Asynchronous Distributed Information Leader Selection in Robotic Swarms

Wenhao Luo*, Shehzaman S. Khatib*, Sasanka Nagavalli*, *Student Member, IEEE*,
Nilanjan Chakraborty†, *Member, IEEE*, Katia Sycara*, *Fellow, IEEE*

***Abstract*— This paper presents asynchronous distributed algorithms for information leader selection in multi-robot systems based on local communication between each robot and its direct neighbours in the system's communication graph. In particular, the information leaders refer to a small subset of robots that are near the boundary of the swarm and suffice to characterize the swarm boundary information. The leader selection problem is formulated as finding a core set that can be used to compute the Minimum-Volume Enclosing Ellipsoid (MVEE) representing the swarm boundary. Our algorithms extract this core set in a fully distributed manner and select core set members as information leaders, thus extending abstract centralized MVEE core set algorithms for robotic swarm applications. We consider different communication conditions (e.g. dynamic network topology) and system configurations (e.g. anonymous robots or uniquely identified robots) and present a variety of approaches for core set selection with associated proofs for convergence. Results for simulated swarms of 50 robots and experiments with a swarm of 10 TurtleBots are provided to evaluate the effectiveness of the proposed algorithms.**

I. Introduction

Distributed multi-robot systems where local interactions between individual robots and their neighbors result in collective behaviors (e.g. foraging, rendezvous, flocking) are known as robotic swarms. Robotic swarms hold great potential for automation in various applications including search and rescue , environmental monitoring (e.g. water monitoring, flood response) and environmental cleanup (e.g. oil spills).

For effective human supervisory control of robot swarms, the swarm's state must be perceptually accessible to the human [1]. However, communication bandwidth constraints may make infeasible communicating the positions of all swarm robots to the human supervisor [2]. Previous experiments showed that enhanced displays [3] enable the human to effectively control the swarm with a summary representation (e.g. convex hull [4]). Thus, effective human control may be possible with position information of a few appropriately selected swarm members. The goal of this paper is to present methods for the swarm to automatically select a set of robots, which we call *information leaders*, whose local state information enables effective construction of a summary representation of the complete swarm state presented to a remotely located supervisor.

Swarm properties of interest are the number of connected (communication graph) components and spatial dispersion of robots in each component. We use *Minimum-Volume Enclosing Ellipsoids* (MVEE) to represent spatial dispersion with one MVEE for each swarm component. MVEE computation for point sets is a well-studied problem in computational geometry with a variety of algorithms proposed to solve the problem. However, these algorithms are centralized, whereas in our problem there is no centralized process that knows the position information of all the robots. Therefore, the key contribution of this paper is distributed algorithms for computing the information leaders under a variety of conditions (e.g. dynamic network topologies, anonymous or uniquely identified robots) employing different underlying methods for MVEE core set selection as needed. Further, we present the results of applying our algorithms in simulated swarms of 50 robots and experiments with 10 TurtleBots.

II. Related Work

In applications of robotic swarms, ellipsoids are often used to represent the swarm shape [5], [6]. Using the MVEE to characterize the swarm shape provides a concise description of the swarm boundary, which motivates selecting boundary robots as information leaders summarizing relevant information for the entire swarm. There is significant prior work on computing the MVEE — known as the Löwner-John ellipsoid [7], [8] — for a set of points. For two dimensional point sets, [9] gives a fast algorithm (time complexity is linear in number of points). A more general approximate approach was presented by Khachiyan in [10], which computes the MVEE for higher dimensional point sets. Modifications to the algorithm can be found in [11],[12] and [13], which extract a subset of points known as the core set to represent the original group of data points and return approximately the same MVEE as [10]. These core set approaches are inspired by [8], which shows that the MVEE in d-dimensional space is supported by a subset of points with size at most $\frac{d(d+3)}{2}$ [14]. Although these algorithms work well in solving high dimensional MVEE problems, the requirement for global information (i.e. all data points) during core set computation makes it challenging to implement them directly on distributed robotic systems.

In contrast to previous work focusing on the abstract problem of finding the MVEE for point sets, we develop practical asynchronous distributed frameworks that employ and extend

This work was supported in part by ONR Grant N0001409-10680 and an NSERC PGS D scholarship.

*The authors are with the Robotics Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. Email: {wenhaol, skhatib, snagaval}@andrew.cmu.edu, katia@cs.cmu.edu. †The authors are with the Department of Mechanical Engineering, Stony Brook University, Stony Brook, New York, USA. Email: nilanjan.chakraborty@stonybrook.edu.

existing MVEEE core set algorithms [12], [11] and apply them to concise boundary representation for robotic swarms. Moreover, by using our MVEE core set formulation instead of all the boundary points employed in [15], the number of effective boundary robots can be effectively reduced to the same size as core sets in [12], [11] and made independent of the size of the swarm.

