve its own (and only its own) position and orientation by reading a ROS topic that is broadcast over wifi by a centralized computer. Each robot also broadcasts auction bids and states to public ROS topics that are readable by the other robots for collaboration. Additionally, the OptiTrack™ system provides us with the ground truth locations of the targets, which are colored ping pong balls, enabling us to directly check the accuracy of the landmark localization errors after the experiment. The information function \mathbf{Q} was derived using a statistical model of stereo vision, which we defer to Appendix A.

Fig. 8 shows the experimental setup for the multi robot, multi landmark localization experiment. We place eight colored ping pong balls in a square workspace of about 2 m

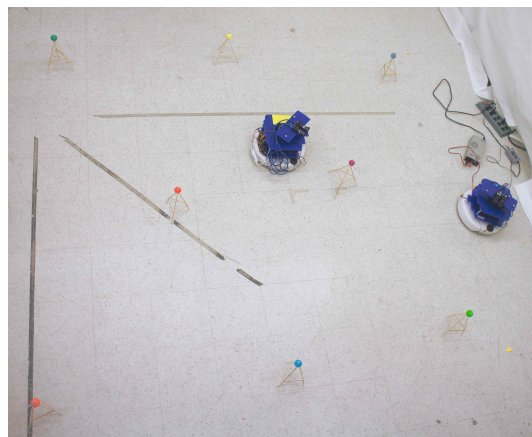


Figure 8. Overhead photograph of the experimental setup

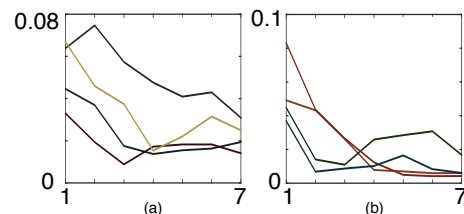


Figure 9. Plotting the filtered target localization error in meters for r1 (a) and r2 (b). The horizontal axis is the number of observations for each. This means that each robot took 28 observations total, localizing four targets each.

side length for the multi robot, multi landmark localization experiment. We use three types of local workspaces: section, semicircle, and full circular polar grids. To avoid collisions between robots and ping pong balls and avoid overlapping local workspaces, the minimum and maximum radii were set to 30 and 50 cm for all local workspaces, respectively. The available poses within each local workspace were located at five equally spaced radii between these two extrema. The polar grids were also divided into increments of 15° . We set the covariance space parameters to $N_{\mathcal{L}} = 10$, $N_{\mathcal{A}} = 3$, $\kappa_{\mathcal{L}} = 10$, and $\kappa_{\mathcal{A}} = 7$. We set $N_{\mathcal{T}}$ to be twice the number of viewing angles in the local workspaces. For these experiments, we set $\rho = 10^{-2}$ and $\gamma = 0.9$.

In ‘transit’ mode, a potential field algorithm guides the robots to the next local workspace and avoids collisions with landmarks. All navigation and waypoint tracking relies on a PID controller using the next waypoint as the set point. In the experiment, robots generally came within 2 cm of their target waypoints. The servo guided the stereo cameras toward the estimate of the target locations with accuracy of $\pm 1^\circ$. Cooperation of r1 and r2 relies on robots updating individual ROS topics and checking neighbors’ ROS topics. The ROS topic for a given robot contains that robot’s current bid value and the subtask at which the bid is directed.

The filtered estimates generated by observations along the paths followed by the robots in one run of the experiments are compared with the ground truth locations of the ping pong balls in Fig. 9. The filtered errors are computed as the Euclidean distances between filtered estimates and the ground truth locations of the ping pong balls. We note that any non-

decreasing aspects of these plots are attributed to stochasticity in the dataset, and the fact that we are only approximating the true noise distribution with a data-driven Gaussian. Note that, due to the relatively constrained space in our lab, the robots are not able to move as freely as in the simulations, which, in addition to unmodeled noise, is responsible for the smaller error reduction in Fig. 9 compared to the simulations.

VIII. CONCLUSION

In this paper, we addressed the task of estimating a finite set of hidden state vectors that have Gaussian priors using a mobile robotic sensor network. We framed the problem using tools from optimal control, ultimately proposing a hierarchical Dynamic Programming solution. By including the covariance matrix in the local state-spaces then discretizing these spaces, we bound the computational complexity for a given error tolerance. This is a desirable property compared to tree-based methods that actively explore the covariance space until the tolerance is reached. Then, we combined the local optimal trajectories in a cluster DP that balances between reducing the uncertainty in the hidden states and traveling among configuration spaces. Our approach is still exponential in the number of hidden states per cluster, but the proposed hierarchical scheme reduces the base of the exponential to two without sacrificing the explicit dependence on the error covariance matrices of the objective function. Then, we proposed distributed auction algorithm to divide the tasks of sensing each cluster among multiple robots. Simulations and experiments on real robots show that the integrated multi-robot system can efficiently localize large groups of landmarks while remaining scalable, a novel pair of characteristics that POMDP and TSP approaches do not have.

