Hybrid coverage and inspection control
for anisotropic mobile sensor teams

Antonio Adaldo∗ Dimos V. Dimarogonas∗ Karl H. Johansson∗

1. INTRODUCTION

A wide variety of applications involve collecting information in hazardous environments, which makes it desirable to delegate such missions to a team of autonomous agents with sensing capabilities. Therefore, in the last few decades, a lot of research interest has been devoted to the problem of autonomous deployment of robot teams in an assigned space—see Cortés et al. (2004); Martinez et al. (2007); Zhong and Cassandras (2011); Le Ny and Pappas (2013) to name just a few. Typically, the goal is to design a distributed algorithm that gradually drives the agents to a spatial configuration such that the team’s collective perception of the environment is optimized. This problem is commonly known as the coverage problem, and it is often approached using Voronoi tessellations and the Lloyd algorithm (see Du et al. (1999)). The majority of the existing work on coverage considers agents with omnidirectional perception of the surrounding environment. Recently, agents with anisotropic sensing patterns (see Stergiopoulos and Tzes (2013); Gusraildi et al. (2008); Laventall and Cortes (2008)) as well as vision-based sensing patterns (see Kantaros et al. (2015); Marier et al. (2012)) have been considered. Dynamic versions of the coverage problem have also been studied, where the agents do not converge to fixed positions, but keep navigating the environment in order to maintain a satisfactory coverage over time. This problem is commonly known as effective or dynamic coverage (see Hussein and Stipanovic (2007)). A vision-based version of effective coverage is studied in Panagou et al. (2015).

Information exchange among the sensing agents constitutes one of the major challenges in the real-world implementation of coverage algorithms. For this reason, a gossip-based communication strategy for coverage is studied in Durham et al. (2012) among others. In some coverage missions, it is convenient to abstract the environment into a finite set of points (see Durham et al. (2012)), which may either correspond to a sparse set of points of interest, or to a discretized approximation of the environment itself.

Our contribution with respect to the related work is twofold. On one hand, we generalize the notions of Voronoi cell and Voronoi tessellation, thus extending the use of this formalism to coverage problems with general, anisotropic sensing patterns. On the other hand, we address the coverage problem with a distributed hybrid control algorithm, where system flow is given by the motion of the sensing agents, while the system jumps are given by the communication instances between different agents.

Using the hybrid system formalism developed in Goebel et al. (2012), we formally show that the agents reach an equilibrium configuration, while a measure of the coverage attained by the team improves monotonically. The effectiveness of the proposed algorithm is then demonstrated by simulating a team of sensors in ROS (Quigley et al. (2009)).

2. PRELIMINARIES

We let \( \mathbb{N} \) denote the set of the nonnegative integers, and \( \mathbb{R}_{\geq 0} \) denote the set of the nonnegative real numbers. For \( V : \mathbb{R}^n \to \mathbb{R} \), with \( n \in \mathbb{N} \), \( \nabla V(\cdot) \) denotes the gradient of \( V \). All vectors in \( \mathbb{R}^n \) (including gradients) are column vectors. We let \( 0_n \) (with \( n \in \mathbb{N} \)) be the vector of \( \mathbb{R}^n \) whose entries are all zero. The set of the unity-norm vectors in \( \mathbb{R}^3 \) is also called the unit sphere, and is denoted...
\( \mathcal{S}^2 \). The special orthogonal group in three dimensions is denoted \( \text{SO}(3) \). Each element of \( \text{SO}(3) \) is represented as a rotation matrix, and each column of such matrix belongs to \( \mathcal{S}^2 \); namely, if \( R \in \text{SO}(3) \), then \( R = [x \, y \, z] \), with \( x, y, z \in \mathcal{S}^2 \). The special Euclidean group in three dimensions is denoted as \( \text{SE}(3) \). Each element of \( \text{SE}(3) \) is represented as a homogeneous transformation matrix; namely \( T = \begin{bmatrix} R & p \\ 0_3 & 1 \end{bmatrix} \), where \( R \in \text{SO}(3) \), and \( p \in \mathbb{R}^3 \).