III. Problem Statement

Consider a robotic swarm consisting of N robots, where each robot knows only its own position $\mathbf{q}_i \in \mathbb{R}^d$ with $d \in \{2,3\}$ in a common reference frame. The set of robot spatial positions is given by $\mathcal{S} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N\}$. Each robot can communicate directly with a subset of the other robots (i.e. neighbours). The communication graph is given by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each node $v \in \mathcal{V}$ represents a robot. If robot $v_i \in \mathcal{V}$ can communicate with robot $v_j \in \mathcal{V}$, then edge $(v_i, v_j) \in \mathcal{E}$. We assume that the communication graph is undirected (i.e. $(v_i, v_j) \in \mathcal{E} \Rightarrow (v_j, v_i) \in \mathcal{E}$), but we do not assume the graph is connected (i.e. it may have multiple components).

For simplicity of exposition, assume the graph consists of only one connected component. We demonstrate later that our algorithms do not require this assumption and have the capability to implicitly handle communication graphs with multiple components. The MVEE for this group of robots is denoted by $\mathrm{MVEE}(\mathcal{S})$, and the corresponding full-dimensional ellipsoid $E_{\mathbf{A},\mathbf{x}_0}$ in $\mathbb{R}^d$ is defined by a symmetric positive definite matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ and its center $\mathbf{x}_0 \in \mathbb{R}^d$. The problem of finding $\mathrm{MVEE}(S)$ can be written as follows:

$$\begin{aligned} \underset{E_{\mathbf{A},\mathbf{x}_0}}{\arg\min} \quad & \det\left(\mathbf{A}^{-1}\right) \\ \text{subject to} \quad & \forall \mathbf{q}_i \in \mathcal{S} : (\mathbf{q}_i - \mathbf{x}_0)^T \mathbf{A} (\mathbf{q}_i - \mathbf{x}_0) \leq 1 \\ & \mathbf{A} \succeq 0 \end{aligned} \tag{1}$$

IV. Minimum-Volume Enclosing Ellipsoids using Core Sets

Before describing the distributed leader selection algorithms, we first look at identifying a core set to generate a $(1+\epsilon)$-approximation to the MVEE as in Khachiyan's algorithm [10] in a centralized manner. A core set is a small subset of points (robots) $X \subseteq \mathcal{S}$ in the entire data set (swarm) $\mathcal{S}$ such that using this subset gives the approximately same MVEE as the one obtained from considering the entire data set. In this paper we mainly discuss two basic core sets: the set of convex hull vertices for $\mathcal{S}$ denoted by *CH core set* $\mathcal{X}_{\text{CH}}$ [11], [16] and the *KY core set* denoted by $\mathcal{X}_{\text{KY}}$ [12]. The convex hull of the robotic swarm is defined as follows.

$$\mathrm{Conv}(\mathcal{S}) = \left\{ \sum_{i=1}^{N} a_i \mathbf{q}_i \mid \forall a_i \geq 0 : \sum_{i=1}^{N} a_i = 1 \right\} \tag{2}$$

In [8], it has been proven that since the convex hull contains all the boundary information for a given point set and $\mathrm{Conv}(\mathcal{S}) \subseteq \mathrm{MVEE}(\mathcal{S})$, then the resulting $\mathrm{MVEE}(\mathcal{X}_{\text{CH}})$ from CH core set $\mathcal{X}_{\text{CH}}$ must also cover all the robots in $\mathcal{S}$, which satisfies the core set definition.

In [12], a modification of Khachiyan's algorithm [10] is proposed with reduced computational complexity that uses a smaller KY core set $\mathcal{X}_{\text{KY}} \subseteq \mathcal{S}$ to incrementally generate a $(1+\epsilon)$-approximation to the $\mathrm{MVEE}(\mathcal{S})$. In this case, the ellipsoid $E_{\mathbf{A},\mathbf{x}_0} \supseteq S$ satisfies $\mathrm{vol}E_{\mathbf{A},\mathbf{x}_0} \leq \mathrm{vol}(1+\epsilon)\,\mathrm{MVEE}(\mathcal{S})$. It follows from (1) that

$$(1+\epsilon)E_{\mathbf{A},\mathbf{x}_0} = \{\mathbf{q}_i \in \mathbb{R}^d : (\mathbf{q}_i - \mathbf{x}_0)^T \mathbf{A}(\mathbf{q}_i - \mathbf{x}_0) \leq (1+\epsilon)^2\} \tag{3}$$

The existence of the core set $\mathcal{X}_{\text{KY}} \subseteq \mathcal{S}$ has been proven in [12] with the property that its size is bounded by $|\mathcal{X}_{\text{KY}}| = O(d\log d + d[(1+\epsilon)^{2/d+1} - 1]^{-1})$.