REFERENCES

- [1] C. Freundlich, P. Mordohai, and M. M. Zavlanos, "Optimal path planning and resource allocation for active target localization," in *IEEE American Control Conf. (ACC)*, pp. 3088–3093, 2015.
- [2] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Information acquisition with sensing robots: Algorithms and error bounds," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 6447–6454, 2014.
- [3] A.-a. Agha-mohammadi, S. Agarwal, S. Chakravorty, and N. M. Amato, "Simultaneous localization and planning for physical mobile robots via enabling dynamic replanning in belief space," *arXiv preprint arXiv:1510.07380*, 2015.
- [4] T. H. Chung, J. W. Burdick, and R. M. Murray, "A decentralized motion coordination strategy for dynamic target tracking," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 2416–2422, 2006.
- [5] C. Freundlich, P. Mordohai, and M. M. Zavlanos, "Hybrid control for mobile target localization with stereo vision," in *IEEE Conf. on Decision and Control (CDC)*, pp. 2635–2640, 2013.
- [6] C. Freundlich, P. Mordohai, and M. M. Zavlanos, "A hybrid control approach to the next-best-view problem using stereo vision," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 4478–4483, 2013.
- [7] J. Vander Hook, P. Tokekar, and V. Isler, "Algorithms for cooperative active localization of static targets with mobile bearing sensors under communication constraints," *IEEE Trans. on Robotics*, vol. 31, no. 4, pp. 864–876, 2015.
- [8] H. Carrillo, Y. Latif, M. L. Rodriguez-Arevalo, J. Neira, and J. A. Castellanos, "On the monotonicity of optimality criteria during exploration in active slam," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1476–1483, 2015.
- [9] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: theory and application to multi-robot slam," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 4775–4782, 2015.

- [10] A. Ryan and J. K. Hedrick, "Particle filter based information-theoretic active sensing," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 574–584, 2010.
- [11] N. Adurthi and P. Singla, "Information driven optimal sensor control for efficient target localization and tracking," in *IEEE American Control Conf. (ACC)*, pp. 610–615, 2014.
- [12] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *Int. J. Robotics Research*, pp. 1–17, 2014.
- [13] J. Le Ny and G. J. Pappas, "On trajectory optimization for active sensing in gaussian process models," in *IEEE Conf. on Decision and Control (CDC)*, pp. 6286–6292, 2009.
- [14] N. Atanasov, B. Sankaran, J. Le Ny, G. J. Pappas, and K. Daniilidis, "Nonmyopic view planning for active object classification and pose estimation," *IEEE Trans. on Robotics*, vol. 30, no. 5, pp. 1078–1090, 2014.
- [15] M. T. Spaan, T. S. Veiga, and P. U. Lima, "Decision-theoretic planning under uncertainty with information rewards for active cooperative perception," *Int. Found. for Autonomous Agents and Multiagent Sys. (IFAAMAS)*, pp. 1–29, 2014.
- [16] A.-a. Agha-mohammadi, S. Chakravorty, and N. Amato, "FIRM: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements," *Int. J. Robotics Research*, vol. 33, pp. 268–304, February 2014.
- [17] H. Kurniawati and V. Yadav, "An online POMDP solver for uncertainty planning in dynamic environment," in *Robotics Research*, pp. 611–629, Springer, 2016.
- [18] K. M. Seiler, H. Kurniawati, and S. P. Singh, "An online and approximate solver for POMDPs with continuous action space," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.
- [19] H. Bai, D. Hsu, and W. S. Lee, "Integrated perception and planning in the continuous space: A POMDP approach," *Int. J. Robotics Research*, vol. 33, no. 9, pp. 1288–1302, 2014.
- [20] S. Omidshafiei, A.-a. Agha-mohammadi, C. Amato, and J. P. How, "Decentralized control of partially observable markov decision processes using belief space macro-actions," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.
- [21] F. A. Oliehoek and C. Amato, "The decentralized POMDP framework," in *A Concise Introduction to Decentralized POMDPs*, pp. 11–32, Springer, 2016.
- [22] H. Kurniawati, D. Hsu, and W. S. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Robotics: Science and Sys. (RSS)*, Zurich, Switzerland, 2008.
- [23] N. Atanasov, M. Zhu, K. Daniilidis, and G. J. Pappas, "Localization from semantic observations via the matrix permanent," *Int. J. Robotics Research*, vol. 35, no. 1-3, pp. 73–99, 2016.
- [24] D. Hardin and E. Saff, "Discretizing manifolds via minimum energy points," *Notices of the AMS*, vol. 51, no. 10, pp. 1186–1194, 2004.
- [25] E. Carlen, "Trace inequalities and quantum entropy: an introductory course," *Entropy and the quantum*, vol. 529, pp. 73–140, 2010.
- [26] M. M. Zavlanos, L. Spesivtsev, and G. J. Pappas, "A distributed auction algorithm for the assignment problem," in *IEEE Conf. on Decision and Control (CDC)*, pp. 1212–1217, 2008.
- [27] J.-Y. Bouguet, "Camera calibration toolbox for Matlab," 2004.
- [28] C. Freundlich, P. Mordohai, and M. M. Zavlanos, "Exact bias correction and covariance estimation for stereo vision," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 3296–3304, 2015.
- [29] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An invitation to 3-D vision: from images to geometric models*, vol. 26. Springer Science & Business Media, 2012.
- [30] L. H. Matthies and S. A. Shafer, "Error modelling in stereo navigation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 3, pp. 239–250, 1987.
- [31] C. Freundlich, Y. Zhang, A. Z. Zhu, P. Mordohai, and M. M. Zavlanos, "Hybrid control for active target localization with mobile stereo vision," *Int. J. Robotics Research*, 2016 (under review).

