# Handling State Uncertainty in Distributed Information Leader Selection for Robotic Swarms

Anqi Li*, Wenhao Luo*, Sasanka Nagavalli*, *Student Member, IEEE*,
Nilanjan Chakraborty†, *Member, IEEE*, Katia Sycara*, *Fellow, IEEE*

***Abstract*—In many scenarios involving human interaction with a remote swarm, the human operator needs to be periodically updated with state information from the robotic swarm. A complete representation of swarm state is high dimensional and perceptually inaccessible to the human. Thus, a summary representation is often required. In addition, it is often the case that the human-swarm communication channel is extremely bandwidth constrained and may have high latency. This motivates the need for the swarm itself to compute a summary representation of its own state for transmission to the human operator. The summary representation may be generated by selecting a subset of robots, known as the information leaders, whose own states suffice to give a bounded approximation of the entire swarm, even in the presence of uncertainty. In this paper, we propose two fully distributed asynchronous algorithms for information leader selection that only rely on inter-robot local communication. In particular, by representing noisy robot states as error ellipsoids with tunable confidence level, the information leaders are selected such that the Minimum-Volume Covering Ellipsoid (MVCE) summarizes the noisy swarm state boundary. We provide bounded optimality analysis and proof of convergence for the algorithms. We present simulation results demonstrating the performance and effectiveness of the proposed algorithms.**

## I. Introduction

Robotic swarms consist of a large number of robots whose global behavior emerges from simple control laws and collective decision making based on local communication or sensing. Since each robot only uses local information from other robots within its spatial neighborhood, robotic swarms demonstrate high robustness and scalability, exhibiting great potential for various applications including search and rescue, environmental monitoring and disaster response. However, recent work in [1] has shown that the high dimensionality of the swarm's state space makes it hard for humans to perceptually access and interpret the state of the swarm, which causes difficulty for human interaction with robotic swarms. Therefore, it is important to simplify or summarize swarm state into low dimensional representations. Previous experiments in [2], [3] showed that a summary representation (e.g. convex hull) can help humans estimate the state of the swarm to more effectively control the swarm. Thus, human control may be enhanced by a small number of appropriately selected swarm members. In addition, when the human operates a remote swarm under conditions where human-swarm communication bandwidth is significantly constrained and human-swarm communication latency is high (e.g. underwater environments, disaster zones with minimal infrastructure), it is important for the swarm itself to compute this summary representation before transmission to the human. In [4], this group of swarm members was termed *information leaders*, whose state information enables constructing an accurate summary representation of the entire swarm. The communication constraints motivate the need to ensure there is minimal dependence between the number of robots in the swarm and the number of selected information leaders or the individual inter-robot message size.

In addition to perceptual accessibility of overall swarm state information by a human operator, another main challenge in human-swarm interaction is the noisy state information from individual robots. Each robot's state information is inherently noisy due to localization errors and when incorporated into a swarm summary representation that does not explicitly account for the noise, such as [4], there may be undesirable artifacts (e.g. incorrect information leaders whose minimum volume enclosing ellipse does not actually enclose the true positions of the robots). In this paper, we extend our previous work [4] on selecting information leaders for swarms by encoding the uncertainty of robot states into our framework. The contributions of this work are (a) distributed algorithms for information leader selection under uncertainty to generate a representative summary of the entire swarm's noisy state to a tunable, user-defined, guaranteed confidence level, (b) bounded optimality analysis and proof of convergence of the proposed distributed algorithms and (c) simulation results evaluating the performance of the algorithms for swarms of various sizes.

## II. Related Work

Within the literature discussing summary representations of robotic swarms, ellipsoids are a commonly used geometric abstraction to describe the overall state of the robot system. For instance, [5] constructs an ellipsoid as an abstraction of a group of robots using first and second order geometric moments. However, [5] assumes that global information (i.e. geometric moments) is shared by all robots in the robotic swarm, which is impractical in multi-robot systems where robots can only communicate with their direct neighbors. Based on this geometric moments abstraction, [6] proposes a distributed proportional-integral average consensus estimator to estimate the global information. Although these geometric moment based algorithms can represent average information

*The authors are with the Robotics Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. Email: {anqil1, wenhaol, 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.

978-1-5090-1897-0/16/$31.00 ©2016 IEEE

among robotic swarm members which enables convenient control and state estimation, these abstractions fail to take care of boundary information without further assumptions regarding distribution of robotic swarm members (e.g. normal distribution [5]). In contrast with geometric moment based algorithms, [4] introduces an abstraction based on the minimum volume enclosing ellipsoid (MVEE) [7] that covers the position of each robot in the robotic swarm. It also defines a set of boundary robots selected in a distributed way called *information leaders* based on a core set which can be used to estimate a $(1+\epsilon)$-approximation of the MVEE.