In [12] the KY core set algorithm works in two stages. The first stage is the initialization of KY core set using [17] which leads to a rough approximation to the volume of $\mathrm{Conv}(\mathcal{S})$ as well as the byproduct of the initial KY core set $\mathcal{X}_{\text{KY}} \leftarrow \mathcal{X}_{\text{BB}}$, where $\mathcal{X}_{\text{BB}} \subseteq \mathcal{S}$ is the union of the vertex of the 2D bounding box of $\mathcal{S}$ with the size at most $|\mathcal{X}_{\text{BB}}| = 2d$. The second stage is the iterative update of KY core set, in which the algorithm goes into a loop to incrementally expand the MVEE from $\mathrm{MVEE}(\mathcal{X}_{\text{BB}})$ to $\mathrm{MVEE}(\mathcal{S})$, by keeping adding violator points $\mathbf{q}_j$ to $\mathcal{X}_{\text{BB}}$ such that $\mathcal{X}_{\text{KY}} \leftarrow \mathcal{X}_{\text{KY}} \cup \{\mathbf{q}_j\}$, until the resulting new expanding ellipsoid $\mathrm{MVEE}(\mathcal{X}_{\text{KY}}) \supseteq \mathrm{MVEE}(\mathcal{S})$. In [12], such a process has been proven to compute the MVEE in a reduced computational complexity of $O(Nd^2(\log d + [(1+\epsilon)^{2/d+1} - 1]^{-1}))$ arithmetic operations, making it advantageous for large-scale cases with $N \gg d$ and reasonably small values of ϵ.

Compared to the CH core set, we prove that for the same swarm $\mathcal{S}$ the KY core set is guaranteed to be a subset of the convex hull vertices as in Figure 1c.

Theorem 1: For the robotic swarm $\mathcal{S} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N\}$, with CH core set $\mathcal{X}_{\text{CH}} \subseteq \mathcal{S}$ and KY core set $\mathcal{X}_{\text{KY}} \subseteq \mathcal{S}$, we have $\mathcal{X}_{\text{KY}} \subseteq \mathcal{X}_{\text{CH}}$.

Proof: Based on the preceding discussion, the KY core set consists of two sets of points: (a) extreme points in $\mathcal{S}$ along the directions of the orthogonal basis vectors for $\mathbb{R}^d$, which are vertices of the corresponding bounding box and (b) points in $\mathcal{S}$ that are furthest from the expanding ellipsoids $E_{\mathbf{A},\mathbf{x}_0}$ in each iteration. By definition of the extreme points, it is straightforward that part (a) of KY core set is a subset of CH core set. For part (b), it follows from (3) that the furthest distance with respect to the current expanding ellipsoid is defined by $\max_{i=1\ldots N}\left((\mathbf{q}_i - \mathbf{x}_0)^T \mathbf{A}(\mathbf{q}_i - \mathbf{x}_0)\right)$, which can be regarded as the affine transformation that preserves points and ratios of distances between points on a straight line. Hence the convex hull and the extreme points (CH core set) in $\mathcal{S}$ are preserved after the transformation, and the furthest points (extreme points) found by (3) must also belong to the vertices of the convex hull, which completes the proof. ■

V. Distributed Information Leader Selection using Core Sets

A. Distributed Leader Selection with Uniquely Identified Robots

Assume that robots have unique identifiers (UIDs) and robots can identify each other by their UIDs. For simplicity of exposition, assume UID(v_i)= i. The UIDs of communication graph neighbors of robot v_i are denoted by $\mathcal{N}_i = \{j \mid v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}\}$. Ideally, we want to solve (3) in a distributed manner by using only local information between connected robots.

1) CH-KY Approach: In this approach, we compute the $\mathcal{X}_{\text{CH}}$ for the swarm $\mathcal{S}$ in a distributed manner and only the component leader robot (with lowest UID) further extracts the KY core set $\mathcal{X}_{\text{KY}}$ from the obtained $\mathcal{X}_{\text{CH}}$, which is called the CH-KY core set $\mathcal{X}_{\text{CH-KY}}$. Each individual robot $v_i \in \mathcal{V}$ maintains a vertex set $\mathcal{X}_i$ such that it lies within the convex hull $\text{Conv}\,(\mathcal{X}_i)$ where initially $\mathcal{X}_i$ just describes the geometry of the robot. Then given the convex hulls $\text{Conv}\,(\mathcal{X}_i)$ and $\text{Conv}\,(\mathcal{X}_j)$ for robot v_i and v_j respectively, the convex hull containing both robots is given by $\text{Conv}\,(\mathcal{X}_i \cup \mathcal{X}_j) = \text{Conv}\,(\text{Conv}\,(\mathcal{X}_i) \cup \text{Conv}\,(\mathcal{X}_j))$.

$$\text{Conv}\left(\bigcup_{\forall v_i \in \mathcal{V}} \mathcal{X}_i\right) = \text{Conv}\left(\bigcup_{\forall v_i \in \mathcal{V}} \text{Conv}\,(\mathcal{X}_i)\right) \quad (4)$$

We want to find the convex hull containing all the robots using only local communication between each robot v_i and its direct neighbours in the communication graph. The UIDs of the neighbors of robot v_i are denoted by $\mathcal{N}_i = \{j | v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}\}$. Most of the convex hull algorithms take a finite set of points $\mathcal{X}$ as input and return the vertices $\mathcal{H} \subseteq \mathcal{X}$ of the convex hull as output.

$$\mathcal{H} = \{\mathbf{x} \in \mathcal{X} \mid \mathbf{x} \notin \text{Conv}\,(\mathcal{X} \setminus \{\mathbf{x}\})\} \quad (5)$$

