

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.0092316

# Dynamic Elliptical Shaping Control for Swarm Robots

SHAE T. HART<sup>1</sup>, (Student Member, IEEE), JAKE KAMENETSKY<sup>1</sup>, and Christopher A. Kitts<sup>1</sup>, (Senior Member, IEEE)

<sup>1</sup>Santa Clara University, Santa Clara, CA 95053, USA

Corresponding author: Shae T. Hart (e-mail: sthart@scu.edu).

**ABSTRACT** Solving the robotic swarm coverage problem for an elliptical area has various applications for exploring novel environments. Solutions for this problem should cover a specified ellipse and seamlessly adapt to changing numbers of robots. Previous solutions used techniques such as formation control, vector fields, and neural networks. While these techniques were successful, they all lacked one or more of the three key tenants of swarm elliptical attraction: complete coverage of an ellipse with commandable parameters, simplicity for scaling in the number of robots, and adaptive sizing. Additionally, no previous work presented guidelines for ensuring that the swarms could successfully and safely converge to the commanded ellipse without collisions. In contrast, this work presents a novel swarm elliptical attraction behavior with all three key tenants with guidelines for ellipse and swarm parameter selection. First, a new Lyapunov stable elliptical attraction behavior for Reactive Particle Swarms is presented. The behavior commands robots to cover the entire ellipse area for a specific semimajor axis, eccentricity, and orientation. Additionally, dynamic interagent spacing naturally ensures coverage for different numbers of robots. Second, the work presents a novel adaptive sizing algorithm that varies the ellipse's semimajor axis based on the swarm state. The adaptive sizing algorithm specifies the eccentricity and orientation using time-varying functions. Third, guidelines for selecting the number of robots, commanded ellipse area, obstacle avoidance distance, and robot communication range that allow for successful aggregation to the commanded ellipse are presented. All three of the results are verified using simulation and hardware-in-the-loop trials.

**INDEX TERMS** coverage problem, elliptical aggregation, shape control, swarm robotics.

## NOMENCLATURE

ACR	Artificial Communication Range.
CM	Center of Mass.
$d_{ij}$	Distance between robot $i$ and $j$ .
$d_l$	Specified ACR for trial.
$d_{OA}$	Specified obstacle avoidance range.
EQ	Ellipse adherence parameter.
$f_b$	Weighting function.
$N$	Number of robots in the swarm.
$k_b$	Base behavior scaling constant.
$k_u$	External control input scaling constant.
PE	Percent error.
RPS	Reactive Particle Swarm.
$\vec{u}_i$	External control input for robot $i$ .
$\vec{v}_{bi}$	Base behavior command velocity for robot $i$ .
$\vec{v}_{ci}$	Composite command velocity for robot $i$ .
$W_i(f_b)$	Weighting matrix defined by function $f_b$ .
$\vec{x}_i$	Pose of robot $i$ .

## I. INTRODUCTION

SWARM robotics uses the decentralized control of multiple agents to perform global behaviors through local individual decisions and are characterized by their simplicity, the interchangeability of agents, and the ability to scale in number [1]. There are a variety of subcategories of swarms, including leader-follower [2], consensus [3], [4], Particle Swarm Optimization [5], and Reactive Particle Swarms (RPS). This work uses RPS for its simplicity and reactivity to the environment and the swarm state. The RPS Control Architecture was developed in Ref. [6].

Multi-agent robotic systems naturally lend themselves to the distribution problem, which attempts to space robots throughout a specific area. Various swarm robotic solutions have been developed for these problems, from simple nearest neighbor dispersal [7] to artificial potential fields with formation control [8] to neural networks [9]. Additionally, intricate shapes beyond the traditional simple geometric shapes can

be covered using bounding functions and superposition [10], [11].

An exciting subset of the distribution problem is the elliptical shaping problem. There are three key tenants to swarm elliptical attraction. First, the swarm should cover the entire elliptical area with specifiable ellipse parameters of semi-major axis, eccentricity, and orientation. This includes the perimeter of the ellipse and the ends when highly eccentric. Second, as with all swarm behaviors, the algorithm should be simple enough to scale to a high number of robots while also dynamically handling the addition or loss of robots. Third, the algorithm should be able to size the commanded ellipse adaptively, responding to the swarm state and the environment [8], [9]. Elliptical shaping can benefit the exploration of novel environments, including giving better sensor measurements across features of interest and adaptive sizing allows the swarm to scale and rotate to avoid obstacles [12].

Previous work demonstrated swarm attraction to a virtual elliptical area with set parameters and inter-agent positions set using explicit formation control [8], [13]. Using formation control in this manner is challenging because the formation needs to be recalculated every time the ellipse parameters or the number of robots in the swarm change. Other elliptical attraction behaviors use the superposition of vector fields to attract the robots and distribute them throughout the ellipse [14], [15]. While these methods have controllable ellipse parameters, they do not allow for adaptive sizing, and some use complex limit functions to force robots to distribute throughout the ellipse.

Some work has shown varying the eccentricity of the commanded ellipse using artificial potential fields to navigate obstacles in a workspace [16]. This work has coverage around the circumference of the ellipse, not the interior region. Other work solves the estimation and formation control problem when communication ranges are very limited [17]. While this work solves the harder problem of estimation in addition to elliptical shaping, it has only been shown for a small number of robots and was unable to achieve good coverage of the ellipse area.

Ref. [18] presents a discrete-time kinematic model for aggregation in a multi-dimensional space. Their algorithm bounds the swarm to elliptical regions in 2D and ellipsoids in 3D. Within the commandable bounding region, the swarm only preserves inter-agent connectivity and does not cover the region. Ref [9] uses a neural network to change the shape of the swarm to pass through narrow channels. However, the algorithm does not allow specific commanding of the ellipse parameters and does not have good coverage of the ends of the ellipse for highly eccentric shapes [9]. Voronoi Tessellations are another popular solution but can be slow to respond to changing ellipse parameters [19]. The slow convergence issue was improved using feedforward action but is still relatively computationally extensive for large numbers of robots [20].

Ref. [12] is the most complete elliptical distribution algorithm in literature, allowing for scaling and rotations of

a specified ellipse. The inter-robot distance is explicitly set and scaled with the ellipse to ensure decent coverage of the ellipse area for that number of robots. Because this distance is constant under scaling, it does not guarantee that the swarm will completely cover the ellipse, especially when robots are added or removed. Additionally, Ref. [12] does not provide guidelines for selecting the inter-robot distance.

Overall, previous methods can elliptically attract the swarm. However, none have all the desired elliptical attraction functions of commandable parameters, simplicity for scaling in the number of robots, and adaptive sizing. We chose to use the RPS architecture to develop an elliptical attraction function that meets the three tenants of elliptical attraction. This choice was motivated by the RPS architecture's natural handling of changes in the number of swarm agents and the emphasis on simple, scalable algorithms. Additionally, the RPS architecture is good at dynamic control, which should yield quick performance when tracking changing ellipse parameters. The proposed elliptical attraction function provides complete coverage of the ellipse area with specified parameters. The commanded ellipse can adaptively size, and the robot density is naturally managed to accommodate changing ellipse parameters and variations in the number of robots. Additionally, our elliptical attraction function ensures coverage of the entire ellipse, including the circumference.

The obstacle avoidance range must be preserved to avoid interagent collisions. However, it might be impossible to adhere to that obstacle avoidance range while fitting the swarm in the commanded ellipse. None of the previously mentioned works provide guidelines for ensuring that the swarm can fit in that desired ellipse which is vital for the widespread implementation of these algorithms.

This paper has three major contributions. First, the work presents a novel Lyapunov stable RPS elliptical attraction behavior with coverage of the ellipse area and circumference and adheres well to the commanded ellipse parameters. Second, a new adaptive sizing algorithm for the elliptical attraction behavior is presented. The algorithm scales the semi-major axis as a function of the swarm state while ensuring successful elliptical aggregation. Third, novel guidelines for swarm and ellipse parameter selection are outlined to ensure successful elliptical attraction. These guidelines relate the ellipse area, communication range, obstacle avoidance range, and the number of robots for successful aggregation. All results are verified using simulated and hardware-in-the-loop case studies.

The remainder of the paper is subdivided as follows. Section II summarizes the RPS architecture. Sections III and IV respectively present the RPS elliptical attraction behavior and the adaptive sizing algorithm. Section V outlines the design processes for the simulation and hardware case studies. Then Section VI outlines guidelines for swarm and ellipse parameter selection to ensure successful ellipse aggregation. Finally, Sections VII and VIII present a series of hardware verification trials for the elliptical attraction behavior and adaptive sizing algorithm.

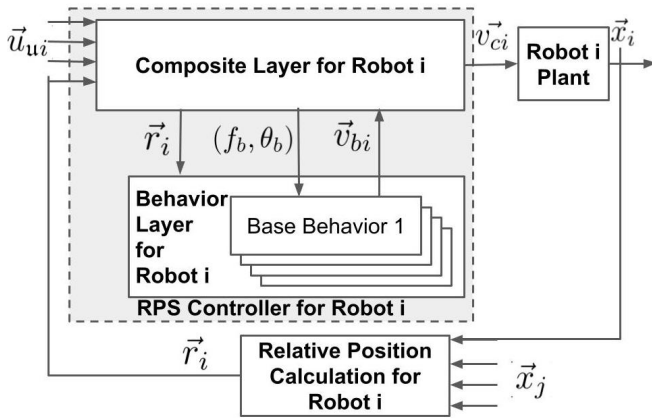


FIGURE 1. Block Diagram showing the RPS control architecture in gray

## II. RPS ARCHITECTURE SUMMARY

This paper uses the planar Reactive Particle Swarm (RPS) control architecture to design an elliptical attraction behavior. RPSs are homogenous swarms with scalable, lightweight, and reactive behaviors that do not require global knowledge, use consensus filtering, or have a hierarchy of robots. RPSs respond to the current swarm state with simple algorithms allowing for reactive control. RPS have broad applications, including various aggregation behaviors, collision avoidance, coverage, formation control, and gradient estimation [6], [21]. The RPS control architecture is a unifying control architecture that allows for the rapid development of novel swarm behaviors. The RPS architecture is summarized in Ref. [6], and Fig. 1 shows the architecture block diagram. In RPSs, there are two layers of control, the upper composite behavior layer and the lower base behavior layer. The composite behavior command velocity for robot  $i$ ,  $\vec{v}_{ci}$ , is a weighted sum of  $p$  base behavior command velocities,  $\vec{v}_{bi}$ , and  $q$  external input velocities  $\vec{u}_{ui}$ , as in (1). The scaling constants  $k$  are constrained as in (2).

$$\vec{v}_{ci} = \sum_{b=1}^p k_b \frac{\vec{v}_{bi}}{\|\vec{v}_{bi}\|} + \sum_{u=1}^q k_u \frac{\vec{u}_{ui}}{\|\vec{u}_{ui}\|} \quad (1)$$

$$\sum_{b=1}^p k_b + \sum_{u=1}^q k_u = 1 \quad k_b > 0 \quad k_u > 0 \quad (2)$$

The  $b^{th}$  base behavior command velocity for robot  $i$ ,  $\vec{v}_{bi}$ , is calculated as in (3) using the pair  $(f_b, \theta_b)$ . The product  $W(f_b)C_i(d_i)D_i\vec{r}_i$  calculates using a reference vector using the weighing function  $f_b$ . Then, the pre-multiplication by  $K(\theta_b)$  rotates the reference vector by  $\theta_b$  about the z-axis to calculate the behavior command velocity.

$$\vec{v}_{bi} = K(\theta_b)W(f_b)C_i(d_i)D_i\vec{r}_i \quad (3)$$

To calculate the reference vector, the relative position vectors pointing from robot  $i$  to all  $N$  robots in the swarm are calculated and vertically augmented together as  $\vec{r}_i \in \mathbb{R}^{2N}$ .