Extending the MVEE concept for enclosing a set of points with an ellipsoid, a minimum volume covering ellipsoid (MVCE) of ellipsoids is defined as the minimum volume ellipsoid to cover a given set of ellipsoids. This problem can be formulated as a minimum determinant problem with linear matrix inequalities, which can be solved by optimization algorithms proposed in [8] and [9]. However, such optimization algorithms cannot be efficiently decentralized and thus cannot easily fit into the swarm robot framework for state estimation. However, [10] proposes an extension to the first-order Khachiyan's algorithm[11], which constructs an approximation of MVCE in an incremental way which can be conveniently decentralized. Moreover, since this algorithm results in a core set which generates a close approximation of the MVCE of all the ellipsoids, it can also be useful for selection of information leaders.

In this paper, we extend the minimum volume enclosing ellipsoid based algorithms proposed in [4]. In contrast to [4], we use a probabilistic representation which describes the position of each robot with a confidence ellipsoid instead of an exact position. We also propose an asynchronous distributed framework for the MVCE core set algorithm proposed in [10].

## III. Problem Statement

Consider a robotic swarm consisting of $N$ robots. The position of each robot is defined as $\mathbf{q}_i \in \mathbb{R}^d$ with $d \in \{2,3\}$ in common reference frames. Each robot can only communicate with its direct neighbors. The communication graph is given by $\mathcal{G} = (V, E)$, where each node $i \in V$ represents a robot. If robot $i \in V$ can communicate with robot $j \in V$, then edge $(i,j) \in E$. We assume that the communication graph is undirected and connected.

In contrast to [4], we do not assume we have perfect state information about every robot and instead assume a probabilistic distribution over robot states. In the scope of this paper, we assume that this distribution satisfies Gaussian distribution. However, it is easy to see that the algorithm we discuss in this paper can be easily extended to any other probabilistic distribution with closed iso-contour.

*Confidence Level Ellipsoids*: For each single robot, we define the $\gamma$-confidence level ellipsoid as,

$$\begin{aligned}\mathcal{E}_i := &\Big\{x \in \mathbb{R}^d : (x-\mu_i)^T A_i(\gamma)(x-\mu_i) \le 1\Big\} \\ \text{s.t.} \quad &\int_{x\in\mathcal{E}_i} p(\mathbf{q}_i = x) = \gamma\end{aligned} \tag{1}$$

Therefore, we have a set of $\gamma$-confidence level ellipsoids $\mathcal{S} = \{\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_n\}$ over the robotic swarm. Without loss of generality, we assume that each of the ellipsoids $\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_N$ is of full dimension, since every degenerate ellipsoid can be approximated by a full-dimensional ellipsoid with some axes extremely short. We define the ellipsoidal norm between a point $x \in \mathbb{R}^d$ and ellipsoid determined by $A, \mu$ as $d_{\mathcal{E}}(x, A, \mu) = \sqrt{(x-\mu)^T A(x-\mu)}$. For simplicity of exposition, we will use $\mathcal{E} = (A, \mu)$ to refer to ellipsoid $\mathcal{E}$ defined by matrix $A$ and $\mu$.

Since we assume that exact position of each robot given its estimation satisfies Gaussian distribution, the probabilistic distribution is determined by mean $\mu_i$ and covariance matrix $\Sigma_i$. In practice, $\mu_i$ and $\Sigma_i$ can be either directly known from sensor or computed by estimators (e.g. Kalman filter).

From the statistics literature [12], [13], for normal distributions, the probability of an observation is defined by the Mahalanobis distance $d_{\Sigma_i^{-1}}(\mathbf{q}_i, \mu_i)$, where

$$d_{\Sigma_i^{-1}}(\mathbf{q}_i, \mu_i) = \sqrt{(\mathbf{q}_i - \mu_i)^T \Sigma_i^{-1} (\mathbf{q}_i - \mu_i)} \tag{2}$$

and $d_{\Sigma_i^{-1}}(\mathbf{q}_i, \mu_i)^2$ satisfies chi-squared distribution and the degree of freedom of chi-squared distribution is same as the dimension of space where the swarm is operated. The probability that true value is $t$ away from $\mu_i$ in terms of Mahalanobis distance is $\sqrt{\chi^2_{1-\gamma}}$. In other words, the $p$-confidence level ellipsoid is $\mathcal{E}_i = ((1/\chi^2_{1-\gamma})\Sigma_i^{-1}, \mu_i)$. In 2 dimensional cases, there is a closed formed solution of the confidence level ellipsoid, with $A_i = -(1/2\ln(1-\gamma))\Sigma_i^{-1}$.