In Algorithm 1, we define the function $\mathcal{H} = \textsc{ConvexHull}(\mathcal{X})$ that applies one of any known convex hull algorithms to compute the convex hull of the robot swarm $\mathcal{S}$ and use a distributed scheme similar to the one in [18] to achieve decentralized computation. For this and the rest parts, $\textsc{CoreSet}()$ is defined as the KY algorithm in [12] to compute the KY core set for the input point sets. Every robot executes asynchronous distributed Algorithm 1 to ensure that the robot with the lowest UID in each connected component has the correct CH-KY core set for the entire connected component. The algorithm inputs are robot's own UID u, convex hull of its body $\mathcal{H}_u$ and set of its neighbour UIDs $\mathcal{N}_u$. On line 2, it initializes its estimate of the CH core set $\mathcal{H}_\mathcal{S}$, CH-KY core set $\mathcal{C}_\mathcal{S}$, leader UID l, number of hops h from leader and master UID m. On lines 3–5, it sends a message to each neighbour $i \in \mathcal{N}_u$ to initiate the algorithm. Then the algorithm interleaves two different tasks: (1) implicitly establish a spanning tree (lines 8–12) and (2) propagate CH-KY information from leaves of the tree to the root of the tree (lines 14–18). The algorithm terminates when every robot in the connected component has the correct leader UID l, hop ID h and master UID m (master is parent in the spanning tree) and has finished propagating all messages from each robot to the leader through their masters. The termination of the algorithm is implicit. Only the component leader knows the final CH-KY core set $\mathcal{X}_{\text{CH-KY}} = \mathcal{C}_\mathcal{S}$ for the swarm and communicates this to the human operator periodically. The total number of inter-robot messages transmitted during execution is $O\,(|\mathcal{V}| + |\mathcal{E}|)$ – the same as a breadth-first search through the communication graph. Since each message includes an estimate of the convex hull, inter-robot message size is $O\,(|\mathcal{V}|)$.

2) KY Approach: Convex hulls have the beneficial property given in (4) where the convex hull of the union of two CH core sets will contain the points enclosed by both individually. Unfortunately, KY core sets don't have the same property. Namely, given KY core sets $\text{MVEE}\,(\mathcal{A}_{\text{KY}}) \subseteq \text{MVEE}\,(\mathcal{A})$ and $\text{MVEE}\,(\mathcal{B}_{\text{KY}}) \subseteq \text{MVEE}\,(\mathcal{B})$, we note that $\text{MVEE}\,(\mathcal{A} \cup \mathcal{B}) \neq \text{MVEE}\,(\mathcal{A}_{\text{KY}} \cup \mathcal{B}_{\text{KY}})$. Algorithm 2 accounts for this difference by maintaining a hypothesis core set that is updated by parents and verified by descendants in the spanning tree. Every robot executes asynchronous

Algorithm 1 Distributed CH-KY Core Set Selection

1: **procedure** $\textsc{DistributedCHKY}(u, \mathcal{H}_u, \mathcal{N}_u)$
2: $\quad\mathcal{H}_\mathcal{S} \leftarrow \mathcal{H}_u$, $\mathcal{C}_\mathcal{S} \leftarrow \textsc{CoreSet}(\mathcal{H}_u)$, $l \leftarrow u$, $h \leftarrow 0$, $m \leftarrow$ NIL
3: $\quad$**for all** $i \in \mathcal{N}_u$ **do**
4: $\qquad\textsc{SendMsg}(i, u, h, l, \mathcal{H}_\mathcal{S})$
5: $\quad$**end for**
6: $\quad$**while** $\{n, h', l', \mathcal{H}_{\mathcal{S}'}\} \leftarrow \textsc{RecvMsg}()$ **do**
7: $\qquad$**if** $(l > l') \vee ((l = l') \wedge (h > h' + 1))$ **then**
8: $\qquad\quad l \leftarrow l'$, $h \leftarrow h' + 1$, $m \leftarrow n$
9: $\qquad\quad \mathcal{H}_\mathcal{S} \leftarrow \mathcal{H}_u$, $\mathcal{C}_\mathcal{S} \leftarrow \textsc{CoreSet}(\mathcal{H}_u)$
10: $\qquad\quad$**for all** $i \in \mathcal{N}_u$ **do**
11: $\qquad\qquad\textsc{SendMsg}(i, u, h, l, \mathcal{H}_\mathcal{S})$
12: $\qquad\quad$**end for**
13: $\qquad$**else if** $(l = l') \wedge (h < h')$ **then**
14: $\qquad\quad \mathcal{H}_\mathcal{S} \leftarrow \textsc{ConvexHull}(\mathcal{H}_\mathcal{S} \cup \mathcal{H}_{\mathcal{S}'})$
15: $\qquad\quad$**if** $m \neq$ NIL **then**
16: $\qquad\qquad\textsc{SendMsg}(m, u, h, l, \mathcal{H}_\mathcal{S})$
17: $\qquad\quad$**else**
18: $\qquad\qquad\mathcal{C}_\mathcal{S} \leftarrow \textsc{CoreSet}(\mathcal{H}_\mathcal{S})$
19: $\qquad\quad$**end if**
20: $\qquad$**end if**
21: $\quad$**end while**
22: **end procedure**