3.2. The coverage of a finite set of landmarks \( L = \{\ell_1, \ldots, \ell_M\} \), attained by a sensor \( s \) is defined as

\[
\text{per}(\sigma, \ell) = f_s(T_{sT_1}^{-1} T_1).
\]

We adopt the convention that a smaller value of the perception corresponds to a better perception, with \( \text{per}(\sigma, \ell) = 0 \) being the best possible perception. Some sensing models used in the literature can be seen as particular cases of this model, as it is illustrated in the following examples.

**Example 3.1.** (Temperature sensors). Typically, a temperature sensor can measure the temperature of the surrounding environment with a degree of accuracy that degrades with the distance from the sensor itself. Therefore, a reasonable footprint for a temperature sensor is

\[
\text{per}(T_{sT_1}^{-1} T_1) = \|p\|^2,
\]

where \( p = T_{sT_1}^{-1} T_1 \) is the position of the sensor and \( p = T_{sT_1} \) is the position where we want to measure the temperature. Most of the existing research on coverage focuses on this particular sensing pattern—see for example Cortés et al. (2004).

**Example 3.2.** (Inspection with aerial robots). Consider the task of inspecting a wind turbine with a team of aerial robots endowed with cameras. The perception that a vehicle has of a point on the surface of the turbine depends on the distance between the vehicle and the point, but also on the direction that the camera is pointing to; perception is best if the point lies on the line of sight of the camera, and it is worse if the point approaches the boundaries of the camera’s cone of view. This type of perception can be described by the following footprint:

\[
f_s(T) = k_1 \|p - \beta e_1\|^2 + k_2 \|\beta e_1 - p\| |(\beta e_1 - p) | e_1, \]

where \( 0 < k_1 < k_2, p = T_{sT_1} \), \( e_1 = [1, 0, 0]^\top \), and \( \beta > 0 \) is the optimal distance between the camera and the observed point. Figure 1 illustrates a contour plot of footprint (4).

Next, we define the coverage of a set of landmarks attained by a sensor as the sum of the perceptions of the landmarks. This concept is formalized in the following definition.

**Definition 3.2.** The coverage of a finite set of landmarks \( \mathcal{L} = \{\ell_1, \ldots, \ell_M\} \), attained by a sensor \( \sigma \) is defined as

\[
\text{coverage}(\mathcal{L} | \sigma) = \sum_{\ell \in \mathcal{L}} \text{per}(\sigma, \ell).
\]
In this paper, our aim is to use a team of sensors in order to optimize the coverage of an environment. Therefore, we shall now formally define the coverage of a set of landmarks attained by a team of sensors.

Definition 3.3. Consider a team of sensors $\mathcal{S} = \{s_1, \ldots, s_N\}$ and a set of landmarks $\mathcal{L} = \{1, \ldots, M\}$. Let $\mathcal{L}$ be partitioned in $N$ subsets $\mathcal{L}_1, \ldots, \mathcal{L}_N$. Each sensor $s_i$ in the team is assigned the landmarks $\mathcal{L}_i$, thus defining a partition $\mathcal{P} = \{\mathcal{L}_1, \ldots, \mathcal{L}_N\}$. Then, we define the coverage of the set $\mathcal{L}$ attained by the sensor team $\mathcal{S}$ with the partition $\mathcal{P}$ as

$$\text{Cov}(\mathcal{S}, \mathcal{P}) := \sum_{i=1}^{N} \text{Cov}(s_i, \mathcal{L}_i).$$

In other words, the coverage attained by the team is simply the sum of the coverages attained by the single sensors, each over its own subset of landmarks.