APPENDIX A NOISE MODEL

In this paper, we assume that individual measurements are subject to a known zero mean Normal noise distribution $\nu \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$. In what follows, we estimate \mathbf{Q} so that this assumption holds for the stereo rigs used in our laboratory experiments. This is critical for a variety of reasons:

- If the mean of ν is biased, then the KF will not converge to the ground truth.
- If \mathbf{Q} is an under approximation to the actual covariance of ν , then the KF will become inconsistent and will not converge to the ground truth, if it converges at all.
- If our choice of \mathbf{Q} is too conservative, it may not be informative enough to be useful in the decision process at the core of the controller.

Making things even more difficult, we want to test the system in relatively extreme conditions, particularly at long ranges, when triangulation error distributions are known to be heavy tailed, biased away from zero, and highly asymmetric [28].

In our experiments we make use of the physics of stereo vision [29]. In particular, following [30] we assume that pixel error are Gaussian, and we propagate them to the localization estimates via triangulation equations, which we can use to give us an accurate distribution for ν . Specifically, if the robot registers a correspondence in the left (L) and right (R) cameras at pixel coordinates $[x_L, x_R, y]^\top$, then the coordinates of the 3-D point that generated the match are

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{b}{d} \begin{bmatrix} \frac{1}{2}(x_L + x_R) \\ y \\ f \end{bmatrix}, \quad (19)$$

where f is the focal distance, b is the stereo baseline, $d = x_L - x_R$ is the disparity. Let \mathbf{J} denote the Jacobian of (19). If the error in $[x_L, x_R, y]^\top$ has covariance matrix \mathbf{P} , then the error covariance of $[X, Y, Z]^\top$ is $\mathbf{J}\mathbf{P}\mathbf{J}^\top$. Moreover, if \mathbf{t} and \mathbf{T} are the translation vector and rotation matrix from the coordinate frame of the camera to a fixed coordinate frame, then the covariance of ν is $\mathbf{T}\mathbf{J}\mathbf{P}\mathbf{J}^\top\mathbf{T}^\top$.

With this noise propagation formula in mind, we now study errors in the observed pixels. For this we follow a data-driven approach that we have recently developed in [31]. Using a set of $n = 600$ pairs of training images for each robot at various ranges and viewing angles, we obtain a regression that maps pixel observations $[x_L, x_R, y]^\top$ to a corrected tuple of pixels $[x_L^c, x_R^c, y^c]^\top$ such that the error in the corrected tuple is zero mean. For every image pair, we project the ground truth landmark onto the image planes using the inverse mapping of (19), giving us $\ell = 1, \dots, n$ individual output vectors \mathbf{Y}_ℓ , which we stack into an $n \times 3$ matrix of outputs \mathbf{Y} . The ground truth data for this training set includes a marker placed on top of the ping pong ball, and the pose information of the stereo rig, captured by a \mathbf{t} and \mathbf{T} . We then compute five features and, because the data are not centered, include one constant, for each raw pixel tuple according to the model

$$\mathbf{X}_\ell = \left[1, y_\ell, d_\ell, x_{L,\ell} + x_{R,\ell}, y d_\ell, \frac{x_{L,\ell} + x_{R,\ell}}{d_\ell} \right]. \quad (20)$$