In this paper, we are interested in distributedly computing the *Minimum Volume Covering Ellipsoid* (MVCE) $\mathcal{E} = \{x \in \mathbb{R}^d : (x-\omega)^T M (x-\omega) \le 1\}$ over the set of confidence level ellipsoid $\mathcal{S} = \{\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_N\}$, which is formulated as follows.

$$\begin{aligned}\arg\min_{M,\omega} \;& \log\det\left(M^{-1}\right) \\ \text{s.t.} \quad & \mathcal{E}_i \subseteq \mathcal{E}, \quad i = 1, 2, \dots, N \\ & M \succ 0\end{aligned} \tag{3}$$

## IV. Minimum Volume Covering Ellipsoid and Core Sets

Instead of optimization based algorithms which are hard to fit into a distributed framework, there are core set based MVCE algorithms [10], [11], [14], [7] that are beneficial to our problem. Instead of computing the exact MVCE over the whole set of ellipsoids, these algorithms identify a core set of ellipsoids to generate a close approximation to the exact MVCE.

*Definition 1:* Given $\epsilon > 0 \in \mathbb{R}$ and a compact convex set $\mathcal{S} \subset \mathbb{R}^d$, an ellipsoid $\mathcal{E} = (A, \mu)$ is said to be a $(1+\epsilon)$-approximation to $\mathrm{MVCE}(\mathcal{S})$ if

$$\mathcal{E} \supseteq \mathcal{S}, \ \mathrm{vol}\ \mathcal{E} \le (1+\epsilon)\mathrm{vol}\ \mathrm{MVCE}(\mathcal{S}),\ \mathrm{MVCE}(\mathcal{S}) \subseteq (1+\epsilon)\mathcal{E} \tag{4}$$

where $(1+\epsilon)\mathcal{E}$ is defined as $(1+\epsilon)\mathcal{E} := \{x \in \mathbb{R}^d : (x-\mu)^T A (x-\mu) \le 1\}$, i.e., $(1+\epsilon)\mathcal{E} = (A/(1+\epsilon), \mu)$.

A core set is a small subset $\mathcal{X} \in \mathcal{S}$, such that using this subset gives the approximately same MVCE as the one obtained from the entire set $\mathcal{S}$. In this paper, we define the core set $\mathcal{X}_{\mathcal{E}}$ of $\mathcal{S}$ as the subset of ellipsoids (confidence ellipsoids of robots) that can approximate the MVCE of the entire set. It satisfies

$$\begin{aligned} \text{vol MVCE}(\mathcal{X}_{\mathcal{E}}) \leq \text{vol MVCE}(\mathcal{S}) &\leq (1+\epsilon)\text{vol MVCE}(\mathcal{X}_{\mathcal{E}}) \\ &\leq (1+\epsilon)\text{vol MVCE}(\mathcal{S}) \end{aligned} \quad (5)$$

### A. Centralized MVCE-from-MVEE Algorithm

The general idea of the discretized MVCE-from-MVEE algorithm is to discretize the boundary of the confidence level ellipsoid into a set of points $\mathcal{S}_i \subset \partial\mathcal{E}_i$, and approximate confidence level ellipsoid by the polyhedron which is the convex hull of $\{\mathcal{S}_i\}$. If the discretization is fine enough, then the MVEE over $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_N\}$ can give a sufficient good approximation of MVCE over ellipsoids $\{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_N\}$.

*Theorem 1:* If localization of each robot is reasonably good, where any axis of the estimated MVEE $\mathcal{E} = (A, \mu)$ over $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_N\} \subset \mathbb{R}^d$ is greater than the length of the major axis of each confidence level ellipsoid, then $\mathcal{E}$ is guaranteed to be an $(1+\epsilon)$-approximation of the Minimum Volume Covering Ellipsoid over $\{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_N\} \subset \mathbb{R}^d$ if $\mathcal{S}_i$ is discretized by same angle interval over $\mathcal{E}_i = (A_i, \mu_i)$, and discretization interval $\Delta\theta$ satisfies,

$$\Delta\theta \leq 2\arctan\left(r_{\min}\sqrt{(1+\epsilon)^{2/d} - 1}\right) \quad (6)$$

where $r_{\min}$ is the maximum among square root of ratio of the minimum and maximum singular value of matrix $A_i$,

$$r_{\min} = \min\left\{\sqrt{\frac{\sigma_{\min}(A_i)}{\sigma_{\max}(A_i)}}, \quad i = 1, \ldots, N\right\}$$

*Proof:* Consider the worst case that two nearby points are located symmetrically about the major axis of an ellipsoid $\mathcal{E}_i = (A_i, \mu_i)$ and the major axis of $\mathcal{E}_i$ coincides with an axis of MVEE $\mathcal{E} = (A, \mu)$. Let the convex hull intersect with that axis at $x_c$, MVEE intersects with that axis at $x_e$ and the furthest point on $\mathcal{E}_i$ located on $x_m$, which is obviously a point on major axis of $\mathcal{E}_i$. To prove Theorem 1, what we have to prove is that in this worst case, $\|x_e - \mu\|_2 / \|x_m - \mu\|_2 \geq 1/(1+\epsilon)$.