Our objective is to control the poses of the sensors and the partition of the landmarks in such a way that the team coverage improves over time (i.e., Cov($\mathcal{S}, \mathcal{P}$) decreases with time). This objective can be formalized as an optimization problem. To this aim, introduce the binary variables $\sigma_{1,N,M}$ such that $\sigma_{i,j} = 1$ if $\ell_j \in \mathcal{L}_i$ (i.e., if the sensor $s_i$ owns the landmark $\ell_j$), and $\sigma_{i,j} = 0$ otherwise. Imposing that $\sigma_{1,1} + \cdots + \sigma_{1,N} = 1$ ensures that each landmark is owned by exactly one sensor. Then, the optimization of the team coverage can be formulated as the following problem:

$$\begin{aligned}
\text{minimize} & \quad \sum_{i=1}^{N} \sum_{j=1}^{M} \sigma_{i,j} f_i(T_i^{-1}T_{\ell_j}), \\
\text{subject to} & \quad \sigma_{i,1} + \cdots + \sigma_{i,N} = 1, \ i \in \{1, \ldots, N\},
\end{aligned}$$

where $f_i$ denotes the footprint of sensor $s_i$, and $T_{\ell_j}$ denotes the pose of landmark $\ell_j$. Using (2), (5) and (6), we can see that the cost function in Problem (7) is indeed equal to the team coverage Cov($\mathcal{S}, \mathcal{P}$). Finding a globally optimal solution of Problem (7) may require computationally demanding optimization algorithms, especially because of the implicit constraint $T_1, \ldots, T_N \in \text{SE}(3)$ (which essentially corresponds to the orthonormality of the vectors that describe the orientation of each sensor). In the rest of this paper, we describe a distributed algorithm to find a locally optimal solution of Problem (7). In the proposed algorithm, most of the computation is performed locally and asynchronously by each individual sensor; communication among the sensors is only intermittent and pairwise. The algorithm can be performed online (i.e., while the sensor team is performing a coverage mission), and guarantees convergence to an equilibrium configuration, with a monotonically decreasing (i.e., improving) value of the team coverage.

4. SINGLE-SENSOR CONTROL FOR COVERAGE IMPROVEMENT

The first step in defining our coverage algorithm is to define how we control the motion of each individual sensor to improve the coverage attained by that sensor. Let $T_i(t) \in \text{SE}(3)$ be the pose of sensor $s_i$ at time $t$, let $v_i$ be its footprint, and let $v_i(t), \omega_i(t) \in \mathbb{R}^3$ be respectively its linear velocity and angular velocity. From the laws of the kinematics of rigid bodies, we have that the pose of the sensor evolves according to the differential equation

$$\dot{T}_i(t) = \begin{bmatrix} S(\omega_i(t))R_i(t) v_i(t) \\ 0 \end{bmatrix},$$

where $R_i = T_i(1:3,1:3)$. We know that the coverage of a set of landmarks $\mathcal{L}_i$ attained by $s_i$ is given by (5), which, using (2), can be rewritten as

$$\text{cov}(s_i(t), \mathcal{L}_i) = \sum_{\ell \in \mathcal{L}_i} f_i(T_i(t)^{-1}T_{\ell}).$$

Taking the time derivative of both sides of (9), and applying the chain rule for the derivative of a scalar with respect to a matrix (see Petersen and Pedersen (2012)), we have

$$\frac{d}{dt} \text{cov}(s_i(t), \mathcal{L}_i) = \sum_{\ell \in \mathcal{L}_i} \text{Tr}\left( \frac{\partial f_i(T_i(t)^{-1}T_{\ell})}{\partial T_{\ell}^T} \dot{T}_i(t) \right).$$