Algorithm 2 Distributed KY Core Set Selection

procedure $\textsc{DistributedKY}(u, \mathcal{C}_u, \mathcal{N}_u)$
$\quad\mathcal{C}_\mathcal{S} \leftarrow \mathcal{C}_u$, $l \leftarrow u$, $h \leftarrow 0$, $m \leftarrow$ NIL
$\quad$**for all** $i \in \mathcal{N}_u$ **do**
$\qquad\textsc{SendMsg}(i, u, h, l, \mathcal{C}_\mathcal{S})$
$\quad$**end for**
$\quad$**while** $\{n, h', l', \mathcal{C}_{\mathcal{S}'}\} \leftarrow \textsc{RecvMsg}()$ **do**
$\qquad$**if** $(l > l') \vee ((l = l') \wedge (h > h' + 1))$ **then**
$\qquad\quad l \leftarrow l'$, $h \leftarrow h' + 1$, $m \leftarrow n$, $\mathcal{C}_\mathcal{S} \leftarrow \mathcal{C}_u$
$\qquad\quad$**for all** $i \in \mathcal{N}_u$ **do**
$\qquad\qquad\textsc{SendMsg}(i, u, h, l, \mathcal{C}_\mathcal{S})$
$\qquad\quad$**end for**
$\qquad$**else if** $(l = l') \wedge (h < h')$ **then**
$\qquad\quad\mathcal{C}_\mathcal{S} \leftarrow \textsc{CoreSet}(\mathcal{C}_\mathcal{S} \cup \mathcal{C}_{\mathcal{S}'} \cup \mathcal{C}_u)$
$\qquad\quad\textsc{SendMsg}(n, u, h, l, \mathcal{C}_\mathcal{S})$
$\qquad\quad$**if** $m \neq$ NIL **then**
$\qquad\qquad\textsc{SendMsg}(m, u, h, l, \mathcal{C}_\mathcal{S})$
$\qquad\quad$**end if**
$\qquad$**else if** $(l = l') \wedge (m = n) \wedge (\mathcal{C}_\mathcal{S} \neq \mathcal{C}_{\mathcal{S}'})$ **then**
$\qquad\quad\mathcal{C}_\mathcal{S} \leftarrow \textsc{CoreSet}(\mathcal{C}_u \cup \mathcal{C}_{\mathcal{S}'})$
$\qquad\quad$**for all** $i \in \mathcal{N}_u$ **do**
$\qquad\qquad\textsc{SendMsg}(i, u, h, l, \mathcal{C}_\mathcal{S})$
$\qquad\quad$**end for**
$\qquad$**end if**
$\quad$**end while**
end procedure

distributed Algorithm 2 to identify a small KY core set $\mathcal{C}_{\mathcal{S}}$ such that $\mathrm{MVEE}\,(\mathcal{C}_{\mathcal{S}})$ will enclose all robots in the same connected component of the communication graph. Lines 2–11 are similar to Algorithm 1. When a message is received from a descendant, the robot updates its KY core set estimate (line 13) and notifies both its master (line 16) and the descendant (line 14). When a message containing a different core set estimate is received from its master, the robot updates its estimate and notifies its neighbours (lines 19–22).

Theorem 2: Assuming all swarm robots use the same orthogonal basis in $\mathbb{R}^d$ in the initialization phase of the KY algorithm, Algorithm 2 is guaranteed to converge to the same KY core set $\mathcal{X}_{\mathrm{KY}}$ as the one obtained by the centralized KY core set algorithm [12].

Proof: Recalling the two stages of the centralized KY algorithm, we write the identified core set using [12] as $\mathcal{C}_{KY} = \mathcal{X}_{\mathrm{BB}} \cup \mathcal{X}'$, where $\mathcal{X}_{\mathrm{BB}}$ and $\mathcal{X}'$ are the initial KY core set approximation (first stage) and the global furthest violator points set based on that (second stage) respectively. Since each robot uses the same orthogonal basis in the initialization process in CORESET(), as messages propagate through the network, each robot will finally have the same $\mathcal{X}_{\mathrm{BB}}$ as the one obtained from the centralized KY core set algorithm. Besides, with line 13 and 19 all violators $\mathbf{q}_k \in \mathcal{X}'$ for $k = 1, \ldots, |\mathcal{X}'|$ are added to the KY core set maintained by each robot. Thus, all robots eventually agree on both $\mathcal{X}_{\mathrm{BB}}$ and $\mathcal{X}'$ with $\mathcal{X}_{\mathrm{KY}} \leftarrow \mathcal{C}_{KY}$, and then the algorithm terminates due to no more messages being transmitted. ■

Algorithm 2 has the benefit over Algorithm 1 that all robots are aware whether they are in the KY core set or not. But due to necessary message transmission from parents to descendants in the spanning tree for Algorithm 2, more messages are sent than in Algorithm 1.

B. Distributed Leader Selection with Anonymous Robots

In many real-world situations, the swarm's communication graph topology may change over time and communication links may appear or disappear due to changing environmental conditions (e.g. dynamic obstacles, multiple inner-connected components). In addition, it may be necessary to dynamically add or remove robots from the swarm without engineering the swarm in advance to ensure uniquely identified robots. For these cases, we consider distributed leader selection and computation of the MVEE with anonymous robots that don't rely on a fixed graph topology. Again, we discuss two approaches with CH-KY and KY core sets respectively.