As is proved in [4], $\|x_e - \mu\|_2 \leq \|x_c - \mu\|_2$. Based on our assumption that localization is reasonably good and axis of $\mathcal{E}_i$ and $\mathcal{E}$ coincide, we have

$$\frac{\|x_e - \mu\|_2}{\|x_m - \mu\|_2} \geq \frac{\|x_c - \mu\|_2}{\|x_m - \mu\|_2} \geq \frac{\|x_c - \mu_i\|_2}{\|x_m - \mu_i\|_2}$$

Therefore, Theorem 1 holds if we can guarantee that $\|x_c - \mu_i\|_2 / \|x_m - \mu_i\|_2 \geq 1/(1+\epsilon)$. The rest of proof follows directly from geometry. ■

Note that Theorem 1 also shows that a bound of discretization interval does not exist without certain assumptions on the shape of the ellipsoid. For example, the bound of discretization interval needs to be extremely small if we consider a near degenerate case. Furthermore, since (6) is a worst case bound, the actual output MVEE of this algorithm may be much closer to the exact MVCE, which results in unnecessary enlargement of the estimated MVCE (note that the estimated MVCE is the $(1+\epsilon)$ expansion of the output MVEE). Another remark is that the cost of discretization grows exponentially as the number of dimensions increase. This means that there is a significant difference between the computational cost of this algorithm in 2 dimensional space and 3 dimensional space.

### B. Centralized MVCE-KY Algorithm

The centralized MVCE-KY algorithm described in [10] is a two stage algorithm. The first stage initializes the core set of size $|\mathcal{X}_0| = 2d$ to generate an initial volume approximation with a bounding box. Each ellipsoid is traversed to find the 2 furthest points in $d$ orthogonal directions. This subroutine has the following analytical solutions

$$x_{i,k}^* = \arg\max_{x \in \mathcal{E}_k}\left\{\pm\left(b^i\right)^T x\right\} = \mu_k \pm \frac{(U_k)^{-1}(U_k)^{-T} b^i}{\left\|(U_k)^{-T} b^i\right\|} \quad (7)$$

where $b^i$ is the unit vector in $i$th direction and $U_k$ is obtained by the Cholesky factorization of $A_k$, which is the shape matrix for each ellipsoid $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_N$. It is noted that the initial estimation $\mathcal{E} = (M_0, \omega_0)$ is constructed by the center of mass of $\mathcal{X}_0$ as center $\omega_0$, and the inverse of the scaled covariance matrix of $\mathcal{X}_0$ as matrix $M_0$.

Then the initial volume approximation ellipsoid is recursively expanded to include the furthest violator point $x_{new} = \arg\max_{x \in \mathcal{S}}\{(x-\omega)^T M(x-\omega)\}$ and expand the estimate of the MVCE until the squared ellipsoid norm of the furthest violator point is within threshold. The algorithm ends up with a core set which spans an ellipsoid which is a $(1+\epsilon)$-approximation to the exact MVCE of ellipsoids.

Note that the sub-procedure for finding the furthermost violator in each $\mathcal{E}_i$, $i = 1, \ldots, N$ w.r.t. current estimation of MVCE is a nonlinear optimization problem. In terms of distributed computation, the centralized MVCE-KY algorithm can be regarded as the basic function to compute the MVCE and core set for sets of ellipsoids.

$$\begin{aligned} &\arg\max_{x}\ (x-\omega_k)^T M_k (x-\omega_k) \\ &\text{s.t.} \quad (x-\mu_i)^T A_i (x-\mu_i) \leq 1 \end{aligned} \quad (8)$$

This problem is a quadratic optimization problem, and can be solved by nonlinear optimization techniques such as *Sequential Quadratic Programming* (SQP).

## V. Distributed Information Leader Selection Using Core Sets

Assume each robot is labeled with a unique identifier (UID) that can be used to identify each other. Without loss of generality, we assume that $\text{UID}(i) = i$. The UIDs of direct neighbors of robot $v_i$ in the communication graph are defined by the set $\mathcal{N}_i = \{j \mid j \in V : (i,j) \in E\}$. In this section, we will discuss algorithms that solve the MVCE problem in a distributed manner and only require local communication between connected robots.