These features are taken from the constituent terms in the nonlinear equation (19). Stacking the \mathbf{X}_ℓ into an $n \times 6$ matrix, we have a linear model $\mathbf{Y} = \mathbf{X}\beta + \epsilon$, where β is a 6×3 matrix of coefficients and ϵ is an $n \times 3$ matrix of zero-mean Normal errors. We experimentally verified that the rows of ϵ are roughly Gaussian, as can be seen in the right panel of Fig. 10. We refer to the raw pixels as *uncorrected*. The associated error vectors (computed with respect to the uncorrected pixels and

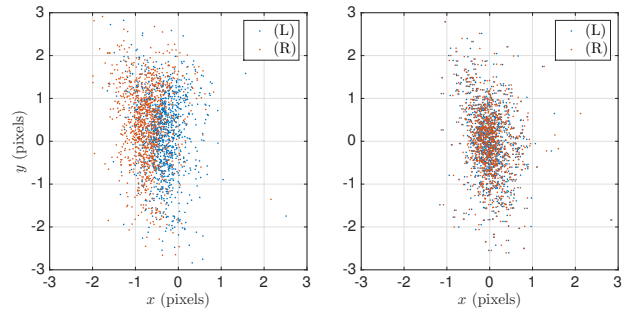


Figure 10. Scatter plots of the residual errors $\epsilon_\ell^{\text{uc}}$ (left panel) and ϵ_ℓ (right panel) for the training data for r1. The plots for r2 are similar.

the projected ground truth) $\epsilon_\ell^{\text{uc}}$ for $\ell = 1, \dots, n$ are plotted in the left panel of Fig 10. In the scatter plot it can be seen that the mean error is nonzero, contributing average bias to individual measurements. Also note the apparent skew of the error distribution in the vertical (y) direction.

Applying the ordinary least squares estimator, the maximum likelihood estimate of the coefficient matrix is $\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$. Using $\hat{\beta}$, the residual covariance in the pixel measurements for the two robots, named r1 and r2, we obtained are

$$\mathbf{P}_{\text{r1}} = \begin{bmatrix} 0.13 & 0.09 & 0.02 \\ 0.09 & 0.13 & -0.03 \\ 0.02 & -0.03 & 0.28 \end{bmatrix}, \quad \mathbf{P}_{\text{r2}} = \begin{bmatrix} 0.22 & 0.16 & 0.04 \\ 0.16 & 0.23 & 0.03 \\ 0.04 & 0.03 & 0.74 \end{bmatrix}.$$

Note that the standard deviation of the y pixel value, corresponding to the variances in the lower right entries of the above matrices, is 0.53 and 0.86 pixels, respectively. This corresponds to errors in the height of the ping pong ball center in vertical coordinates. The right panel of Fig. 10 shows the residual errors in the training set ϵ_ℓ for $\ell = 1, \dots, n$ for the corrected vector $\mathbf{X}\hat{\beta}$ on r1.

For prediction, if r1 makes a new observation (x_L^*, x_R^*, y^*) , it forms a 1×6 vector \mathbf{X}^* . Then, r1 calculates the corrected pixels $[x_L^c, x_R^c, y^c]^\top = \mathbf{X}^* \hat{\beta}$, which are subject to zero mean Normal errors. Using the corrected pixels, r1 triangulates the relative location of the target via (19), propagates \mathbf{P}_{r1} via the Jacobian \mathbf{J} , rotates and translates the estimates to global coordinates, and thus the assumptions that the error terms ν are zero mean with covariance $\mathbf{Q} = \mathbf{T}\mathbf{J}\mathbf{P}_{\text{r1}}\mathbf{J}^\top\mathbf{T}^\top$ are approximately satisfied. The case for r2 is analogous.

Finally, note that the regression takes place in three dimensions, whereas the algorithm is designed only for two dimensions. For planning purposes, we consider only the components of \mathbf{Q} that lie on the plane of the workspace, i.e., we do not use the third row and column for planning. Of course, the experiments take place in three dimensions, so we need to use the full covariance matrices in the Kalman Filters. Note that the algorithm proposed in this paper could be implemented in 3D at the expense of increasing the pose and covariance spaces accordingly.