Robots outside the communication range are assumed to be infinitely far away. The matrix  $D_i$  normalizes the individual relative position vectors to relative unit vectors. Normalizing the vectors allows the behavior to be purely determined by the weighting function.

Then  $C_i(d_i)$  uses  $d_i$ , the artificial communication range (ACR), to limit which robots robot  $i$  can use information from. This is done by zeroing the relative unit vectors for robots that are further than  $d_i$  away from robot  $i$ . The ACR should be less than or equal to the actual communication range of the robots. Using an ACR smaller than the actual communication range allows the behavior to both avoid intermittent communication at the limit of the communication range and control the local region of information used.

Next, the weighted sum of the ACR limited relative unit vectors is calculated by multiplying by  $W_i(f_b) \in \mathbb{R}^{2 \times 2N}$ . Here  $f_b(i, j)$  is a weighting function that returns for a pair of robots  $i$  and  $j$  a 2-by-2 weighting matrix. The  $N$  weighting matrices calculated for robot  $i$  are horizontally augmented to form  $W_i(f_b)$ . Using matrices to weight the relative unit vectors allows for more complicated behaviors than simple scalar weighting.

The weighted sum is premultiplied by  $K(\theta_b)$  to determine the velocity for the behavior. Here,  $\theta_b$  is an angle, and  $K$  is the standard rotation matrix about the z-axis. A  $\theta_b = 0$  deg will move the robot along the reference vector, while  $\theta_b = 180$  deg will move the robot along the negative of the reference vector. Other angles of  $\theta_b$  are possible but are not used in this paper. Therefore, the  $b^{th}$  behavior in an RPS composite behavior is specified by the weighting function and angle pair  $(f_b, \theta_b)$ .

Ref. [6] provides a detailed explanation of architecture and equations. It also establishes a library of common RPS behaviors found in literature, including several stable aggregation behaviors, a coverage behavior, a formation maintenance behavior, and several new orbiting behaviors. RPSs are reactive to their environment and the current swarm state. For this reason, the architecture design is very mathematical, avoiding complicated or nested conditionals to allow for easy scaling up and down in the number of robots. The architecture also uses simple functions and superpositions, avoiding long computational delays associated with consensus filtering [4]. The architecture uses only the current swarm state, unlike Particle Swarm Optimization (PSO). PSO saves the past local and global best solutions, which leads to non-reactive control, especially in rapidly changing environments [5].

## III. ELLIPTICAL ATTRACTION ALGORITHM

RPS elliptical attraction is formed using two base behaviors: 1) an elliptical weighting function,  $f_{EQ}$ , and 2) a nearest neighbor dispersion behavior,  $f_{\bar{n}}$ . The following two subsections discuss each base behavior individually before describing the composite behavior.

### A. WEIGHTING FUNCTION: $f_{EQ}$

The characteristic equation of an ellipse is expressed in (4) for points  $(x, y)$  on an ellipse.

$$1 = \frac{\left( (x - x_c) \cos \alpha + (y - y_c) \sin \alpha \right)^2}{a^2} + \frac{\left( -(x - x_c) \sin \alpha + (y - y_c) \cos \alpha \right)^2}{b^2} \quad (4)$$

In (4),  $(x_c, y_c)$  are the coordinates of the center of the ellipse,  $\alpha$  is the angle between the semi-major axis and the x-axis,  $a$  is the semi-major axis, and  $b$  is the semi-minor axis.

For any point  $(x, y)$ , (5) calculates the parameter EQ.

$$EQ(x, y) = \frac{\left( (x - x_c) \cos \alpha + (y - y_c) \sin \alpha \right)^2}{a^2} + \frac{\left( -(x - x_c) \sin \alpha + (y - y_c) \cos \alpha \right)^2}{b^2} \quad (5)$$

An EQ of one indicates that the point is on the ellipse. If EQ is less than one, the point is inside the ellipse, whereas if EQ is greater than one, the point is outside the ellipse. The parameter  $max(EQ)$  indicates how well a swarm has aggregated to the commanded ellipse.  $max(EQ)$  is the maximum EQ value for a robot in the swarm. Since EQ indicates the position of a robot relative to the ellipse,  $max(EQ)$  provides a good indicator if the swarm has not fully spread through the commanded ellipse ( $max(EQ) < 1$ ), has fully expanded to the ellipse ( $max(EQ) \approx 1$ ), or has not contracted to the desired ellipse ( $max(EQ) > 1$ ).

The eccentricity,  $e$ , of an ellipse is related to the semi-major and semi-minor axes as in (6).

$$e = \sqrt{1 - \frac{b^2}{a^2}} \quad (6)$$

Equations (7) and (8) describe the weighting function,  $f_{EQ}$  using ellipse parameters  $a$ ,  $b$ , and  $\alpha$ .

$$f_{EQ}(i, j) = \frac{d_{ij}}{\mathbb{N}} M_{EQ} [EQ > 1] \quad (7)$$

$$M_{EQ} = \begin{bmatrix} \frac{a \sin^2 \alpha + b \cos^2 \alpha}{b} & \cos \alpha \sin \alpha \left( -\frac{a}{b} + \frac{b}{a} \right) \\ \cos \alpha \sin \alpha \left( -\frac{a}{b} + \frac{b}{a} \right) & \frac{a \cos^2 \alpha + b \sin^2 \alpha}{a} \end{bmatrix} \quad (8)$$

Here,  $d_{ij}$  is the distance between robots  $i$  and  $j$ ,  $\mathbb{N}$  is the number of robots within the ACR of robot  $i$ . The center of the ellipse  $\vec{x}_c = (x_c, y_c)$  is the local center of mass (CM) based on robot  $i$ 's view of the swarm,  $\vec{x}_{CM,i}$ . The Iverson bracket notation indicates that the weighting function is only active when the robot is outside the desired ellipse.

The matrix  $M_{EQ}$  was derived using the gradient of (5) as shown in (9). The elliptical attraction function moves the

robot parallel to the negative gradient of the elliptical field centered at  $\vec{x}_c$  with a given eccentricity and orientation.

$$\nabla(EQ) = \frac{2}{ab} M_{EQ} \begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix} \quad (9)$$

### B. WEIGHTING FUNCTION: $f_{\bar{n}}$

The weighting function  $f_{\bar{n}}$  is a nearest  $n$  neighbor weighting function as presented in Ref. [6]. Equation (10) calculates the reference vector as the weighted summation of the relative position vectors pointing from robot  $i$  to its  $n$  closest neighbors. The distance between robot  $i$  and its  $n^{th}$  closest neighbor is  $d_{\bar{n}}$ . Using  $n = 1$  has been shown to distribute robots through an area well [6].

$$f_{\bar{n}}(i, j) = d_{ij} I_2 [d_{ij} \leq d_{\bar{n}}] \quad (10)$$

### C. ELLIPTICAL ATTRACT COMPOSITE BEHAVIOR

RPS elliptical attraction is formed using two base behaviors: 1) an elliptical attraction with  $(f_{EQ}, 0 \text{ deg})$  with desired ellipse parameters,  $a$ ,  $\alpha$ , and  $b$  and 2) the nearest neighbor dispersion  $(f_{\bar{n}}, 180 \text{ deg})$  with  $n = 1$ . The elliptical attraction behavior will bring the robots toward the desired ellipse. Once the robot is on or inside the ellipse, the elliptical attraction behavior turns off, and the nearest neighbor dispersion will distribute the robots throughout the ellipse. Intuition says that the scaling constant for elliptical attraction needs to be greater than that for the dispersion behavior to stabilize the composite behavior. Section VI-A discusses the selection of scaling constants in more detail.

### D. LYAPUNOV STABILITY ANALYSIS

The stability proof of this elliptical attraction behavior is broken into three parts. First, the stability of a single generic behavior with the center of mass as an equilibrium point is proven. Second, the stability of a composite behavior incorporating the CM stable behavior is proven. Finally, these results are applied to the elliptical attraction behavior presented in this work to show that it is Lyapunov Stable.

#### 1) Center of Mass Behaviors

Consider a single RPS base behavior of the form  $f_b(i, j) = M \frac{d_{ij}}{\mathbb{N}}$  where  $M \in \mathbb{R}^{2 \times 2}$  is a constant matrix. Let  $\theta_b = 0$ , making  $K(\theta_b) = I_2$ . For simplicity, assume all robots are within  $d_l$  of all other robots, so  $\mathbb{N} = N$  and  $C_i(d_l) = I_{2N}$ . So the behavior command velocity becomes

$$\vec{v}_{bi} = W(f_b) D_i \vec{r}_i \quad (11)$$

$$= \sum_{j=1}^N f_b(i, j) \frac{1}{d_{ij}} (\vec{x}_j - \vec{x}_i) \quad (12)$$



Combining this with the assumed general form of  $f_b$ , the command velocity simplifies to the following where  $\vec{x}_{CM}$  is the center-of-mass position vector.

$$\vec{v}_{bi} = \sum_{j=1}^N \frac{1}{N} M(\vec{x}_j - \vec{x}_i) \quad (13)$$

$$= M(\vec{x}_{CM} - \vec{x}_i) \quad (14)$$

Equation (1) calculates the robot command velocity for a single RPS base behavior with no external inputs.

$$\dot{\vec{x}}_i = \vec{v}_{ci} = \hat{u}_{bi} = \frac{M(\vec{x}_{CM} - \vec{x}_i)}{\|M(\vec{x}_{CM} - \vec{x}_i)\|} \quad (15)$$

Clearly, if  $\vec{x}_i = \vec{x}_{CM}$  then  $\dot{\vec{x}}_i = \vec{0}$ . This is true for all robots in the swarm, indicating that  $\vec{x}_{CM}$  is an equilibrium point.