### A. Distributed MVCE-from-MVEE Algorithm

As introduced in Section IV-A, the MVCE-from-MVEE algorithm discretizes each error ellipsoid of a robot into a set of boundary points and approximates the ellipsoid based on the convex hull of these discrete points. Then the MVEE of these points from all the robots can provide a sufficiently precise approximation to MVCE over confidence level ellipsoids.

Since this algorithm only computes the MVEE over a finite set of points, we adopt the framework introduced in [4], [15] into Algorithm 1, which uses the CH-KY framework and calls the exact MVEE core set algorithm presented in [14] to select MVEE core set. Note that [14] never computes the MVEE explicitly, but rather outputs the MVEE core set directly.

**Algorithm 1** Distributed MVCE-from-MVEE Algorithm Core Set Selection

```
1: procedure DISTRIBUTED MVCE-FROM-MVEE(u, E_u, N_u)
2:     l ← u, h ← 0, m ← NIL
3:     H_S ← DISCRETIZE(E_u), C_S ← MVEECORESET(H_S)
4:     for all i ∈ N_u do
5:         SENDMSG(i, u, h, l, H_S)
6:     end for
7:     while {n, h', l', H_S'} ← RECVMSG() do
8:         if (l > l') ∨ ((l = l') ∧ (h > h' + 1)) then
9:             l ← l', h ← h' + 1, m ← n
10:            H_S ← DISCRETIZE(E_u), C_S ← MVEECORESET(H_S)
11:            for all i ∈ N_u do
12:                SENDMSG(i, u, h, l, H_S)
13:            end for
14:        else if (l = l') ∧ (h < h') then
15:            H_S ← CONVEXHULL(H_S ∪ H_S')
16:            if m ≠ NIL then
17:                SENDMSG(m, u, h, l, H_S)
18:            else
19:                C_S ← MVEECORESET(H_S)
20:            end if
21:        end if
22:    end while
23: end procedure
```

In Algorithm 1 every robot first initializes its belief of leader UID $l$, number of hops $h$ from leader, master UID $m$, convex hull estimate $\mathcal{H}_{\mathcal{S}}$, and MVEE core set estimate $\mathcal{C}_{\mathcal{S}}$ with its own knowledge (lines 2–3) and sends its estimate to its neighbors (lines 4–6).

Then the algorithm simultaneously performs two tasks: (1) implicitly establish a spanning tree rooted at the robot leader with the lowest UID, which is optimal in terms of number of hops to leader and the diameter of the tree (lines 8–13), and (2) propagate convex hull information from leaves of the tree to the root robot leader (lines 15–17).

Note that the convex hull has the property that given several sets of points $\mathcal{X}_1, \ldots, \mathcal{X}_n$,

$$\mathrm{Conv}\left(\bigcup_{i=1,\ldots,n} \mathcal{X}_i\right) = \mathrm{Conv}\left(\bigcup_{i=1,\ldots,n} \mathrm{Conv}\left(\mathcal{X}_i\right)\right) \tag{9}$$

which means that the convex hull of the union of convex hulls is the convex hull of $\mathcal{X}_1, \ldots, \mathcal{X}_n$. Therefore, messages in Algorithm 1 are only needed to be sent one way from leaves to the root of tree and there is no convergence issue. The total number of messages transmitted is $O(|V| + |E|)$ and the size of message is bounded by $O(|V|)$.

The algorithm will end up with the leader having the correct estimate of MVCE-from-MVEE core set. However, this information is not shared by other robots.

### B. Distributed MVCE-KY Algorithm

As is pointed out in [4], core sets selected by KY algorithm in [4] for MVEE problem don't have the same property that the union of two core sets contains all the points enclosed by both individually. It is the same for MVCE-KY core set. Namely, given core sets $\mathrm{MVCE}\left(\mathcal{X}_{\mathcal{E}A}\right) \subseteq \mathrm{MVCE}\left(\mathcal{S}_A\right)$ and $\mathrm{MVCE}\left(\mathcal{X}_{\mathcal{E}B}\right) \subseteq \mathrm{MVCE}\left(\mathcal{S}_B\right)$, $\mathrm{MVCE}\left(\mathcal{S}_A \cup \mathcal{S}_B\right) \neq \mathrm{MVCE}\left(\mathcal{X}_{\mathcal{E}A} \cup \mathcal{X}_{\mathcal{E}B}\right)$. Algorithm 2 accounts for this property by maintaining a hypothesis core set that is updated by parents and verified by descendants in the spanning tree. On line 15 and 21, MVCECORESET() is defined as a specific algorithm to merge several core sets and will be introduced.