Substituting (8) into (10), expanding the trace, and using the property $S(w)w = -S(w)w \forall w \in \mathbb{R}^3$, we have

$$\frac{d}{dt} \text{cov}(s_i(t), \mathcal{L}_i) = \sum_{\ell \in \mathcal{L}_i} \sum_{\xi \in \{x_i(y_i), y_i(z_i)\}} \frac{\partial f_i(T_i(t)^{-1}T_{\ell})}{\partial x_i} v_i(t)$$

$$- \sum_{\ell \in \mathcal{L}_i, \xi \in \{x_i(y_i), y_i(z_i)\}} \frac{\partial f_i(T_i(t)^{-1}T_{\ell})}{\partial \xi} S(\xi(t)) \omega_i(t),$$

where $[x_i, y_i, z_i] = T_i(1:3,1:3)$. From (11), we can see immediately that cov($s_i(t), \mathcal{L}_i$) can be made nonincreasing by choosing

$$v_i(t) = -\sum_{\ell \in \mathcal{L}_i} \frac{\partial f_i(T_i(t)^{-1}T_{\ell})}{\partial x_i},$$

$$\omega_i(t) = -\sum_{\ell \in \mathcal{L}_i, \xi \in \{x_i(y_i), y_i(z_i)\}} \frac{\partial f_i(T_i(t)^{-1}T_{\ell})}{\partial \xi} S(\xi).$$

5. MULTI-AGENT COVERAGE ALGORITHM

Before we start illustrating the proposed multi-agent coverage algorithm, we characterize the class of locally optimal solutions of Problem (7) that constitute our control objective. A feasible solution of Problem (7) is defined by specifying the pose and the landmarks of each sensor, with the constraint that the each landmark must belong exactly to one sensor. We are looking for a feasible solution such
that the value of the team coverage cannot be improved by only moving one sensor in a neighborhood of its position (without changing the ownership of any landmark), or by only changing the ownership of some landmarks (without moving any sensor). We call this class of feasible solution Voronoi solutions, because of their analogy with Voronoi tessellations (see Du et al. (1999)). The formal definition of a Voronoi solution is given next.

**Definition 5.1.** A feasible solution \((T_1, \ldots, T_N, \sigma_{1,1}, \ldots, \sigma_{N,M})\) of Problem (7) is called a Voronoi solution if it satisfies the following properties for all \(i \in \{1, \ldots, N\}\):

\[
\sum_{\ell \in \mathcal{L}_i} \frac{\partial f_i(T_i^{-1}T_\ell)}{\partial p_{i,k}} = 0; \quad (14)
\]

\[
\sum_{\ell \in \mathcal{L}_i} \sum_{\xi \in \{s_1, y_1, z_1\}} S(\xi) \frac{\partial f_i(T_i^{-1}T_\ell)}{\partial \xi} = 0; \quad (15)
\]

\[
\sigma_{i,j} = 1 \implies \text{per}((T_i, f_i, \ell_j) \leq \text{per}((T_i, f_i, \ell_j)). \quad (16)
\]

Obviously, any globally optimal solution of Problem (7) is a Voronoi solution, but there may exist Voronoi solutions that are not globally optimal. Our goal is to define a distributed algorithm that drives the sensors to a Voronoi solution.

We are now ready to present the proposed multi-agent coverage algorithm. The input to the algorithm is a team of sensors \(s_1, \ldots, s_N\) in their initial poses \(T_1^0, \ldots, T_N^0\), and a set of landmarks \(\mathcal{L} = \{\ell_1, \ldots, \ell_M\}\). The landmarks constitute an abstract representation of an object or an area of interest that we want to keep under observation by using our sensor team.