Next, let us look at the stability of this behavior. To begin, we use the coordinate transform in (16) to express the positive definite Lyapunov candidate function in (18).

$$\vec{y}_i = \vec{x}_{CM} - \vec{x}_i \quad (16)$$

$$\dot{\vec{y}}_i = \dot{\vec{x}}_{CM} - \dot{\vec{x}}_i \quad (17)$$

$$V = \frac{1}{2} \sum_{i=1}^N \vec{y}_i^T \vec{y}_i \quad (18)$$

then

$$\dot{V} = \sum_{i=1}^N \vec{y}_i^T \dot{\vec{y}}_i \quad (19)$$

$$= \sum_{i=1}^N (\vec{x}_{CM} - \vec{x}_i)^T (\dot{\vec{x}}_{CM} - \dot{\vec{x}}_i) \quad (20)$$

$$= \sum_{i=1}^N \left( (\vec{x}_{CM} - \vec{x}_i)^T \dot{\vec{x}}_{CM} - (\vec{x}_{CM} - \vec{x}_i)^T \dot{\vec{x}}_i \right) \quad (21)$$

$$= \sum_{i=1}^N (\vec{x}_{CM} - \vec{x}_i)^T \dot{\vec{x}}_{CM} - \sum_{i=1}^N (\vec{x}_{CM} - \vec{x}_i)^T \dot{\vec{x}}_i \quad (22)$$

Since  $\dot{\vec{x}}_{CM}$  is independent of the summation, it can be moved outside the summation.

$$\dot{V} = \dot{\vec{x}}_{CM}^T \sum_{i=1}^N (\vec{x}_{CM} - \vec{x}_i) - \sum_{i=1}^N (\vec{x}_{CM} - \vec{x}_i)^T \dot{\vec{x}}_i \quad (23)$$

$$= N \dot{\vec{x}}_{CM}^T (\vec{x}_{CM} - \vec{x}_{CM}) - \sum_{i=1}^N (\vec{x}_{CM} - \vec{x}_i)^T \dot{\vec{x}}_i \quad (24)$$

$$= - \sum_{i=1}^N (\vec{x}_{CM} - \vec{x}_i)^T \dot{\vec{x}}_i \quad (25)$$

Substituting for  $\dot{\vec{x}}_i$ , the time derivative becomes

$$\dot{V} = - \sum_{i=1}^N (\vec{x}_{CM} - \vec{x}_i)^T \frac{M(\vec{x}_{CM} - \vec{x}_i)}{\|M(\vec{x}_{CM} - \vec{x}_i)\|} \quad (26)$$

Therefore, as long as  $M$  is a positive definite matrix, then  $\dot{V}$  is less than zero, implying that the equilibrium point,  $\vec{x}_{CM}$ , is Lyapunov stable for this behavior.

## 2) Composite Behavior with CM Stable Behavior

Next, we consider how this changes if the stable behavior is one of  $p$  base behaviors in an RPS composite behavior. There are also  $q$  external inputs. For simplicity, call the CM stable behavior  $b = 1$ , with a base behavior command velocity unit vector of  $\hat{u}_{1i}$ .

In the worst-case scenario, all other  $p + q - 1$  behaviors are unstable, working against behavior  $b = 1$ . Then for the remaining  $p - 1$  base behaviors with  $b \neq 1$ ,  $\hat{u}_{bi} = -\hat{u}_{1i}$ . Additionally, for all  $q$  external inputs  $\hat{u}_{ui} = -\hat{u}_{1i}$ . The robot command velocity becomes

$$\dot{\vec{x}}_i = \vec{v}_{ci} = k_1 \hat{u}_{1i} - \sum_{b=2}^p k_b \hat{u}_{1i} - \sum_{u=1}^q k_u \hat{u}_{1i} \quad (27)$$

$$= k_1 \hat{u}_{1i} - \left( \sum_{b=2}^p k_b + \sum_{u=1}^q k_u \right) \hat{u}_{1i} \quad (28)$$

Applying the scaling constant constraint equation

$$\sum_{b=1}^p k_b + \sum_{u=1}^q k_u = 1 \quad (29)$$

$$\sum_{b=2}^p k_b + \sum_{u=1}^q k_u = 1 - k_1 \quad (30)$$

Then

$$\dot{\vec{x}}_i = (2k_1 - 1) \hat{u}_{1i} \quad (31)$$

Now that we have derived an equation for  $\dot{\vec{x}}_i$ , let us consider the equilibrium points. We know that  $\hat{u}_{1i} = \frac{\vec{v}_{1i}}{\|\vec{v}_{1i}\|}$ . Combining this, (3), and (31), it is clear that since  $\vec{x}_{CM}$  was an equilibrium point for the base behavior, it is also an equilibrium point of the composite behavior.

Using the same Lyapunov candidate function as before,

$$V = \sum_{i=1}^N \vec{y}_i^T \vec{y}_i \quad (32)$$

$$\dot{V} = \sum_{i=1}^N \vec{y}_i^T \dot{\vec{y}}_i \quad (33)$$

$$= - \sum_{i=1}^N (\vec{x}_{CM} - \vec{x}_i)^T \dot{\vec{x}}_i \quad (34)$$

Substituting in for the robot velocity, the time derivative of the Lyapunov candidate function becomes

$$\dot{V} = -(2k_1 - 1) \sum_{i=1}^N (\vec{x}_{CM} - \vec{x}_i)^T \frac{M(\vec{x}_{CM} - \vec{x}_i)}{\|M(\vec{x}_{CM} - \vec{x}_i)\|} \quad (35)$$

We have already shown that  $M$  must be a positive definite matrix for behavior  $b = 1$  to be Lyapunov stable. So for  $\dot{V}$  to be less than zero,  $k_1$  must be greater than 0.5.

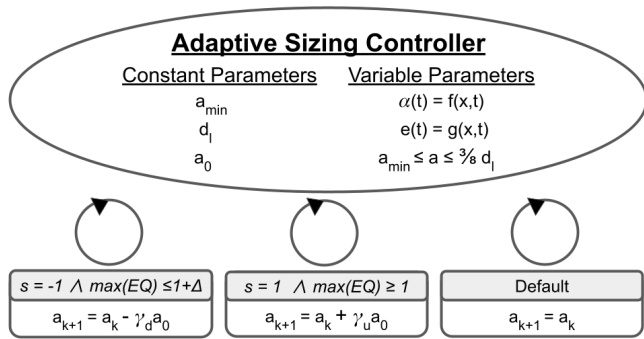


FIGURE 2. Adaptive sizing state machine for elliptical attraction.

In conclusion, a CM-stable RPS behavior causes a composite RPS behavior to be stable if and only if its corresponding scaling constant is greater than 0.5.

### 3) Stability of $f_{EQ}$

Now that we know that weighting functions of the form  $f_b(i, j) = M \frac{d_{ij}}{N}$  are stable for positive definite matrices  $M$ , we must show that  $M_{EQ}$  is positive definite. Since  $M_{EQ}$  is a symmetric matrix and the eigenvalues  $b/a$  and  $a/b$  are both positive,  $M_{EQ}$  is a positive definite matrix. Therefore, the proposed elliptical attraction composite behavior is Lyapunov stable if  $k_{b\_EQ}$  is greater than 0.5.

## IV. ELLIPTICAL ATTRACTION ADAPTIVE SIZING

Elliptically attracting is useful, but *a priori* knowledge of the perfect size for the ellipse might be challenging. Instead, it would be helpful to adaptively size the ellipse based on the swarm state. This version of adaptive sizing scales the commanded semi-major axis while allowing the eccentricity and orientation of the ellipse to be specified as functions with respect to time.

Adaptive sizing can be done using the state machine in Fig. 2. The state machine uses a given minimum semi-major axis  $a_{min}$  and an ACR, and the original commanded semi-major axis  $a_0$  as well as functions  $f(x, t)$  and  $g(x, t)$ . The function  $f(x, t)$  calculates the commanded ellipse orientation  $\alpha$  while  $g(x, t)$  calculates the ellipse's eccentricity. Both  $f$  and  $g$  are functions of the swarm state and time.

As the state machine scales the semi-major axis, the value is clamped to be within  $a_{min}$  and  $\frac{3}{8} d_l$ . The lower bound  $a_{min}$  is an engineering choice based on the application. The upper bound prevents the swarm from expanding out too far, losing communication between robots, and breaking the elliptical attraction behavior, as shown in Sec. VI-C1.

The state machine uses a function  $s$  that can be a function of variables such as swarm state or time. The ellipse should shrink if  $s$  is -1 and expand if  $s$  is 1. There are three transitions for the state machine based on  $s$  and the  $\max(EQ)$  of the swarm. First, if  $s$  is -1, the swarm should only scale down if the swarm has reached the currently commanded ellipse indicated by having a  $(\max(EQ) - 1)$  less than or equal to

some tolerance,  $\Delta$ . When the robots have fully attracted to the ellipse, the obstacle avoidance or the nearest neighbor dispersion may push robots outside the ellipse for a time step.  $\Delta$  should be selected to provide a buffer for when this happens. If these two conditions are satisfied, the new semi-major axis is  $a_{k+1} = a_k - \gamma_d a_0$ . Second, if  $s$  is 1, then the swarm should scale up, with  $a_{k+1} = a_k + \gamma_u a_0$ , only if it has  $\max(EQ)$  greater than or equal to one. This condition prevents the swarm from scaling up on successive scaling up steps before the robots have fully expanded into the currently commanded ellipse.  $\gamma_d$  and  $\gamma_u$  are positive constants representing the fraction of the original semi-major axis the algorithm should scale by on each step. Third, if neither of the previous two cases is met, the state machine does not scale the semi-major axis.

## V. CASE STUDY DESIGN

Next, a series of simulated and on-hardware case studies explore more details of the elliptical attract and adaptive sizing algorithms. All case studies were performed using Santa Clara University's Swarm Behavior Simulator (SBS) [22]. SBS is a MATLAB and Simulink-based platform capable of performing simulated and hardware-in-the-loop trials. The hardware trials connect to Santa Clara University's Decabot Testbed, a 3.5 m by 8.25 m workspace that localizes robots using an OptiTrack system [23]. The testbed commands up to 12 custom holonomic Decabots, as seen in Fig. 3. Fig. 4 shows the Decabots in a portion of the testbed. These holonomic robots allow for testing behaviors on hardware without having to deal with non-linear robot dynamics, as is customary with RPS. SBS also performs two types of simulations, single and batch simulations with randomized initial conditions, using simulated Decabot dynamics. These batch simulations allow for the comparison of swarm performance for different parameter sets, as seen later in this paper.

SBS uses an obstacle avoidance function independent of the RPS architecture to prevent collisions between robots. If two robots are within a set distance of each other,  $d_{OA}$ , then (36) provides the robot command velocity. Otherwise, (1) calculates the robot command velocity.

$$\vec{v}_{OA} = \llbracket d_{ij} \leq d_{OA} \rrbracket \frac{-1}{d_{ij}^4} \vec{r}_{ij} \quad (36)$$

In order to determine how well the robots attract to the commanded ellipse, two performance parameters are used 1)  $\max(EQ)$ , as discussed previously, and 2) the ellipse parameters and associated area for the final positions of the robots. The area and ellipse parameters are calculated using the Khachiyan Algorithm for prescribing a minimum volume enclosing ellipsoid [24]. The code to calculate the ellipsoid is adapted from the MATLAB File Exchange code in Ref. [25].

This paper uses two statistical tests to compare batches of simulations. The first method is the one-way ANOVA. The one-way ANOVA analyzes the difference between the means of more than two groups for one independent variable. The one-way ANOVA null hypothesis is that there is no

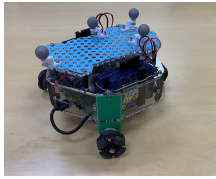


FIGURE 3. A Decabot used for hardware trials.

difference among the group means. This test returns a  $p$ -value. If the  $p$ -value is less than the standard threshold of 0.05, the null hypothesis is rejected, indicating that at least one mean is different, and a post-hoc analysis is needed. The Tukey-Kramer post-hoc test will determine which groups have different means for both ANOVA methods. The Tukey-Kramer Test determines a  $p$ -value for each pair of groups. The means of the two groups are considered statistically different if  $p < 0.05$ . For all analyses, outliers were defined as 1.5 times the interquartile range above the 75<sup>th</sup> percentile or below the 25<sup>th</sup> percentile [26].