**Algorithm 2** Distributed MVCE-KY Core Set Selection

```
1: procedure DISTRIBUTED MVCE-KY(u, E_u, N_u)
2:     l ← u, h ← 0, m ← NIL
3:     C_S ← E_u
4:     for all i ∈ N_u do
5:         SENDMSG(i, u, h, l, C_S)
6:     end for
7:     while {n, u', h', l', C'_S} ← RECVMSG() do
8:         if (l > l') ∨ ((l = l') ∧ (h > h' + 1)) then
9:             l ← l', h ← h' + 1, m ← n
10:            C_S ← E_u
11:            for all i ∈ N_u do
12:                SENDMSG(i, u, h, l, C_S)
13:            end for
14:        else if (l = l') ∧ (h < h') then
15:            C_S ← MVCECORESET(C_S ∪ C'_S ∪ E_u, ε)
16:            SENDMSG(n, u, h, l, C_S)
17:            if m ≠ NIL then
18:                SENDMSG(i, u, h, l, C_S)
19:            end if
20:        else if (l = l') ∧ (m = n) ∧ (C_S ≠ C'_S) then
21:            C_S ← MVCECORESET(C_S ∪ C'_S ∪ E_u, ε)
22:            for all i ∈ N_u do
23:                SENDMSG(i, u, h, l, C_S)
24:            end for
25:        end if
26:    end while
27: end procedure
```

Similar to Algorithm 1, on line 2–3, each robot initializes its estimate of leader UID $l$, number of hops $h$ from leader, master UID $m$ and core set estimate $\mathcal{C}_{\mathcal{S}}$. On lines 4–6, it sends a message to each of its neighbours $i \in \mathcal{N}_u$.

The way Algorithm 2 constructs the spanning tree is the same as MVCE-from-MVEE algorithm. However, since the MVCE-KY core set does not have a similar property to the convex hull in (9), information transmission is bi-directional. When a message is received from a descendant, the robot re-estimates its MVCE-KY core set using the algorithm introduced in Section IV-B (line 14) and sends a message to its master (line 18) and back to that descendant (line 16). When a message is received from its master, the robot updates its estimate and informs its direct neighbors, since we do not assume robots know which robot is its descendant (lines 20–25).

The algorithm terminates when every robot in the connected component has same and correct leader UID $l$, minimum hops to leader $h$ and a consensus on the MVCE-KY coreset.

*Theorem 2:* In the subroutine MVCECoreSet() of Algorithm 2, we assume each robot uses the same orthogonal basis in $\mathbb{R}^d$ for the initialization phase, which ensures the algorithm will converge to the same core set $\mathcal{X}_\mathcal{E}$ as the one obtained by the centralized MVCE core set algorithm [10].

*Proof:* Recall that the centralized MVCE-KY algorithm [10] as the subroutine of Algorithm 2 works in the following way. In the initialization phase (first stage), the initial MVCE-KY core set is identified by finding the bounding box of the input ellipsoids defined in (7). In the second stage, the MVCE core set is then recursively expanded by including the furthest violator from the current non-core set input ellipsoids at each round w.r.t. the MVCE defined from the current core set ellipsoids until no violators are found. In this manner even if the Algorithm 2 is asynchronous and distributed, a computation sequence is imposed such that as messages propagate through the robotic network, each robot will first agree on the same bounding box of all the input robot ellipsoids (given the same orthogonal basis as supporting vectors) and then compute the remaining furthest violators in the same order. Hence each robot will eventually agree on the same core set as obtained from the centralized MVCE-KY algorithm and mute from bottom to top of the spanning tree, which terminates the algorithm. ■

## VI. Results

We implement the distributed MVCE-from-MVEE algorithm and distributed MVCE-KY algorithm on a simulated 2 dimensional robotic swarm. The robots can communicate with their neighbor within communication range $R = 10$ and the ratio of length of minor axis and major axis of confidence level ellipsoids is bounded by $[1/6, 1]$. In this section, we will compare these two distributed MVCE algorithms in terms of approximation accuracy, core set size, message size and number. Figure 1 shows an example of distributed MVCE-KY algorithm on a swarm consisting of 35 robots. The spanning tree constructed by our algorithms is shown in Figure 1a. Figure 1b shows the MVCE and MVCE-KY core set estimates by robots in an intermediate step. Robots within their own estimate of the core set are shown in red. The algorithm converges to the MVCE estimate shown in Figure 1c where 4 robots within MVCE-KY core set is shown in red. As a comparison, convex hull of the swarm consists of 10 robot. Robots within the convex hull, but not within MVCE-KY core set are shown in green.