The algorithm is initialized by assigning each landmark to one of the sensors. This initial assignment must not be considered a centralized operation: in fact, it can be done at random or by assigning all the landmarks to the same sensor. The pose of each sensor is continuously controlled by (12) and (13), while occasionally a sensor interacts with another sensor to transfer the ownership of some landmarks. A landmark is transferred if the recipient sensor has a better perception of that landmark than the original owner. Following these rules, moving a sensor or transferring a landmark always produces an improvement in the team coverage \(\text{Cov}(\mathcal{D}, \mathcal{P})\). The time instants when a sensor \(s_i\) contacts another sensor to give it some landmarks are denoted as \(t_{i,k}\), with \(k \in \mathbb{N}\). These time instants may be periodic, event-triggered, or triggered by user requests. We only require that there exist a minimum and a maximum dwell time between two consecutive communication instances; namely,

\[
\exists \tau_{\min}, \tau_{\max} > 0: \tau_{\min} \leq t_{i,k+1} - t_{i,k} \leq \tau_{\max}. \quad (17)
\]

For the sake of generality, consider the threshold functions

\[
\varsigma_i(\tau_i, T_i, \mathcal{L}_i) \in \mathbb{R}_{\geq 0}, \quad i \in \{1, \ldots, N\},
\]

where \(\tau_i\) is a clock, and let the communication times \(t_{i,k}\) be triggered by the condition \(\varsigma_i(\tau_i, T_i, \mathcal{L}_i) \geq 0\). Note that this formalism encompasses both time-triggered and event-triggered communication. If \(s_i\) contacts another sensor \(s_j\) but cannot give it any of its landmarks (i.e., if \(\text{per}(s_j, \ell) \geq \text{per}(s_i, \ell) \text{ for all } \ell \in \mathcal{L}_i\)), then \(s_i\) contacts another sensor until it manages to transfer at least one landmark, or it has contacted all the other sensors in the team. The described program can be formalized as the following algorithm.

**Algorithm 5.1.** (executed by each sensor \(s_i\) at each \(t \geq 0\)).

1. compute \(v_i\) by (12).
2. compute \(\omega_i\) by (13).
3. update pose \(T_i\) as by (8).
4. update the clock as by \(\dot{\tau}_i = 1\).
5. if \(\varsigma_i(\tau_i, T_i, \mathcal{L}_i) = 0\) then
   - \(\mathcal{L}_i := \{\mathcal{D}\} \setminus s_i\)
   - while \(\mathcal{L}_i \neq \emptyset\) do
     - choose another sensor \(s_j\) from \(\mathcal{L}_i\)
     - if \(\text{per}(s_j, \ell) < \text{per}(s_i, \ell)\) then
       - transfer \(\ell\) to \(s_j\)
     - break while
     - else remove \(s_j\) from \(\mathcal{L}_i\)
   - end if
   - end while
6. end if

**6. MAIN RESULT**

Our main result is to show that Algorithm 5.1 drives the sensors to a Voronoi solution of Problem (7). To make this result into a formal statement, first we model the sensor team and their landmarks as a hybrid system according to the formalism defined in Section 2.

The state of the system is given by the values of the clocks, the poses of the sensors, and the ownership of the landmarks. The ownership of the landmarks is captured by the variables \(\sigma_{1,1}, \ldots, \sigma_{N,M}\), introduced in Section 3. Hence, we let

\[
X = (\tau_1, \ldots, \tau_N, \text{vec}(T_1)^\top, \ldots, \text{vec}(T_N)^\top, \sigma_{1,1}, \ldots, \sigma_{N,M})^\top \in \mathbb{R}^{N+16N+N^2M},
\]

where \(\text{vec}()\) denotes the vectorized form of a matrix. Since \(X\) is the state of a hybrid system, it is a function of both the time elapsed and the number of jumps occurred, and we should write \(X(t, k)\); the same applies to all the state variables. However, we omit these dependencies whenever possible to avoid clutter in the notation.

The state flow is given by the movement of the sensors, described by (8), (12), and (13). State flow can happen from any state; therefore, we set \(\mathcal{E} = \mathbb{R}^{N+16N+N^2M}\). The flow map captures the clocks’ progression and the dynamics of the sensors’ motion. Namely, we set \(F(X) = \{\phi(X)\}\), with \(\phi(X) = (1, \ldots, 1, \text{vec}(T_1)^\top, \ldots, \text{vec}(T_N)^\top, 0_{N^2M}^\top)^\top\), where \(T_i\) denotes here the closed-loop derivative of \(T_i\), obtained by using (12) and (13) in (8).