1) CH-KY Approach (Anonymous): Each anonymous swarm robot can execute asynchronous distributed Algorithm 3 to identify CH-KY core set for the swarm. The algorithm takes as input $\mathcal{H}_u$, which is a set of points that describe the geometry of the robot itself. If robot geometry is not important, this can just be the position of the robot's centroid. After each robot initializes its estimate of the convex hull vertices (CH core set $\mathcal{H}_{\mathcal{S}}$) and CH-KY core set ($\mathcal{C}_{\mathcal{S}}$) for the robot swarm (line 2), it propagates this message of CH core set to its direct neighbours (line 3), and then falls into the loop (lines 4–10) that for each received message it keeps updating its CH and CH-KY core set and broadcasting the updated CH core set message to others until the received CH core set is identical to its current estimate, which eventually causes the algorithm to implicitly terminate due to that the CH core set for each connected component is unique. The algorithm does not assume that the graph topology is fixed during execution.

Algorithm 3 Distributed CH-KY Core Set Selection with Anonymous Robots

```
1: procedure DISTRIBUTEDANONYMOUSCHKY(H_u)
2:     H_S ← H_u, C_S ← CORESET(H_u)
3:     BROADCASTMSG(H_S)
4:     while H_S' ← RECVMSG() do
5:         if H_S ≠ H_S' then
6:             H_S ← CONVEXHULL(H_S ∪ H_S')
7:             C_S ← CORESET(H_S)
8:             BROADCASTMSG(H_S)
9:         end if
10:    end while
11: end procedure
```

2) KY Approach (Anonymous): Each anonymous swarm robot can execute asynchronous distributed Algorithm 4 to identify the KY core set for the swarm. Each robot communicates and updates its KY core set estimate until the estimate agrees with all of its neighbors. As discussed in the termination analysis in *Theorem 2* for the KY approach with uniquely identified robots, this exchange of information will eventually result in all robots agreeing on single KY core set and hence lead to the termination of the algorithm.

Algorithm 4 Distributed KY Core Set Selection with Anonymous Robots

```
procedure DISTRIBUTEDANONYMOUSKY(C_u)
    C_S ← C_u
    BROADCASTMSG(C_S)
    while C_S' ← RECVMSG() do
        if C_S ≠ C_S' then
            C_S ← CORESET(C_S' ∪ C_S ∪ C_u)
            BROADCASTMSG(C_S)
        end if
    end while
end procedure
```

Similar to Algorithm 3, Algorithm 4 does not assume the communication graph topology is fixed, but it does assume that robot do not move during execution. Compared to Algorithm 1 and Algorithm 2 which require uniquely identified swarm robots, even though Algorithm 3 and Algorithm 4 are simpler, their lack of use of the graph structure usually results in a larger number of messages transmitted during execution. Note that the function CORESET() may also implement other algorithms such as the minimal core set algorithm [9] (for $d = 2$) and be proven to converge similarly to the proof of *Theorem 2*.

Remark 1: If the robot position information is noisy, each robot can maintain a particle set (e.g. for particle filter), instead of the deterministic point set formulating its body, to approximate the belief distribution of the robot's position and form its initial core set estimate. In this way, the proposed algorithms can generate a core set whose MVEE contains all hypotheses of all robot positions that characterizes the swarm boundary.

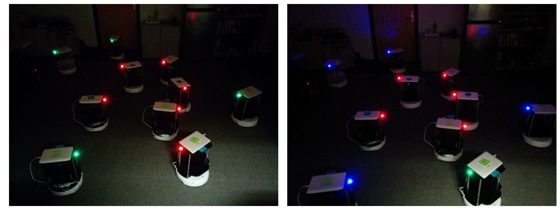
VI. Results

A. Simulation Results

Figure 1 shows the distributed KY core set algorithm implemented on a simulated robot swarm with 35 robots labeled with their UIDs, where each robot is only able to communicate with its direct neighbours due to its limited communication range $R = 25$. The communication graph and the spanning tree for the system is shown in Figure 1a. In simulation, we track "time steps" for the purpose of estimating the time to convergence for each algorithm. At each time step every robot will handle all of its received messages from the last "time step" and send out new messages as determined by the algorithm. As shown in Figure 1a, after n time steps, where n is equal to the maximum hops from leader in the graph ($n = 5$ in this case), the distributed KY algorithm will implicitly form a complete spanning tree (red edges) in the graph rooted at the black robot with UID 1. Figure 1b shows the intermediate state of the algorithm at the same time step $T = 5$, where there are still five different hypothesis core sets each producing one of the blue ellipsoids. At time step $T = 9$, Figure 1c shows that the algorithm terminates (no more messages sent) having converged to one specific KY core set consisting of 4 robots (red) that can well characterize the swarm boundary. For comparison, the distributed CH-KY core set algorithm was also implemented on the same swarm. As in the preceding discussion, although the selected CH-KY core set is identical to the distributed KY core set, only the robot with minimum UID, instead of all the robots, knows the CH-KY core set in the CH-KY algorithm.