## VI. GUIDELINES FOR SUCCESSFUL SWARM ELLIPTICAL AGGREGATION

### A. ADEQUATE SELECTION OF SCALING CONSTANTS

First, we look at selecting the scaling constants,  $k_{b\_EQ}$  and  $k_{b\_n}$ . The stability analysis in Section III-D showed that selecting  $k_{b\_EQ}$  greater than 0.5 should yield a stable behavior. Various numbers of robots were simulated attracting to an ellipse with an eccentricity of 0.9 and a semi-major axis of 10 m. The value of  $k_{b\_EQ}$  was varied from 0.2 to 0.9. The complete set of parameters is specified in Table 1. For a given number of robots and value of  $k_{b\_EQ}$ , 100 simulations were run, and a  $max(EQ)$  was determined for each simulation.

First, Fig. 5 shows the  $max(EQ)$  for each pair of number of robots and  $k_{b\_EQ}$ . The most prominent factor influencing  $max(EQ)$  is  $k_{b\_EQ}$ . To further investigate this, Fig. 6 shows the distribution of  $max(EQ)$  for each value of  $k_{b\_EQ}$ . Values of  $k_{b\_EQ}$  less than 0.5 result in large  $max(EQ)$  values due to the instability of the behavior. When  $k_{b\_EQ}$  is 0.5,  $max(EQ)$  is still relatively large compared to when  $k_{b\_EQ}$  is greater than 0.6. These results agree with the stability analysis in Section III-D.

Next, we examine for  $k_{b\_ellipse} > 0.5$  whether increasing the scaling constant value yields better aggregation to the commanded ellipse. The mean  $max(EQ)$  for all distributions is less than 1.1, indicating good adherence to the ellipse. A one-way ANOVA analysis determined that no mean differed from the others ( $f(3) = 2.568, p = 0.0529$ ). Therefore, increasing  $k_{b\_EQ}$  for pure elliptical attraction does not improve performance. Therefore, the remainder of the case studies use a  $k_{b\_EQ}$  value of 0.9.

### B. DEMONSTRATION OF PERFORMANCE

The next series of simulations demonstrate that the commanded ellipse is reached for a range of angles, eccentricities, and semi-major axes. The parameters for the simulations are

Parameter	Value
$k_{b\_EQ}$	0.2 - 0.9
Semi-Major Axis (m)	10.0
Eccentricity	0.9
Angle $\alpha$ (deg)	0
Avoidance Range (m)	1
ACR (m)	1000
Number of Robots	5 - 20
Robot Speed (m/s)	0.1
Simulation Run Time (s)	200
Number of Trials per Case	100
Initial Condition Area ( $m^2$ )	20 x 20

TABLE 1. Parameter Summary for Scaling Constant Analysis

Parameter	Value
$k_{b\_EQ}$	0.9
Semi-major Axis (m)	5 - 50
Eccentricity	0 - 0.95
Angle $\alpha$ (deg)	0 - 135
Avoidance Range (m)	0.25
ACR (m)	1000
Number of Robots	25
Robot Speed (m/s)	0.1
Simulation Run Time (s)	200 - 750
Number of Trials per Case	100
Initial Condition Area ( $m^2$ )	10 x 10 or 40 x 40

TABLE 2. Parameter Summary for Demonstration of Performance

summarized in Table. 2. One hundred simulations were run for each set of angles, semi-major axes, and eccentricities. Simulations with commanded semi-major axes of 25 and 50 used runtimes of 500 and 700 seconds, respectively, to provide ample time for the robots to disperse through the ellipse. The robots started with random initial conditions in a 10 m by 10 m area for smaller semi-major axes and in a 40 m by 40 m area for semi-major axes of 25 and 50. In order to remove the division by zero when calculating the percent error when  $\alpha = 0$ , the equivalent  $\alpha$  of 180 deg is used.

First, Table. 3 calculates the mean and standard deviation for the distributions for each of the commanded semi-major axes. These results show good adherence between the commanded and calculated semi-major axes with a percent error (PE) of the mean less than 0.1% from the commanded. The mean semi-major axis is expected to be lower than the commanded one because of the elliptical weighting function pushing robots in and the bounding ellipse calculating the tightest ellipse possible.

Second, Table 4 shows the mean and standard deviations for the five eccentricities. There is good adherence between

Semi-major axis (m)	5	10	25	50
Mean (m)	5.000	9.990	24.976	49.951
Standard Deviation (m)	0.005	0.006	0.017	0.062
Percent Error of Mean (%)	0.00	-0.10	-0.10	-0.10

TABLE 3. Mean and Standard deviations of semi-major axis distributions

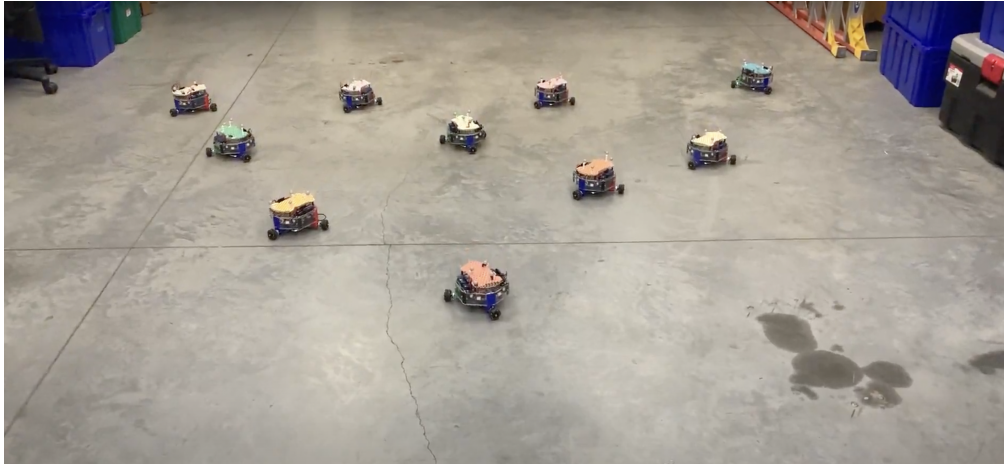


FIGURE 4. Decabots in the Decabot Testbed

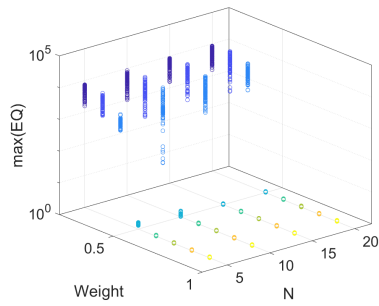


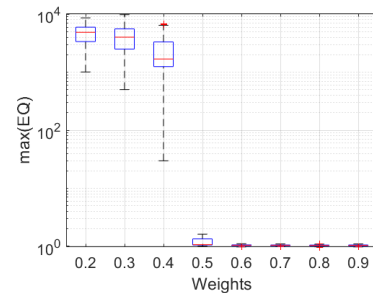
FIGURE 5. Max(EQ) for each pair of scaling constant ( $k_{b\_EQ}$ ) and number of robots.

Eccentricity	0.1	0.25	0.5	0.75	0.95
Mean	0.101	0.250	0.500	0.750	0.950
Standard Deviation	0.014	0.005	0.002	0.001	0.001
Percent Error of Mean (%)	1.00	0.00	0.00	0.00	0.00

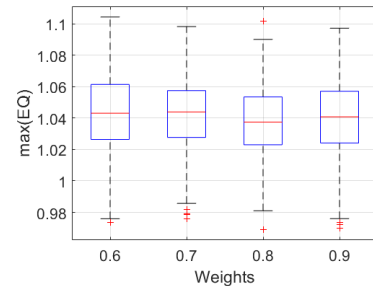
TABLE 4. Mean and standard deviation of eccentricity distributions

the desired and calculated eccentricities with a percent error in the mean of at most 1%. The standard deviations and percent error of the mean eccentricity are more significant for smaller eccentricities because, for less eccentric shapes, there is more variation in the bounding ellipse used to calculate ellipse parameters due to the robot positions.

Third, let us examine how well the angle of the ellipse tracks the commanded ellipse. For less eccentric ellipses, we expect more error in the angle calculated by the bounding ellipse. Again, this is because the bounding ellipse calculates the tightest ellipse based on the individual robot positions but these low eccentricity ellipses are very circular, making calculating  $\alpha$  more error-prone. For this reason, we determine the mean and standard deviations of the angles separated by commanded angle and eccentricity. Table 5 summarizes each distribution's mean and standard deviation as well as the per-



(a)



(b)

FIGURE 6. Max(EQ) dispersions for attraction scaling constants (a) from 0.2 to 0.9 (b) from 0.6 to 0.9.

cent error in the mean of each distribution. As expected, there were more significant standard deviations and larger percent errors for the calculated angles for less eccentric ellipses. However, overall the mean calculated angles matched the commanded angles to within a degree and with percent errors less than 1%, indicating good adherence to the commanded angle.

Fig. 7 shows the distributions of  $max(EQ)$  for each semi-major axis and eccentricity. The color represents the mean value of  $max(EQ)$  with lighter shades indicating results that adhere less well to the commanded ellipse. The size of



e	Commanded $\alpha$ (deg)											
	0			45			90			135		
	Mean (deg)	SD (deg)	PE	Mean (deg)	SD (deg)	PE	Mean (deg)	SD (deg)	PE	Mean (deg)	SD (deg)	PE
0.1	0.40	5.20	0.22	45.02	5.98	0.04	90.22	5.47	0.24	135.86	6.71	0.64
0.25	0.02	1.00	0.01	44.95	0.86	0.11	89.95	1.09	0.06	135.03	0.84	0.02
0.5	-0.01	0.29	0.01	45.00	0.27	0.00	90.00	0.22	0.00	135.01	0.24	0.01
0.75	0.00	0.07	0.00	45.00	0.07	0.00	90.01	0.08	0.01	135.00	0.07	0.00
0.95	0.00	0.03	0.00	45.00	0.03	0.00	90.00	0.03	0.00	135.00	0.03	0.00

TABLE 5. Mean, standard deviation, and percent error of the mean of calculated angle  $\alpha$  for the commanded angle and eccentricity in degrees.

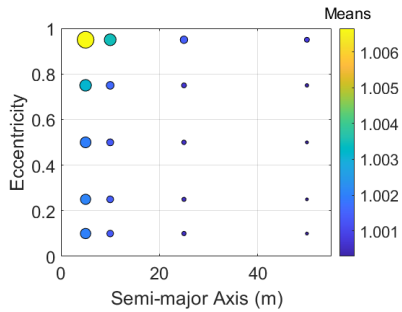


FIGURE 7. For a given semi-major axis and eccentricity, the color of the represents the average  $max(EQ)$ , and the size represents the standard deviation of  $max(EQ)$ .

the circles represents the standard deviation in the values of  $max(EQ)$ . Overall, there is good adherence to the ellipse with a maximum mean  $max(EQ)$  of 1.0067 and a maximum standard deviation of 0.0030. The smaller the semi-major axis and the more eccentric the shape, the higher the mean and SD of  $max(EQ)$ . The reason for this is twofold. First, as the semi-major axis decreases and the eccentricity increases, the area of the commanded ellipse decreases, making it harder for all the robots to fit in the commanded ellipse while honoring the obstacle avoidance range. Second, for these ellipses with smaller commanded areas, (5) gives larger values for the same disturbance outside the ellipse.