The state jumps are given by the landmark transfers, and they are triggered by the conditions \(\varsigma_i(\tau_i, T_i, \mathcal{L}_i) = 0\). Hence, state jumps are only possible from states such that \(\varsigma_i(\tau_i, T_i, \mathcal{L}_i) = 0\) for some sensor. Therefore, we let

\[
\mathcal{D} = \{X \in \mathcal{E} : \max_{i=1, \ldots, N} \varsigma_i(\tau_i, T_i, \mathcal{L}_i) = 0\}.
\]

The jump map captures the rules by which the sensors transfer landmarks to each other. Let \(g_{i,j}(X)\) be the new state after \(s_i\) has transferred landmarks to \(s_j\). Then, we have

\[
g_{i,j}(X) = (\tau_1^+, \ldots, \tau_N^+, 0_{16N}^\top, \sigma_{1,1}^+, \ldots, \sigma_{N,M}^+).
\]
where: \( \tau_i^+ = 0 \) and \( \tau_q^+ = \tau_q \) for all \( q \neq i; \sigma_i^+ = 0 \)
and \( \sigma_q^+ = 1 \) if \( \sigma_{i,j} = 1 \) and \( \text{per}(s_i, \ell_j) < \text{per}(s_i, \ell_j); \sigma_{\xi^+} = \sigma_{\xi^-} \).
Let \( g_i(X) \) be the state after \( s_i \) has failed to transfer landmarks to all the other sensors. Then, we have
\[
g_i(X) = (\tau_{0N}^+, \ldots, \tau_{0N}^+, 0_{16N}^+, \sigma_{1,1}, \ldots, \sigma_{N,M}).
\] (22)
with \( \tau_q^+ = 0 \) and \( \tau_q^+ = \tau_q \) for all \( q \neq i. \) With this notation, we can write the jump map as
\[
G(X) = \begin{cases} 
\{g_i(X) : g_i(X) \neq X\} & \text{if } g_i(X) \neq 0, \\
\{g_i(X)\} & \text{otherwise}.
\end{cases}
\] (23)
Our hybrid system is now defined as \( \mathcal{H} = (\mathcal{C}, F, \mathcal{D}, G). \)

Now we need to define Voronoi solutions in terms of \( \mathcal{H}; \) in other words, we have to recast (14), (15), (16) in terms of \( X. \) To this aim, first note (using (8), (12) and (13)) that (14) and (15) are equivalent to
\[
\phi(X) = (1, \ldots, 1, 0_{16N}^+, 0_{16N}^+)^T.
\] (24)
Then, note that (16) means that it is not possible to transfer any landmark, which, using (23), can also be written as
\[
G(X) = \{g_i(X)\}.
\] (25)
Therefore, the set of the states \( X \) corresponding to the Voronoi solutions of Problem (7) is
\[
\mathcal{V} := \{X \in \mathbb{R}^{16N+NM+N} : X \in \mathcal{C} \implies \phi(X) = (1, \ldots, 1, 0_{16N}^+, 0_{16N}^+) \},
\] (26)
where \( \mathcal{C} \) and \( \mathcal{D} \) are the corresponding hybrid system defined by (19)–(23).

Our main result can now be formalized as the following theorem.

**Theorem 6.1.** (Main theorem). Consider a team of sensors \( \mathcal{S} = \{s_1, \ldots, s_N\} \), with initial poses \( T_1^0, \ldots, T_N^0 \in \text{SE}(3) \), and a set of landmarks \( \mathcal{L} = \{\ell_1, \ldots, \ell_M\}. \) Let \( \mathcal{L}_i \) be the subsets of the landmarks owned by sensor \( s_i. \) Let \( \mathcal{L}_i^0 \) be the set of the landmarks initially assigned to sensor \( s_i, \) so that \( \{\mathcal{L}_1^0, \ldots, \mathcal{L}_N^0\} \) is a partition of \( \mathcal{L}. \) If each sensor executes Algorithm 5.1, and the threshold functions \( \varsigma_i \) are such that (17) is satisfied, then each complete solution of \( \mathcal{H} \) converges to the set \( \mathcal{V} \) of the Voronoi solutions of Problem (7).