To further compare the performance of each algorithm in a robot swarm with / without UIDs, 1000 simulation trials were conducted with each trial consisting of a randomly distributed robot swarm containing 50 robots. For each trial, we ran the distributed KY, distributed CH, distributed CH-KY and centralized KY algorithms on the swarm with UIDs and then the distributed anonymous KY and distributed anonymous CH-KY algorithm on the same swarm without UIDs (anonymous). In particular, for this simulation with planar robots, an optimal (minimum size) core set can be computed using the centralized algorithm in [9], which served as a benchmark for the minimal core set size that can be achieved for each trial.

The core set size comparison is given in Figure 2a. As proven in the preceding discussion, the converged core sets from distributed KY, distributed CH-KY, distributed anonymous KY and CH-KY algorithms are identical to the one computed from the centralized KY algorithm. In addition, the proposed KY-based distributed algorithms identify core sets that are minimum size (match the benchmark) most of the time. Since our algorithms are derived from the centralized KY algorithm, they are directly applicable to both 2D and 3D robot swarms, rather than being limited to 2D as in [9].



(a) (b)

Fig. 3: Distributed algorithms running on 10 TurtleBots. Robots with LEDs lit red are not in any core set. (a) Distributed KY algorithm results. Robots with LEDs lit green are in the KY core set. (b) Distributed CH-KY algorithm results. Robots with LEDs lit blue are in the CH-KY core set. Robots with LEDs lit green are in the CH core set, but not KY core set.

	Time (s)	Mess. Transmitted	Ave. Mess. Size (bytes)
KY	9.74	881	192
CH-KY	3.96	60	191
Anonymous KY	8.54	1045	95
Anonymous CH-KY	5.27	533	101

TABLE I: Experimental results summary. Convergence time, number of messages transmitted and average message size for each of the distributed algorithms are reported. In all cases, the final core set included 4 robots.

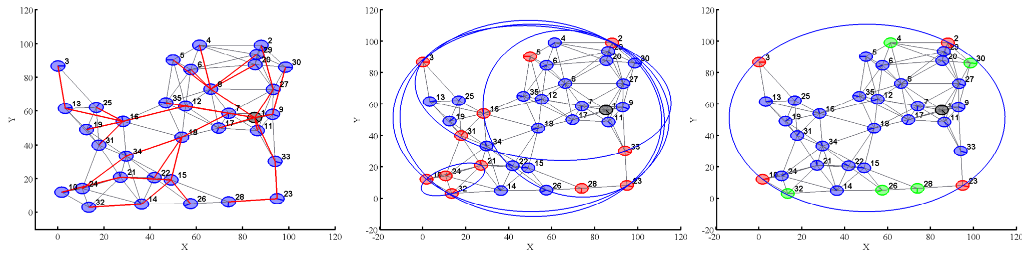
With the proposed algorithms the size of the core set is still independent of the number of robots and in large-scale robot swarms it would still return a small set of boundary robots.

In Figure 2b–Figure 2d, algorithm performance on communication-related metrics is reported. The message size is represented by the number of floating point numbers transmitted in one message (e.g. IDs, position coordinates). Since each message includes a core set estimate, message size also indirectly reflects the intermediate core set size for each algorithm. In Figure 2b, note that the anonymous distributed algorithms converge in fewer communication rounds (time steps), but as shown in Figure 2c, the average number of messages transmitted across trials is significantly higher than for algorithms using uniquely identified robots.

B. Experimental Results

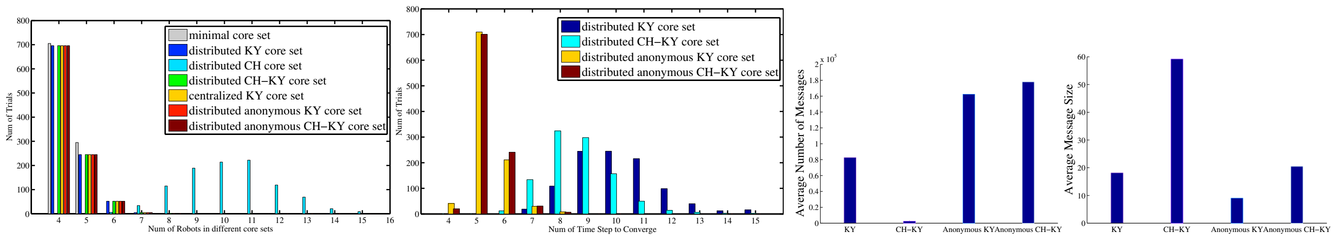
To evaluate our algorithms on real robots, we conducted experiments on a swarm of 10 TurtleBots running the Robot Operating System (ROS). Each TurtleBot was assigned a unique UID (only for experiments involving Algorithm 1 and 2) and communicated with its neighbours in a predefined communication graph. Each TurtleBot estimated its position from an Extended Kalman Filter using onboard sensor measurements (wheel encoders, gyroscope). Using the algorithms described in Section V, the TurtleBots autonomously identified a representative core set to generate a MVEE for the swarm. Qualitative results are shown in Figure 3.

Table I summarizes quantitative results from experiments on 10 TurtleBots. All algorithms returned a core set consisting of 4 robots. The average message sizes for anonymous KY and anonymous CH-KY algorithms are much lower than KY and CH-KY with UIDs respectively. However, since the algorithms for anonymous robots do not exploit any communication graph structure (spanning tree), the number of messages transmitted during their execution is much higher than their UID counterparts.