Overall, these results indicate that the proposed elliptical attraction function can attract swarms to a wide range of commanded ellipses. For simplicity, since we know the elliptical attraction function tracks the commanded parameters very well, we will often only use the final area and  $max(EQ)$  as performance parameters. The final area indicates how well the commanded semi-major axis and eccentricity were tracked, and  $max(EQ)$  indicates how well the robots cover the commanded area. Additionally, most subsequent analyses use  $\alpha = 0$  since this is just a coordinate rotation.

### C. PREDICTING SWARM SUCCESS FROM COMMANDED PARAMETERS

Next, we investigate how to predict if a swarm can elliptically attract to a region, which depends upon the number of robots, the obstacle avoidance distance, the artificial communication range, and the area of the commanded ellipse.

Parameter	Range of Values
$k_{b\_EQ}$	0.9
Semi-Major Axis (m)	10
Eccentricity	0.95
Angle $\alpha$ (deg)	0
Avoidance Range (m)	0.5
ACR (m)	5, 10, 20
Number of Robots	20
Robot Speed (m/s)	0.1
Simulation Run Time	200 sec
Initial Condition Area ( $m^2$ )	10 x 10

TABLE 6. Parameter Summary for ACR Test

#### 1) Influence of ACR

A swarm's artificial communication range influences its ability to attract to the desired ellipse successfully. For this elliptical attraction behavior, each robot calculates the local CM of the visible robots and uses that as the center of the commanded ellipse. If the ACR is too small, the robots cannot communicate with robots on the other side of the commanded ellipse, and the performance will be degraded, as seen in Fig. 8. Table 6 summarizes the parameters for the three simulation tests. Twenty robots are commanded from the same initial conditions to an ellipse with a semi-major axis of 10 m for three ACR ranges. An ACR of 5 m is too small, and the robots distribute through the space but not elliptically. When the ACR increases to 10 m, the resultant shape is more eccentric, but because the robots on either end can only see half of the ellipse, they do not converge to have the proper semi-major axis. Finally, when the ACR is 20 m, the robots can barely see the ellipse's other end, and the correct semi-major axis and eccentricity are achieved.

To remedy this issue, a good rule of thumb is to only command robots to ellipses with a semi-major axis of at most  $\frac{3}{8}d_l$ . This means that if two robots were perfectly on opposite ends of the ellipse, they would be  $\frac{3}{4}d_l$  away from each other, providing margin.

#### 2) Commanded Area vs. Final Area

Given an obstacle avoidance range and the number of robots, it might be impossible to fit all the robots in the commanded area. Table 7 presents the simulation parameters for this case study. Fig. 9 shows how, for an obstacle avoidance range of 1 m, and different commanded areas, as the number of robots increases, eventually, the robots cannot all fit inside the com-

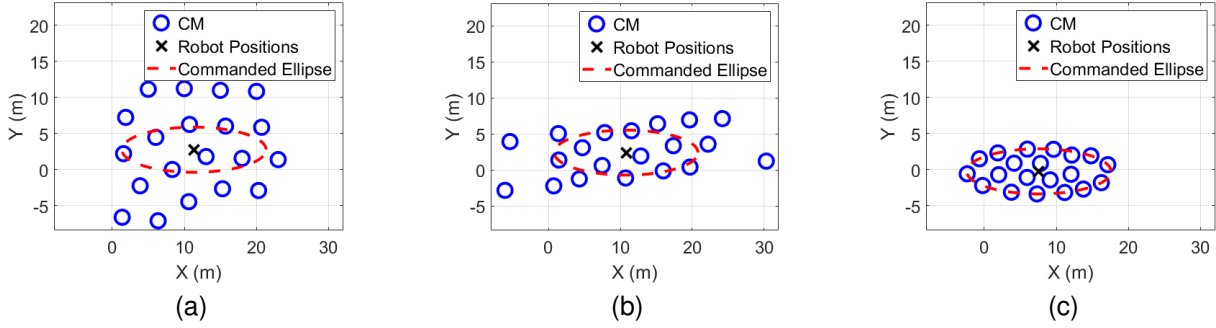


FIGURE 8. Final positions for ACR test with ACRs of (a) 5 m, (b) 10 m, and (c) 20 m.

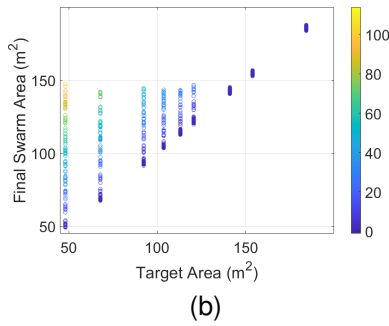
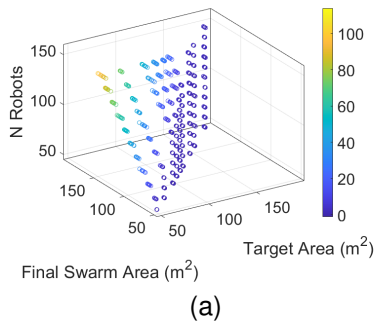


FIGURE 9. Commanded area vs. final area for the different number of robots.

manded ellipse. Interestingly, when the robots fail to attract to the desired ellipse in Fig. 9, the final areas form a plane. This indicates that there could be a relationship between the minimum area a swarm can attract to, the number of robots, and the obstacle avoidance distance. Sec. VI-C3 investigates the relationship between these parameters.

### 3) Relationship Between Minimum Area, $d_{OA}$ , and N

A series of simulations investigate the relationship between the minimum area a swarm can attract to, the obstacle avoidance range  $d_{OA}$ , and the number of robots. Robots were commanded to an ellipse with a semi-major axis of 0.1 m and an eccentricity of 0.9 for multiple obstacle avoidance ranges. This ellipse area is too small for the robots to fit in. Instead, the robots will attempt to minimize the area they

Parameter	Value
$k_{b\_EQ}$	0.9
Semi-Major Axis (m)	6 - 8
Eccentricity	0.00-0.95
Angle $\alpha$ (deg)	0
Avoidance Range (m)	1
ACR (m)	1500
Number of Robots	50 - 150
Robot Speed (m/s)	2
Simulation Run Time (sec)	200
Number of Trials per Case	10
Initial Condition Area ( $m^2$ )	20 x 20

TABLE 7. Parameter Summary for Commanded Area vs. Final Area Case Study

Parameter	Range of Values
$k_{b\_EQ}$	0.9
Semi-Major Axis (m)	6
Eccentricity	0.95
Angle $\alpha$ (deg)	0
Avoidance Range (m)	0.1 - 10
ACR(m)	1500
Number of Robots	10-100
Robot Speed (m/s)	0.1
Simulation Run Time (sec)	200
Number of Trials per case	10
Initial Condition Area ( $m^2$ )	20 x 20

TABLE 8. Parameter Summary for Swarm Area in Failure Case Study

cover while honoring the obstacle avoidance range, providing a reasonable estimate of the minimum area that a number of robots and  $d_{OA}$  can attract. For each  $d_{OA}$  and number of robots, 100 simulations were performed. A summary of the simulation parameters is found in Table. 8

Fig. 10 shows that for a given  $d_{OA}$ , the slope relating the number of robots to the final area is relatively linear. Using a linear best fit, the slope ( $\rho$ ) and offset ( $b$ ) were determined for each  $d_{OA}$  and are summarized in Table. 9. Fig. 11 shows that the density versus  $d_{OA}$  is linear on a log-log scale. Eqn (37) express the linear best-fit equation for the robot density for a given obstacle avoidance range,  $\rho(d_{OA})$ .

$$\rho(d_{OA}) = \frac{1.040}{d_{OA}^{2.014}} \quad (37)$$

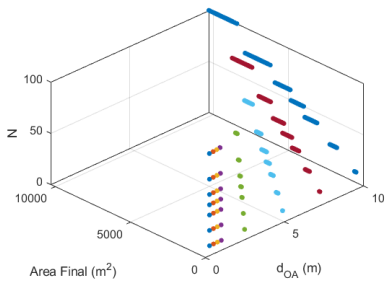


FIGURE 10. Relationship between the final area a swarm can attract to for a given number of robots and ACR.

$d_{OA}$ (m)	$\rho$ (robots/ $m^2$ )	b (robots)
0.25	16.17	3.11
0.50	4.25	4.17
0.75	1.91	4.24
1	1.07	4.47
2.5	0.17	5.56
5	0.04	5.95
7.5	0.02	6.31
10	0.01	5.16

TABLE 9. Slopes and offsets relating final area and number of robots for each  $d_{OA}$

The density is approximately proportional to the inverse square of the obstacle avoidance range, which agrees with intuition. Equation (37) does not have a bias because as the obstacle avoidance range goes to infinity, we expect the robot density to go to zero. Then the number of robots  $N_{est}$  that can fit in some area  $A_{min}$  can be estimated as in (38). The average bias from Table 9 was used to generalize the bias in (38).

$$N_{est} = \rho(d_{OA})A_{min} + 4.87 \quad (38)$$

In Fig. 10, for each  $d_{OA}$  and number of robots, there is a range of areas that the swarms attract to. There is a larger spread for the final area for larger values of  $d_{OA}$  and a higher number of robots. The minimum area that the estimate equations calculate is roughly in the middle of the area distributions due to the use of the linear-best fit. Therefore, the estimate equations will sometimes underestimate the area a swarm can attract to for a given  $d_{OA}$  due to initial conditions and the constant velocity controller. Using the data from Fig. 10, the estimated area underestimated the final area by at most 10% for all cases. Therefore, a margin of 10% should be used when applying the estimate equations right at the boundary of possible ellipses. The estimate equation will also sometimes overestimate the minimum final area for some initial conditions. This is not a concern because if the swarm can fit in a smaller area for a given number and  $d_{OA}$ , the swarm will easily achieve the larger area.

#### D. APPLICATION OF ESTIMATE EQUATION

There are three different ways the above equations can be used: 1) determine how many robots can fit given an area and

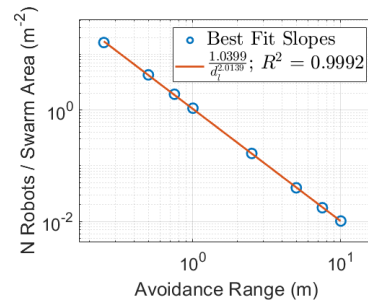


FIGURE 11. Relationship between avoidance range and swarm density.

$d_{OA}$ , 2) determine the largest obstacle avoidance that allows a specified number of robots to fit in a prescribed area, and 3) determine the minimum area a specific number of robots can fit in given  $d_{OA}$ . The following three subsections perform each of these determinations to see how well the equations predicts the results.

1) Application One: Given A and  $d_{OA}$ , find max N

Now that we can estimate the number of robots that can fit into the desired ellipse given  $d_{OA}$ , let us see how it holds up in simulation. Pairs of eccentricities and semi-major axes were always selected to have a commanded area of  $200 m^2$ .

Equation (37) predicts a robot density of  $0.2575 robots/m^2$  for a  $d_{OA}$  of 2 m. Equation (38) estimates that 56.37 robots should fit in the desired ellipse. Table. 10 lists the simulations' full set of parameters. If we include the 10% margin for the area, we reduce the area to  $180 m^2$  so that any underestimation will still allow the swarm to fit in a  $200 m^2$  area. The robot density remains the same, so only 51.22 robots should fit in the commanded area.