**Proof.** Consider the function \( \tilde{\text{Cov}}(X) := \text{Cov}(\mathcal{S}, \mathcal{P}). \) We are going to show that this function is nonincreasing both along the system flow and upon the system jumps, so that we can then apply Lemma 2.1.

Let us first study the evolution of this function along the system flow. Differentiating \( \tilde{\text{Cov}}(X) \) with respect to \( X, \) we have
\[
\nabla \tilde{\text{Cov}}(X) = \begin{pmatrix}
0, \\
\text{vec}
\left(\frac{\partial \text{cov}(s_1, \mathcal{L}_1)}{\partial T_1}\right)^T, \\
\vdots, \\
\text{vec}
\left(\frac{\partial \text{cov}(s_N, \mathcal{L}_N)}{\partial T_N}\right)^T, 0_{16N}^+, 0_{16N}^+ \end{pmatrix}^T.
\] (27)
Trasposing both sides, and right-multiplying by \( \phi(X), \) we have
\[
\nabla \tilde{\text{Cov}}(X)^T \phi(X) = \sum_{i=1}^N \text{vec}
\left(\frac{\partial \text{cov}(s_i, \mathcal{L}_i)}{\partial T_i}\right)^T \text{vec}(T_i).
\] (28)
Using (5) and (10) in (28), we can rewrite the right-hand side of (28) as
\[
\nabla \tilde{\text{Cov}}(X)^T \phi(X) = \sum_{i=1}^N \frac{d}{dt} \text{cov}(s_i, \mathcal{L}_i) \leq 0.
\] (29)
Let us now study the evolution of \( \tilde{\text{Cov}}(X) \) upon the system jumps. Consider the generic jump time \( t_{ik}. \) If \( s_i \) transfers some landmarks to another sensor \( s_j, \) then the change in the value of \( \text{Cov}(\cdot) \) is
\[
\tilde{\text{Cov}}(g_i'(s_j)) - \tilde{\text{Cov}}(X) = \sum_{\{t \in \mathcal{F} : \text{per}(s_i, \ell_j) < \text{per}(s_i, \ell_j)\}} \text{per}(s_i, \ell_i) - \text{per}(s_i, \ell_i) < 0.
\] (30)
If it is not possible to transfer landmarks to any other sensor, then \( G(X) = \{g_i(X)\}. \) Since \( g_i(X) \) has the same values of the \( \sigma_{i,j} \) variables as \( X, \) we have
\[
\tilde{\text{Cov}}(g_i(X)) - \tilde{\text{Cov}}(X) = 0.
\] (31)
which means that the value of \( \tilde{\text{Cov}}(\cdot) \) does not change upon this jump. Since (28), (30) and (31) hold, we can invoke Lemma 2.1 to conclude that the largest weakly invariant set \( \mathcal{F} \) contained in \( \mathcal{F} := \{X \in \mathcal{C} : \nabla \tilde{\text{Cov}}(X)^T \phi(X) = 0\} \) is empty, and any complete and bounded solution converges to it. Moreover, recalling that sensor footprints are radially unbounded, while \( \text{Cov}(\cdot) \) is nonincreasing, we can conclude that all solutions of \( \mathcal{H} \) are bounded.