(a) (b) (c)

Fig. 1: Simulation Example on a robot swarm with 35 uniquely identified robots. (a) Robot swarm communication graph (grey) and spanning tree (red) at time step $T = 5$. (b) Intermediate state of the estimated MVEE for the swarm at time step $T = 5$. Blue ellipses show individual robot's current MVEE estimates and red robots are members of the current KY core set estimates. (c) Final MVEE (blue ellipse) formed from KY core set (red) at time step $T = 9$. Robots belonging only to the CH core set are marked by green.



(a) Final Core Set Size

(b) Number of Time Steps to Convergence

(c) Average Number of Messages Transmitted

(d) Average Message Size

Fig. 2: Simulation Results Summary.

VII. Conclusion

We developed four asynchronous distributed algorithms for information leader selection in swarm robot systems based on local communication and considering both cases of uniquely identified robots and anonymous robots. Our algorithms identify a small core set of robots which are a representative subset of robots at the boundary of the swarm that may be communicated back to the human operator to generate an MVEE that is guaranteed to enclose all robots in the swarm. Key performance properties (e.g. execution time, average message size, total number of messages transmitted) and convergence of the proposed algorithms were explored and proven. The performance for the algorithms was evaluated in both simulations and experiments.

References

[1] S. Nagavalli, S.-Y. Chien, M. Lewis, N. Chakraborty, and K. Sycara, "Bounds of neglect benevolence in input timing for human interaction with robotic swarms," in *ACM/IEEE Int'l Conf. on Human-Robot Interaction*, 2015, pp. 197–204.

[2] S. Nunnally, P. Walker, A. Kolling, N. Chakraborty, M. Lewis, K. Sycara, and M. Goodrich, "Human influence of robotic swarms with bandwidth and localization issues," in *IEEE Int'l Conf. on Systems, Man, and Cybernetics*, 2012, pp. 333–338.

[3] P. Walker, S. Nunnally, M. Lewis, A. Kolling, N. Chakraborty, and K. Sycara, "Neglect benevolence in human control of swarms in the presence of latency," in *IEEE Int'l Conf. on Systems, Man, and Cybernetics*, 2012, pp. 3009–3014.

[4] A. Becker, C. Ertel, and J. McLurkin, "Crowdsourcing swarm manipulation experiments: A massive online user study with large swarms of simple robots," in *IEEE Int'l Conf. on Robotics and Automation*, 2014, pp. 2825–2830.

[5] N. Michael, C. Belta, and V. Kumar, "Controlling three dimensional swarms of robots," in *IEEE Int'l Conf. on Robotics and Automation*, 2006, pp. 964–969.

[6] S. Kalantar and U. R. Zimmer, "Distributed shape control of homogeneous swarms of autonomous underwater vehicles," *Autonomous Robots*, vol. 22, no. 1, pp. 37–53, 2007.

[7] M. Henk, "Löwner–john ellipsoids," *Documenta Math*, pp. 95–106, 2012.

[8] F. John, "Extremum problems with inequalities as subsidiary conditions," in *Studies and Essays Presented to R. Courant on his 60th Birthday, January 8, 1948*. Interscience Publishers, Inc., New York, N. Y., 1948, pp. 187–204.

[9] B. Gärtner and S. Schönherr, "Smallest enclosing ellipses–fast and exact," *Technical report B 97-03*, Free University, Berlin, 1997.

[10] L. G. Khachiyan, "Rounding of polytopes in the real number model of computation," *Mathematics of Operations Research*, vol. 21, no. 2, pp. 307–320, 1996.

[11] M. J. Todd and E. A. Yıldırım, "On khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids," *Discrete Applied Mathematics*, vol. 155, no. 13, pp. 1731–1744, 2007.

[12] P. Kumar and E. A. Yildirim, "Minimum-volume enclosing ellipsoids and core sets," *Journal of Optimization Theory and Applications*, vol. 126, no. 1, pp. 1–21, 2005.

[13] S. D. Ahipaşaoğlu, "Fast algorithms for the minimum volume estimator," *Journal of Global Optimization*, vol. 62, no. 2, pp. 351–370, 2015.

[14] D. Martinez-Rego, E. Castillo, O. Fontenla-Romero, and A. Alonso-Betanzos, "A minimum volume covering approach with a set of ellipsoids," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2997–3009, 2013.

[15] J. McLurkin and E. D. Demaine, "A distributed boundary detection algorithm for multi-robot systems," in *IEEE Int'l Conf. on Robots and Systems*, 2009, pp. 4791–4798.

[16] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, Dec. 1996.

[17] U. Betke and M. Henk, "Approximating the volume of convex bodies," *Discrete & Computational Geometry*, vol. 10, no. 1, pp. 15–21, 1993.

[18] S. Nagavalli, A. Lybarger, L. Luo, N. Chakraborty, and K. Sycara, "Aligning coordinate frames in multi-robot systems with relative sensing information," in *IEEE Int'l Conf. on Robots and Systems*, 2014, pp. 388–395.