Fig. 12 shows the distributions for  $max(EQ)$  for each eccentricity and number of robots. The color of each circle represents the mean of  $max(EQ)$  while the radius represents the standard deviation of  $max(EQ)$ . For less than 65 robots, the  $max(EQ)$  is close to 1, indicating that the desired ellipse was reached. Additionally, there is minimal variation in the standard deviation, indicating that eccentricity did not influence the swarm's ability to attract to the ellipse successfully. The solid and dashed horizontal lines indicate the number of robots possible from the estimate equation without and with margin, respectively. Both numbers of robots are within the region of successful attraction.

At 65 robots, the standard deviation increases, but the mean  $max(EQ)$  is still very close to 1, indicating that the swarm is right on the threshold of fitting in the ellipse. As the number of robots increases over 65, the standard deviations stay relatively constant. However, the mean  $max(EQ)$  steadily increases, indicating that the threshold has been passed and there are too many robots to fit in the ellipse. Overall, this case study demonstrates that (37) and (38) predict the robot density and number of robots well.

Parameter	Application 1	Application 2	Application 3
$k_b_{EQ}$	0.9	0.9	0.9
Semi-Major Axis (m)	8.92 - 21.24	6.91 - 12.37	3.39 - 3.85
Ellipse Eccentricity	0.6 - 0.99	0 - 0.95	0 - 0.95
Area ( $m^2$ )	200	150	36 - 46
Avoidance Range (m)	2	2 - 3.66	2
ACR (m)	1500	50	50
Number of Robots	40 - 80	25	15
Robot Speed (m/s)	0.1	0.1	0.1
Simulation Run Time (sec)	500	200	200
Number of Trials per case	100	100	100
Initial Conditions Area ( $m^2$ )	20 x 20	20 x 20	20 x 20

TABLE 10. Parameter Summary for Swarm Success Threshold Validation Study

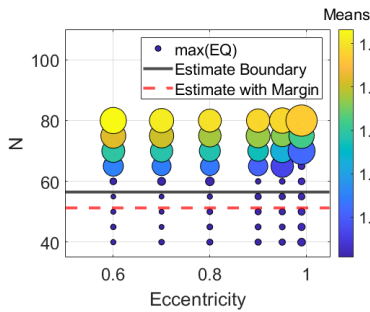


FIGURE 12. For a given eccentricity and number of robots, the mean and standard deviation of  $max(EQ)$

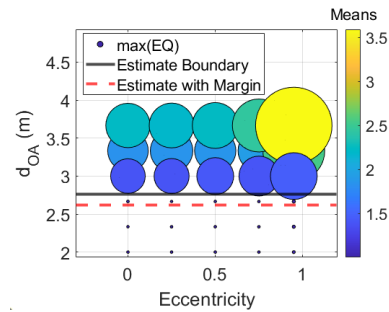


FIGURE 13. For a given eccentricity and obstacle avoidance range, the mean and standard deviation of  $max(EQ)$ .

2) Application Two: Given A and N, find  $d_{OA}$  max

The next application of the estimate equations uses the area of the desired ellipse and the number of robots to determine the greatest obstacle avoidance range. Here 25 robots are commanded to an area of  $150 m^2$ . The equations predict a density of  $0.13 robots/m^2$  and a maximum obstacle avoidance range of 2.76 m. Including the 10% margin on the area, the area is reduced to  $135 m^2$ . The estimated maximum density is then  $0.15 robots/m^2$ , and the maximum obstacle avoidance range is 2.62 m. The simulation parameters are listed in Table 10.

Fig. 13 shows the resulting mean and standard deviations of  $max(EQ)$  for each parameter set. For  $d_{OA}$  less than 3, the mean of  $max(EQ)$  is very close to 1 with little standard deviation. The  $max(EQ)$  means and standard deviations increased as the obstacle avoidance range increased over 3 m. The estimated maximum  $d_{OA}$  without margin, the solid black line in Fig. 13, is right on the threshold of the robots fitting in the ellipse. Using the 10% margin provides some buffer for the estimate equation. Overall, the estimate equations predicted the maximum obstacle avoidance range very well.

3) Application Three: Given  $d_{OA}$  and N, find minimum A

Next, we look at how well the equation predicts the minimum area the robots can fit in for a given obstacle avoidance range and a number of robots. For 15 robots with an obstacle avoidance range of 2 m, the equations predict a robot density of  $0.2575 robots/m^2$  and a minimum area of 39.34 square

meters. This area is increased by 10% to  $43.28 m^2$  to accommodate the margin. Table 10 shows the complete set of simulation parameters.

The simulation results are displayed in Fig. 14. The simulations have large standard deviations for the commanded areas at or below 42 square meters for  $max(EQ)$ . Even though the mean  $max(EQ)$  for a commanded area of  $42 m^2$  is close to 1, the high standard deviation indicates that the swarm is struggling to attract to the correct area for all initial conditions. For commanded areas above  $42 m^2$ , there is a significant decrease in the standard deviation of the ellipse, especially at high eccentricities. Additionally, the mean  $max(EQ)$  becomes less than 1.1, indicating that the robots have converged to the desired ellipse.

The estimate equations predicted that the 15 robots with a  $d_{OA}$  of 2 m could attract to an area of  $39.34 m^2$  and an area of  $43.38 m^2$  with margin. Based on the mean and standard deviation of  $max(EQ)$ , the estimate equation underpredicted the area to which the swarm could aggregate. The 10% margin provided the extra room to ensure that the commanded area was sufficient to allow the swarm to aggregate successfully. In conclusion, the overall equations provided a good prediction of the minimum area.

VII. HARDWARE VERIFICATION OF ELLIPTICAL ATTRACTION BEHAVIOR

The following subsections use a series of hardware-in-the-loop case studies to demonstrate the performance of the elliptical attraction behavior.



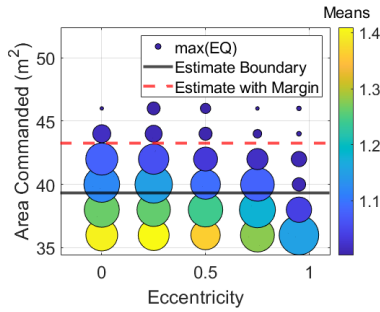


FIGURE 14. For a given eccentricity and commanded area, the mean and standard deviation of  $max(EQ)$

### A. INDIVIDUAL PARAMETER COMMANDING

The previous case studies all used constant ellipse parameters. Additionally, the motion of the CM of the ellipse was not commanded, so the swarm aggregated to an arbitrary position. This case study demonstrates the attraction functions' ability to track the commanded ellipse parameters while undergoing translations, rotations, scaling, and shaping. For each transformation, the swarm will attract to the initial commanded ellipse parameters for 45 seconds. Then the attraction function will track 1) a CM trajectory for translation as in (39), 2) a trajectory in  $\alpha$  for rotation as in (40), 3) a trajectory in  $a$  for scaling as in (41), and 4) changes in eccentricity for shaping as in (42). All other ellipse parameters will be held constant in each stage to demonstrate individual transformations. The translation behavior uses an external input command velocity in addition to the elliptical attract behavior. The trajectory for the CM is expressed in (39), and the external behavior command velocity is expressed in (43). The parameters for the individual ellipse parameter commanding hardware trial are summarized in Table 11.

$$\vec{x}_{cmd}(t) = \begin{cases} \begin{bmatrix} 0.75 \\ 0 \end{bmatrix} & t < 45 \\ \begin{bmatrix} \frac{0.75}{45}(t - 45) + 0.75 \\ 0 \end{bmatrix} & 45 \leq t < 90 \\ \begin{bmatrix} 1.5 \\ 0 \end{bmatrix} & 90 \leq t \end{cases} \quad (39)$$

$$\alpha(t) = \begin{cases} 45 & t < 90 \\ 45 + (t - 90) & 90 \leq t < 135 \\ 90 & 135 \leq t \end{cases} \quad (40)$$

$$a(t) = \begin{cases} 1 & t < 135 \\ 1 + \frac{0.5}{45}(t - 135) & 135 \leq t < 180 \\ 1.5 & 180 \leq t \end{cases} \quad (41)$$

Parameter	Individual	Combined
$[k_{b\_EQ}, k_{b\_n}, k_{traj}]$	[0.9, 0.05, 0.05]	[0.9, 0.05, 0.05]
Semi-Major Axis (m)	(41)	(45)
Eccentricity	(42)	(46)
Angle $\alpha$ (deg)	(40)	(44)
Avoidance Range (m)	0.25	0.25
ACR (m)	10	10
Number of Robots	12	12
Robot Speed (m/s)	0.1	0.1
Simulation Run Time (sec)	225	180
Initial Condition Area ( $m^2$ )	3.5 x 3.5	3.5 x 3.5

TABLE 11. Parameter Summary for Individual and Combined Parameter Commanding Trials

$$e(t) = \begin{cases} 0.9 & t < 180 \\ 0.9 - \frac{0.4}{45}(t - 180) & 180 \leq t < 225 \end{cases} \quad (42)$$

$$\vec{u}_{u\_traj,i} = \vec{x}_{cmd} - \vec{x}_c \quad (43)$$

Fig. 15 shows the individual transformation hardware trial results. The vertical lines in each figure mark the transitions between each transformation. Fig. 15a shows the  $max(EQ)$  for the trial. The swarm stays within the commanded ellipse after the initial transient for the robots to converge to the desired ellipse, even as the ellipse translates or the parameters change. Fig. 15b shows the disturbance magnitude from the commanded CM trajectory. The CM tracks the commanded trajectory well with less than a 1 cm disturbance.

Finally, Fig. 15c shows the percent error in the commanded ellipse parameters for each transformation. The swarm converges to the commanded ellipse after 20 seconds. Then as the commanded ellipse undergoes translation, rotations, and scaling in the subsequent three transformations, the swarm tracks the commanded parameters well, with at most a 3% error in all of the parameters. In the final phase, where the eccentricity changes, there is a spike in the percent error in all ellipse parameters. This is expected because when the robots are inside the commanded ellipse, their relative positions are controlled by the nearest neighbor dispersion algorithm, and it takes time for the robots to distribute through the commanded area. While distributing, the swarm does not track the commanded ellipse parameters. Additionally,  $max(EQ)$  for this translation does not spike, indicating that the swarm is still inside the commanded ellipse. Overall this shows good adherence to the desired ellipse parameters while the swarm undergoes individual transformations of translating, rotating, scaling, and shaping.

### B. COMPOSITE PARAMETER COMMANDING

The previous subsection showed that this attraction function could handle individual translations, rotations, scalings, and shapings. This case study varies all three ellipse parameters simultaneously according to the Eqns. (44)-(46) while also translating as in (47).