Therefore, any complete solution of \( \mathcal{H} \) converges to \( \mathcal{F}, \) and we can conclude the proof by showing that \( \mathcal{F} = \mathcal{V}. \) Obviously, \( \mathcal{V} \) is invariant, and it is a subset of \( \mathcal{F}, \) so \( \mathcal{V} \subseteq \mathcal{F}. \) Suppose by contradiction that \( \mathcal{V} \subseteq \mathcal{F}. \) Let \( \xi = \arg \max_{X \in \mathcal{F} \setminus \mathcal{V}} \tilde{\text{Cov}}(X). \) Since \( \mathcal{F} \) is weakly invariant, for any \( \tau > 0 \) there exists at least one maximal solution \( \Xi(t, j) \) such that \( \Xi(t^*, j^*) = \xi \) for some \( t^* + j^* > \tau. \) Since \( \text{Cov}(\Xi) \) is nondecreasing, it must be \( \text{Cov}(\Xi(t, j)) = \xi \) for all \( (t, j) \leq (t^*, j^*). \) However, since (17) holds, if we choose \( \tau \) large enough, the solution \( \Xi \) must comprise both flow and jumps between (0, 0) and \( (t^*, j^*). \) Hence, the whole solution must lie in the subset of \( \mathcal{F} \) such that \( \text{Cov}(\Xi) \) does not change with either flow or jumps. From (26), we can see that this set is \( \mathcal{V}. \) But this is a contradiction, because we have supposed that \( \xi \notin \mathcal{F} \setminus \mathcal{V}. \) Hence, we can conclude that \( \mathcal{F} = \mathcal{V}. \)

**7. SIMULATION**

In this section, we present a simulation of Algorithm 5.1 built upon the ROS middleware, with each sensor being implemented as a single ROS node.

We consider a team of \( N = 4 \) sensors, which are required to monitor a square room with side of length \( 6. \) The room is abstracted into a set of \( M = 625 \) landmarks, which are equally spaced about the room. Initially, all the landmarks are assigned to the same sensor. The initial positions of sensor \( s_i \) is chosen as \( p_i = (0, 1.25 - 0.25j, 0), \) while the initial orientation is chosen as \( R_i = I_3 \) for all the sensors. The threshold functions are chosen as
\[
\varsigma_i(t_i, T_i, \mathcal{L}_i) = \min\{t_i - t_d, \epsilon - \|v_i\|, \epsilon - \|\omega_i\|\},
\] (32)
where $\epsilon = 2 \cdot 10^{-2}$, and $v_i$ and $\omega_i$ are computed as (12) and (13) respectively. It is easy to verify that the thresholds (32) satisfy (17).

The footprint of the first two sensors is chosen as (4), with $k_1 = 0.6$ and $k_2 = 0.4$, while $\beta = 1$ for the first sensor and $\beta = 2$ for the second sensor. The footprint of the other two sensors is chosen as (3). This case corresponds to the scenario where some agents have normal cameras with different focal distance, while some other agents have omnidirectional cameras. The results of the simulation are shown in Figure 2, where we can see how the sensors are progressively deployed about the room, and how the landmark distribution assumes patterns corresponding to better coverage costs. Note that the final landmark distribution pattern reflects the shape of the different sensor footprints.

8. CONCLUSIONS AND FUTURE DEVELOPMENTS

We have presented a distributed algorithm for coverage and inspection tasks with multi-agent sensor teams. The proposed algorithm is based on the abstraction of the environment into a finite set of landmarks, and can handle sensors with anisotropic and heterogeneous sensing patterns. The sensor team is modeled as a hybrid multi-agent system, and the algorithm is formally shown to drive the agents to an equilibrium configuration, while a global cost function representing the coverage attained by the team is nonincreasing. Future work will extend the proposed algorithm to explicitly account for practical issues such as collision avoidance.

REFERENCES


Fig. 2. Results of the simulation. Seconds elapsed in each snapshot, in lexicographical order: 0, 0.2, 1.0, 2.0, ... Control and Data Collection with Mobile Sensor Networks. IEEE Transactions on Automatic Control, 56(10), 2445–2455.