We tested both the distributed MVCE-from-MVEE algorithm and distributed MVCE-KY algorithm on simulated robot swarms with 20, 35, 50, 65 robots. For each swarm size, 20 trials were conducted and other settings are the same as the previous section. Core set size, message size and message number are compared in Figure 2.

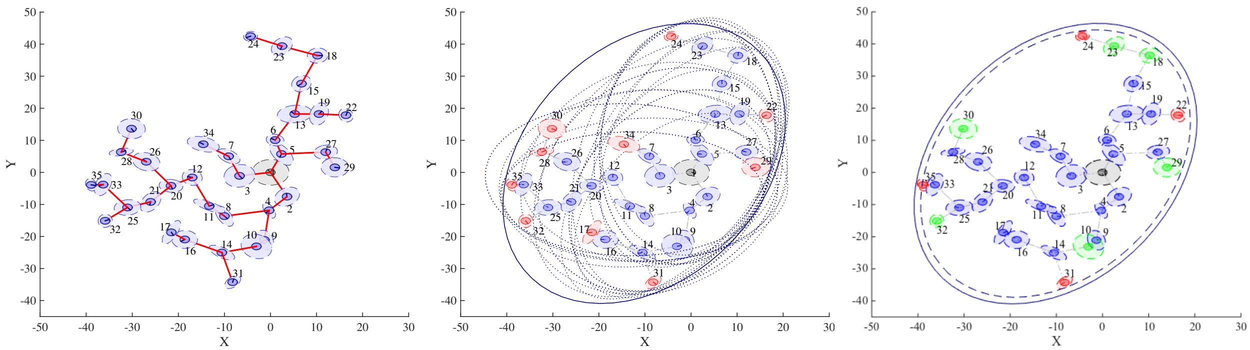
Figure 2a shows the core set size of the robot swarm in simulations of 20, 35, 50 and 65 robots respectively. For each size of the swarm, the average, maximum and minimum core set size are shown. A convex-hull over discretized points (CH) algorithm is also implemented as a benchmark. As is shown in Figure 2a, both the distributed MVCE-from-MVEE and distributed MVCE-KY algorithm have the property that the size of the core set is independent of the size of the swarm, which is not satisfied by CH algorithm. Moreover, one may observe that the core set size of the distributed MVCE-from-MVEE algorithm is sometimes a few less than the core set size of the distributed MVCE-KY algorithm. This is because the core set of the distributed MVCE-KY algorithm must include confidence level ellipsoids on the bounding box to guarantee convergence as is shown in Theorem 2. However, these ellipsoids may not lie in the exact core set which is generated by the distributed MVCE-from-MVEE algorithm.

The average and maximum message size transferred are shown in Figure 2b. The message size is independent of the number of robots in the distributed MVCE-KY algorithm, since it only transfers each robot's current estimate of the core set. However, the message size for the distributed MVCE-from-MVEE algorithm is subject to the number of boundary robots, which is dependent on swarm size and configuration.

Figure 2c shows the number of messages transmitted per robot in the two algorithms. One may observe that the number of messages for the distributed MVCE-from-MVEE algorithm is much less than that of distributed MVCE-KY algorithm. This is due to the distributed MVCE-from-MVEE transmitting convex hulls and only needing one-way message transfer from each node to its master. However, the distributed MVCE-KY algorithm requires bi-directional transmission of messages. Although transmitting more messages, the distributed MVCE-KY algorithm implicitly ends with all robots having consensus on the core set. However, in the distributed MVCE-from-MVEE algorithm, only the leader has an accurate and complete estimate of the core set. Thus, it could require more messages if a consensus over the core set is needed.