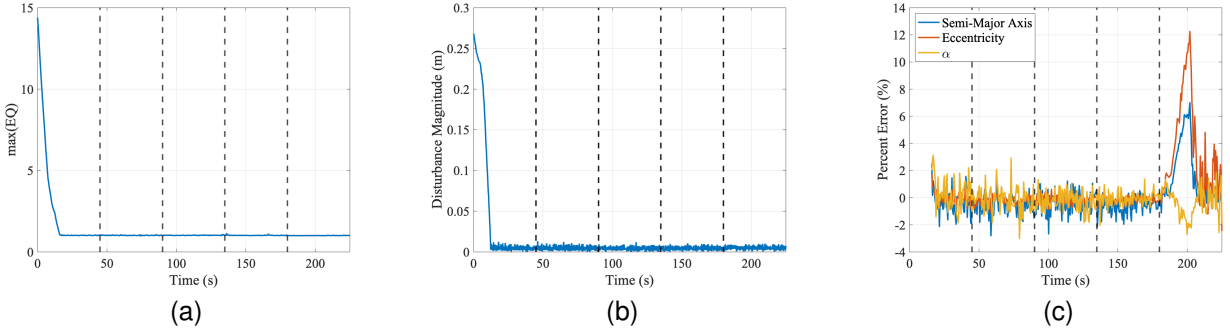


FIGURE 15. Individual parameter commanding case study results. a)  $max(EQ)$ , b) CM Disturbance Magnitude, c) Ellipse Parameter Percent Error

$$\alpha(t) = \begin{cases} 45 & t < 45 \\ 45 + \frac{45}{135}(t - 45) & 45 \leq t \end{cases} \quad (44)$$

$$a(t) = \begin{cases} 1 & t < 45 \\ 1 + \frac{0.5}{135}(t - 45) & 45 \leq t \end{cases} \quad (45)$$

$$e(t) = \begin{cases} 0.9 & t < 45 \\ 0.9 - \frac{0.4}{135}(t - 45) & 45 \leq t \end{cases} \quad (46)$$

$$\vec{x}_{cmd}(t) = \begin{cases} \begin{bmatrix} 0.75 \\ 0 \end{bmatrix} & t < 45 \\ \begin{bmatrix} \frac{0.75}{135}(t - 45) + 0.75 \\ 0 \end{bmatrix} & 45 \leq t \end{cases} \quad (47)$$

The parameters for the trial are summarized in Table. 11 and Fig. 16 shows the results for the case study. The vertical line marks when the ellipse parameters start to change, and the translation begins. Fig. 16a shows that after the swarm settles to the commended ellipse,  $max(EQ)$  stays very close to 1, indicating good adherence to the commanded area. Fig. 16b shows that the swarm CM tracks the commanded trajectory well, with about a 1 cm disturbance. Finally, Fig. 16c shows the percent error in the three ellipse parameters. The semi-major axis percent error after the swarm converges is less than 2%. The percent errors in the angle and eccentricity are less than 4% but grow with time. This is expected because, for lower eccentricities, the bounding ellipse that measures the ellipse parameters is less accurate due to the small number of robots used in the trial. Additionally, for more circular shapes, the angle calculation is more error-prone. The spike in the ellipse parameters percent error that was present during the individual shaping trial is not present here because the elliptical shaping weighting function,  $f_{EQ}$ , is also influencing the swarm motion, not just the nearest-neighbor dispersion,  $f_{\bar{n}}$ . Overall, a percent error of less than 4% for all ellipse parameters while also undergoing translations indicates that this elliptical attraction behavior performs very well.

### C. APPLICATION OF ESTIMATE EQUATION

In Section VI-C3, a relationship between the minimum area of an ellipse, obstacle avoidance distance, and the number of robots was determined. Then, a series of simulations showed that these equations held. Now, the next three subsections demonstrate on hardware that these equations hold.

#### 1) Application One: Given A and $d_{OA}$ , find max N

This case study predicts how many robots can fit in an area given an ACR. The trial parameters are listed in Table 12. The final ellipse parameters, the minimum distance between robots, and  $max(EQ)$  are listed in Table 13. The estimate equations calculate that 10.12 robots should fit into an ellipse with a  $d_{OA}$  of 1.1 m and an area of  $6.1216 m^2$  ( $a = 1.5$  m,  $e = 0.5$ ). Including the 10% margin, only 9.60 robots should fit in the area.

Based on  $max(EQ)$  and the percent error in the commanded ellipse parameters, twelve robots did not fit, and eight robots did. The  $max(EQ)$  for ten robots was 1.24, which indicates that the robots did not all fit in the ellipse. While the original estimate equations predicted that ten robots should have fit in the area, the 10% margin dropped the estimate to below ten robots, so it is not surprising that ten robots did not fit. This shows, especially when using the 10% margin, that the estimate equations can accurately predict the number of robots that can fit in an area.

#### 2) Application Two: Given A and N, find $d_{OA}$ max

This case study tests the largest obstacle avoidance range,  $d_{OA}$ , that allows  $N$  robots to fit in a specified area. The equations predict a maximum  $d_{OA}$  of 1.5 m for 12 robots to fit within an ellipse area of  $15.71 m^2$  ( $a = 2.75$  m,  $e = 0.75$ ). The addition of 10% margin reduces  $d_{OA}$  to at most 1.43 m. The trial parameters are listed in Table 12.

Table 14 lists the final ellipse parameters, minimum distance between robots, and  $max(EQ)$  for each trial. According to  $max(EQ)$  and the final ellipse parameters, commanding an obstacle avoidance range of 1.4 m allows the robots to fit in the area with the correct ellipse parameters while commanding an obstacle avoidance range of 1.6 does not. These results satisfy both the original estimate and the

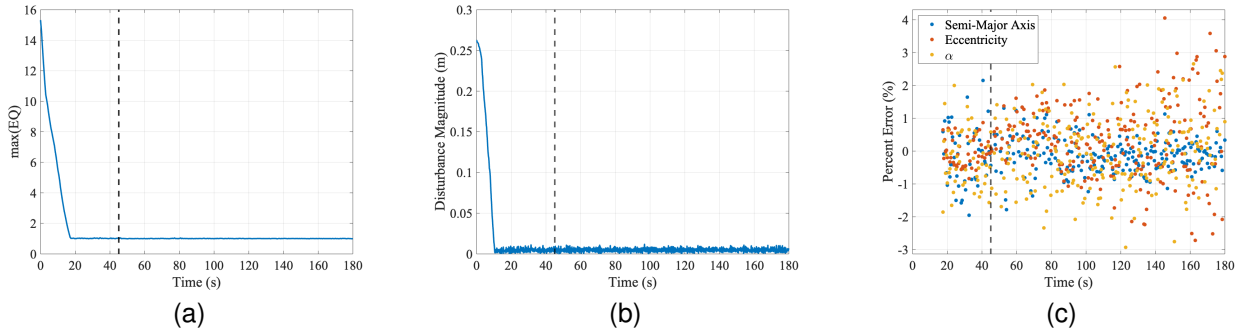


FIGURE 16. Composite parameter commanding case study results. a)  $max(EQ)$ , b) CM Disturbance Magnitude, c) Ellipse Parameter Percent Error

Parameter	Application 1	Application 2	Application 3
$[k_{b\_EQ}, k_{b\_a}]$	[0.9, 0.1]	[0.9, 0.1]	[0.9, 0.1]
Semi-Major Axis (m)	1.5	2.75	1.75, 2.25
Eccentricity	0.5	0.75	0.95
Angle $\alpha$ (deg)	0	0	0
Avoidance Range (m)	1.1	1.4, 1.6	0.75
ACR (m)	10	10	10
Number of Robots	8,10,12	12	12
Robot Speed (m/s)	0.1	0.1	0.1
Simulation Run Time (sec)	60	60	60
Initial Condition Area ( $m^2$ )	6.5 x 3.6	6.5 x 3.6	6.5 x 3.6

TABLE 12. Command Parameter Summary for Estimate Equation Hardware Verification Trials

Final Parameter	N					
	8		10		12	
a (m)	1.50	0.00	1.66	10.67	1.92	28.00
e	0.50	0.00	0.63	26.00	0.60	20.00
$\alpha$ (deg)	0.43	0.43	174.68	5.32	6.05	6.05
$min(d_{ij})$ (m)	1.19	-	1.10	-	1.07	-
$max(EQ)$	1.00	-	1.24	-	1.73	-

TABLE 13. Final Ellipse Parameters for Application 1 Hardware Trial

Final Parameter	Commanded a (m)			
	2.25		1.75	
a (m)	2.24	0.44	1.72	1.71
e	0.95	0.00	0.87	8.42
$\alpha$ (deg)	0.20	0.20	0.75	0.75
$min(d_{ij})(m)$	0.83	-	0.73	-
$max(EQ)$	1.00	-	2.41	-

TABLE 15. Final Ellipse Parameters for Application 3 Hardware Trial

Final Parameter	Robot $d_{OA}$ (m)			
	1.4		1.6	
a (m)	2.75	0.00	3.23	17.45
e	0.75	0.00	0.76	1.33
$\alpha$ (deg)	0.53	0.53	173.94	6.06
$min(d_{ij})$ (m)	1.44	-	1.57	-
$max(EQ)$	1.01	-	1.62	-

TABLE 14. Final Ellipse Parameters for Application 2 Hardware Trial

estimate with margin. This shows that the estimate equations predict the maximum obstacle avoidance range well.

### 3) Application Three: Given $d_{OA}$ and N, find minimum A

This case study determines the minimum area a certain number of robots can fit in given  $d_{OA}$ . For the trial parameters in Table 12, (37) and (38) estimate that 12 robots can attract to a minimum area of  $3.84 m^2$  ( $a = 1.98 m$ ,  $e = 0.95$ ) given an

avoidance range of 0.75 m. Using the 10% margin increases the minimum area to  $4.23 m^2$  ( $a = 2.08 m$ ,  $e = 0.95$ )

We command an area above and below the limit keeping the eccentricity constant. The final ellipse parameters,  $max(EQ)$ , and minimum final distance between robots are listed in Table 15. As expected, commanding a semi-major axis of 2.25 m fits all robots, while commanding a semi-major axis of 1.75 m will not as based on  $max(EQ)$  and the ellipse parameters. These results satisfy both the original equation and the equation with the margin.

## VIII. ADAPTIVE SIZING HARDWARE VERIFICATION

The previous hardware studies had the swarm follow trajectories for the ellipse parameters. Adaptive sizing allows the size of the ellipse to be varied without needing to specify a trajectory. This section tests the proposed adaptive sizing algorithm on hardware using  $s$  as expressed in (48). For the first 45 seconds, if the swarm has aggregated to within a

tolerance of the commanded ellipse for 5 seconds, the ellipse should scale down. After 45 seconds, the ellipse should scale up if the  $max(EQ)$  has been greater than 1 for 5 seconds.

$$s(t) = \begin{cases} -1 & 1 \leq max(EQ) < 1.05 \text{ for 5 seconds} \\ & \text{and } t < 45 \text{ sec} \\ 1 & 1 \leq max(EQ) \text{ for 5 seconds and } t \geq 45 \\ 0 & \text{otherwise} \end{cases} \quad (48)$$

The parameters for the simulation are shown in Table. 16. With each scaling step, the swarm scales down by 0.15 and up by 0.2. The minimum semi-major axis,  $a_{min}$ , was selected to 0.45 m because Eqns. (38) and (37) estimated a minimum possible semi-major axis of 0.4 m. The ACR was selected to limit the semi-major axis to 1.25 m.

The commanded and measured semi-major axis and  $max(EQ)$  with respect to time are shown in Fig. 17.  $max(EQ)$  spikes every time that the adaptive sizing state machine changes the ellipse parameters, but then always settles close to one for at least the expected 5 seconds. The semi-major axis tracks the commanded value very well when scaling up and down. For the first 45 seconds, the commanded semi-major axis never goes below  $a_{min}$ . When the semi-major axis is 0.55 m,  $max(EQ)$  settles close to but never goes below the 1.05 threshold. Therefore  $a_{min}$  should have been selected with more margin.

For the last 45 seconds, the commanded semi-major axis never goes above  $\frac{3}{8}d_l$ , which is 1.25 m. The percent error in semi-major axis, eccentricity, and angle  $\alpha$ , are calculated only for times when the swarm has settled to the ellipse,  $1 \leq max(EQ) < 1.05$ . This is because the swarm can take on any shape as it expands or contracts to the desired ellipse. Table 17 presents the mean and max percent errors for the adaptive sizing trials. The semi-major axis tracks very well with an average error of less than 1%. The eccentricity and angle have average percent errors of less than 5%, which is very good. The bounding ellipse that calculates the ellipse parameters is much more sensitive to individual robot positions for small numbers of robots with low eccentricities and small semi-major axes. Additionally, SBS uses a constant velocity controller, which could cause robots to overshoot the desired ellipse for a timestep. Therefore it is not surprising that these spikes are present.

## IX. CONCLUSION

The elliptical coverage problem for robotic swarms has several unique solutions. Successful swarm elliptical coverage should have complete coverage of an ellipse with commandable parameters, should be simple enough to both scale to a large number of robots and handle the addition and loss of robots naturally, and be able to adaptive sizing the commanded ellipse. However, no existing method has all three properties, and most methods are generally complicated. Additionally, no guidelines exist for selecting swarm and ellipse parameters to ensure successful elliptical aggregation. This work first presents a new elliptical attraction behavior

Parameter	Value
$[k_{b\_EQ}, k_{b\_n}]$	[0.9, 0.1]
Initial Semi-Major Axis (m)	1
Eccentricity	0.5
Angle $\alpha$ (deg)	45
Avoidance Range (m)	0.25
ACR (m)	3.33333
Number of Robots	12
Robot Speed (m/s)	0.1
Simulation Run Time (sec)	90
Initial Condition Area ( $m^2$ )	3.5 x 3.5
$\gamma_d$	0.15
$\gamma_u$	0.2
$a_{min}$ (m)	0.45
$\Delta$	0.05

TABLE 16. Parameter Summary for Adaptive Sizing Trial

Parameter	Mean Percent Error (%)	Max Percent Error (%)
a	0.69	4.54
e	3.27	18.06
$\alpha$	3.9477	16.2015

TABLE 17. Mean and max percent error when ellipse has converged

that meets all three tenants of swarm elliptical coverage. The Lyapunov stable elliptical attraction behavior for Reactive Particle Swarms has good adherence to the commanded ellipse parameters and coverage of the area. Second, an adaptive sizing algorithm for the elliptical attraction behavior is presented. With little error, the dynamic control can rapidly adjust the swarm's position to track the changing parameters. Third, guidelines for selecting swarm parameters of the number of robots and communication limits, as well as all ellipse parameters, are presented to ensure the swarm can aggregate to the commanded ellipse. All results are validated with simulated and on-hardware case studies. Future work will use this elliptical attraction behavior and the associated swarm and ellipse selection guidelines to solve several navigation problems.

## REFERENCES

- [1] L. Bayindir and E. Sahin, "A review of studies in swarm robotics," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 15, pp. 115–147, 2007.
- [2] M. A. Lewkowicz, R. Agarwal, and N. Chakraborty, "Distributed algorithm for selecting leaders for supervisory robotic swarm control." *IEEE*, Aug 2019, pp. 112–118.
- [3] S. Avrahami and N. Agmon, "The robotic swarm contamination problem." *IEEE*, Aug 2019, pp. 119–125.
- [4] L. Briñón-Arranz, A. Seuret, and C. Canudas-de Wit, "Collaborative estimation of gradient direction by a formation of augs under communication constraints," *Proceedings of the IEEE Conference on Decision and Control*, pp. 5583–5588, 12 2011.
- [5] M. S. Couceiro, R. P. Rocha, and N. M. F. Ferreira, "A novel multi-robot exploration approach based on particle swarm optimization algorithms." *IEEE*, Nov 2011, pp. 327–332.
- [6] S. T. Hart and C. A. Kitts, "Unifying control architecture for reactive particle swarms," *IEEE/ASME Transactions on Mechatronics*, pp. 1–11, 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9914633>
- [7] J. D. McLurkin, "Stupid robot tricks: a behavior-based distributed algorithm library for programming swarms of robots," Ph.D. dissertation, 2004.



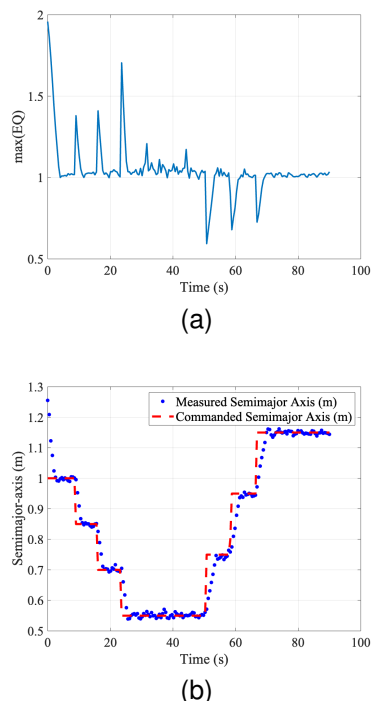


FIGURE 17. Adaptive Sizing Case Study a)  $\max(EQ)$  and b) semi-major axis with respect to time.

[8] D. Roy, A. Chowdhury, M. Maitra, and S. Bhattacharya, "Geometric region-based swarm robotics path planning in an unknown occluded environment," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 7, pp. 6053–6063, 2021.

[9] X. Lan, Z. Wu, W. Xu, and G. Liu, "Adaptive-neural-network-based shape control for a swarm of robots," *Complexity*, vol. 2018, pp. 1–8, 12 2018.

[10] S. Hou and C. Cheah, "Dynamic compound shape control of robot swarm," *IET Control Theory & Applications*, vol. 6, no. 3, p. 454, 2012.

[11] R. Haghighi and C. C. Cheah, "Asynchronous dynamic multi-group formation for swarm robots." *IEEE*, Dec 2011, pp. 2744–2749.

[12] S. Hou, C. Cheah, and J. Slotine, "Dynamic region following formation control for a swarm of robots." *IEEE*, May 2009, pp. 1929–1934.

[13] F. Chen and W. Ren, "A connection between dynamic region-following formation control and distributed average tracking," *IEEE Transactions on Cybernetics*, vol. 48, no. 6, pp. 1760–1772, Jun 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/7959651>

[14] L. Barnes, M. Fields, and K. Valavanis, "Swarm formation control utilizing elliptical surfaces and limiting functions," *IEEE Transactions on Systems Man and Cybernetics Part B (Cybernetics)*, vol. 39, no. 6, pp. 1434–1445, Dec 2009. [Online]. Available: <https://ieeexplore.ieee.org/document/4915742>

[15] C. C. Cheah, S. P. Hou, and J. J. E. Slotine, "Region-based shape control for a swarm of robots," *Automatica*, vol. 45, no. 10, pp. 2406–2411, Oct 2009. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0005109809003215>

[16] P. S. Trakas, C. P. Bechlioulis, and G. A. Rovithakis, "Coordinated navigation of holonomic robot swarms in complex workspaces via dynamic formation tracking control." *IEEE*, Jun 2022, pp. 79–84. [Online]. Available: <https://ieeexplore.ieee.org/document/9837243/>

[17] C. J. Stamouli, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Multi-agent formation control based on distributed estimation with prescribed performance," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2929–2934, Apr 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/8977390/>

[18] G. Fedele, L. D'Alfonso, and A. Bono, "Swarm Fixed-Time Reference Tracking: a Discrete Model," *International Journal of Control*, pp. 1–27, 2021.

[19] E. Teruel, R. Aragues, and G. Lopez-Nicolas, "A practical method to cover evenly a dynamic region with a swarm," *IEEE Robotics and*

*Automation Letters*, vol. 6, no. 2, pp. 1359–1366, Apr 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9349134/>

[20] E. Teruel, R. Aragues, and G. López-Nicolás, "A distributed robot swarm control for dynamic region coverage," *Robotics and Autonomous Systems*, vol. 119, pp. 51–63, Sep 2019. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0921889018309436>

[21] E. Rosero and H. Werner, "Cooperative source seeking via gradient estimation and formation control (part 1)." *IEEE*, Jul 2014, pp. 628–633.

[22] S. Hart, N. Metzger, M. Reese, R. McDonald, M. Neumann, and C. Kitts, "Robotics Simulator for Development and Verification of Swarm Behaviors," ser. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. Volume 5A: 43rd Mechanisms and Robotics Conference, 08 2019, v05AT07A050. [Online]. Available: <https://doi.org/10.1115/DETC2019-97622>

[23] S. Tomer, C. Kitts, M. Neumann, R. McDonald, S. Bertram, R. Cooper, M. Demeter, E. Guthrie, E. Head, A. Kashikar, J. Kitts, M. London, A. Mahacek, R. Rasay, D. Rajabhandharaks, and T.-y. Yeh, "A low-cost indoor testbed for multirobot adaptive navigation research," in *2018 IEEE Aerospace Conference*, 2018, pp. 1–12.

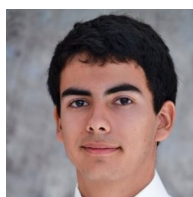
[24] L. G. Khachiyan, "Rounding of polytopes in the real number model of computation." *Mathematics of Operations Research*, vol. 21, no. 2, pp. 307–320, May 1996. [Online]. Available: <http://pubsonline.informs.org/doi/10.1287/moor.21.2.307>

[25] N. Moshtagh, "Minimum volume enclosing ellipsoid," <https://www.mathworks.com/matlabcentral/fileexchange/9542-minimum-volume-enclosing-ellipsoid>, 2021, [Online; accessed Aug 31, 2021].

[26] J. McClave and T. Sincich, *Statistics*, 13th ed. Pearson Education, 2017, ch. 8-10.



SHAE T. HART received a B.S. degree in physics from Carnegie Mellon University in 2016. She received an M.S. degree in mechanical engineering from Santa Clara University in 2019, where she is currently pursuing a Ph.D. in mechanical engineering. She was a Teaching Assistant with the Department of Mechanical Engineering and a Research Assistant with the Santa Clara's Robotic Systems Laboratory.



JAKE KAMENETSKY Jake Kamenetsky received a B.S. degree in mechanical engineering from University of California, Santa Barbara in 2017. He received a M.S degree in mechanical engineering from Santa Clara University in 2022. Since 2017, he has been a Guidance, Navigation, and Control Engineer at Lockheed Martin.



CHRISTOPHER A. KITTS received the B.S.E. degree in mechanical and aerospace engineering from Princeton University, the M.P.A degree in international and defense policy from the University of Colorado, and the M.S. degree in aeronautics and astronautics and the Ph.D. degree in mechanical engineering from Stanford University. His industrial background includes service as a Satellite Constellation Mission Controller with the U.S. Air Force and work as a Computer Scientist while he was a Contractor with the NASA Ames Research Center's Computational Sciences Division. He is currently a Professor of mechanical engineering, the Director of the Robotic Systems Laboratory, and the Associate Dean of the Research and Interdisciplinary Programs for the School of Engineering, Santa Clara University. He is a Fellow of the American Society of Mechanical Engineers.

• • •