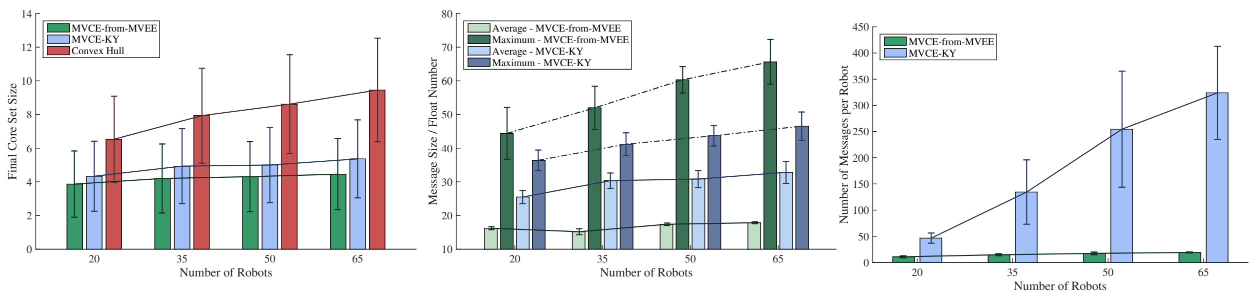
## VII. Conclusion

We introduced two distributed algorithms for information leader selection for robotic swarms that explicitly consider uncertainty of individual robot states. Both algorithms select a core set of robots which are used to generate a summary representation of overall swarm state with relatively low dimensionality and is independent of swarm size. These robots can uniquely define an ellipsoid over the environment that is guaranteed to enclose all the robots up to a desired user-defined confidence level and proof of convergence is provided. Simulation results comparing the two algorithms in the paper evaluating key properties such as core set size, message size and total number of messages show that the distributed MVCE-from-MVEE algorithm almost always results in a smaller core set size and transmits significantly fewer messages, but the MVCE-KY algorithm has a lower maximum message size.




(a) (b) (c)

Fig. 1: Simulation example for a simulated robotic swarm with 35 robots. (a) Spanning tree (red) constructed by our algorithm. Robot positions are shown in dark color but each robot only knows their confidence level ellipsoids defined in (1), shown as light color ellipsoids. Black robot 1 is the root of the tree. Other robots' confidence level ellipses are shown in light blue ellipses. (b) An intermediate state of MVCE estimation (dark blue dotted line) for robots. Robots which are in their own estimation of core set is shown in red. The final estimation of MVCE is shown in dark blue solid line. (c) Final estimation of MVCE, including both the $(1+\epsilon)$-approximation ellipse (dash line), its enlargement (solid line) and the core set, including MVCE-KY core set (red) and convex hull (both red and green)




(a) Number of Robots in Selected Core Set

(b) Message Size Per Robot (Average and Maximum)

(c) Number of Messages Per Robot

Fig. 2: Simulation results for a robot swarm consisting of 20, 35, 50 or 65 robots (20 trials each) operating in a 2-dimendional workspace. The distributed MVCE-from-MVEE algorithm almost always results in a smaller selected core set and significantly fewer messages transmitted per robot. However, in this algorithm since a convex hull must be transmitted in each message, the maximum message size grows significantly with the number of swarm robots. In contrast, the distributed MVCE-KY algorithm has a much smaller upper bound on message size, so though it has a higher message size on average, the maximum message size does not grow as much with the number of swarm robots.

## VIII. Acknowledgements

This work was supported in part by ONR Grant N0001409-10680, AFOSR Grant FA9550-15-1-0442 and an NSERC PGS D scholarship.

## References

[1] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 9–26, Feb 2016.

[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 *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 333–338.

[3] A. Becker, C. Ertel, and J. McLurkin, "Crowdsourcing swarm manipulation experiments: A massive online user study with large swarms of simple robots," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2825–2830.

[4] W. Luo, S. S. Khatib, S. Nagavalli, N. Chakraborty, and K. Sycara, "Asynchronous distributed information leader selection in robotic swarms," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug 2015, pp. 606–611.

[5] C. Belta and V. Kumar, "Abstraction and control for groups of robots," *Robotics, IEEE Transactions on*, vol. 20, no. 5, pp. 865–875, 2004.

[6] F. Morbidi, R. A. Freeman, and K. M. Lynch, "Estimation and control of uav swarms for distributed monitoring tasks," in *American Control Conference (ACC), 2011*. IEEE, 2011, pp. 1069–1075.

[7] 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.

[8] K.-C. Toh, "Primal-dual path-following algorithms for determinant maximization problems with linear matrix inequalities," *Computational Optimization and Applications*, vol. 14, no. 3, pp. 309–330, 1999.

[9] L. Vandenberghe, S. Boyd, and S.-P. Wu, "Determinant maximization with linear matrix inequality constraints," *SIAM journal on matrix analysis and applications*, vol. 19, no. 2, pp. 499–533, 1998.

[10] E. A. Yildirim, "On the minimum volume covering ellipsoid of ellipsoids," *SIAM Journal on Optimization*, vol. 17, no. 3, pp. 621–641, 2006.

[11] 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.

[12] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences (Calcutta)*, vol. 2, pp. 49–55, 1936.

[13] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000.

[14] B. Gärtner and S. Schönherr, "Smallest enclosing ellipses–fast and exact," 1997.

[15] S. Nagavalli, A. Lybarger, L. Luo, N. Chakraborty, and K. Sycara, "Aligning coordinate frames in multi-robot systems with relative sensing information," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 388–395.