

Guaranteed Encapsulation of Targets with Unknown Motion by a Minimalist Robotic Swarm

Himani Sinhmar, Hadas Kress-Gazit

Abstract—We present a decentralized control algorithm for a robotic swarm given the task of encapsulating static and moving targets in a bounded unknown environment. We consider minimalist robots without memory, explicit communication, or localization information. The state-of-the-art approaches generally assume that the robots in the swarm are able to detect the relative position of neighboring robots and targets in order to provide convergence guarantees. In this work, we propose a novel control law for the guaranteed encapsulation of static and moving targets while avoiding all collisions, when the robots do not know the exact relative location of any robot or target in the environment. We make use of the Lyapunov stability theory to prove the convergence of our control algorithm and provide bounds on the ratio between the target and robot speeds. Furthermore, our proposed approach is able to provide stochastic guarantees under the bounds that we determine on task parameters for scenarios where a target moves faster than a robot. Finally, we present an analysis of how the emergent behavior changes with different parameters of the task and noisy sensor readings.

Index Terms—Collision avoidance, Decentralized control, Minimalist robot swarm, Lyapunov stability, Target tracking

I. INTRODUCTION

TYPICAL approaches to swarm robotics propose simple local behaviors for large numbers of simple robots such that they collectively accomplish a complex task; many approaches study the properties of the emergent behavior [1]–[3]. In this work, we consider a swarm consisting of homogeneous robots which are minimalist; they have no memory, cannot broadcast or receive location information from their neighbors and are unable to plan ahead. Minimalistic robotic swarms [4], [5] have a number of applications, ranging from nanomedicine to underwater monitoring and surveillance [6], [7], where robots might not be able to efficiently communicate with a central controller or with each other, and might not have the ability to self localize. For example, in an underwater mission, communication may be limited to acoustic signals, which are sensitive to interference and lead to errors in the relative positioning of nearby entities. Similarly, in nanomedicine applications a swarm of micro-robots may be required to search, encapsulate and destroy tumors by following a chemical gradient [8].

In this paper, we focus on the problem of encapsulating multiple targets, which are moving in unknown motion patterns, by a minimalist robotic swarm. A robot in the swarm has no knowledge of the exact relative location of nearby robots, targets, or the boundary of the environment.

This work extends our previous work on encapsulating static targets [5] by addressing moving targets. We develop an orbiting behavior for robots to encapsulate the targets, in addition to the searching for targets and avoiding collisions within the swarm, as in [5]. We compare the efficiency of our previous algorithm with the one introduced in this paper. Furthermore, we also show the behavior of our algorithm when applied to non-circular robots.

Related Work: There has been extensive work on developing various techniques to localize and track a moving target while ensuring collision avoidance [9]–[14]. In [15], authors introduced a motion planning strategy for a single robot based on velocity pursuit to intercept a target moving with unknown maneuvers. For target tracking using a multi-robot system, most approaches use artificial potential fields to design a controller consisting of a virtual attraction force to move towards a target and a repulsion force to avoid collision with obstacles [16]. Another widely used approach to guarantee collision avoidance with dynamic obstacles is using a limit cycle method [17]. The authors of [18] introduced a hybrid approach where they instead used the limit cycle method to encircle a moving target using a swarm of holonomic robots, and artificial potential fields for collision avoidance. Since the use of the limit cycle method, either for surrounding a target or avoiding collision with obstacles requires the exact knowledge of the neighbor’s relative position information, we cannot use it for our minimalist robotic swarm.

Pursuit-evasion games [19]–[21] provide guarantees for catching a faster-moving evader by constructing an encircling formation of pursuers composed of a series of Apollonius circles around a target and slowly closing the escape paths of the evader. In this approach, an evader is captured if a pursuer meets the evader at the same point at the same time. Most of the pursuit-evasion methods in the literature assume knowledge of the target’s motion model. In this work, we do not assume such knowledge.

Existing research [22] in “hunting” of dynamic targets generally makes use of communication within the team and formation-keeping control strategies, while approaching the target, to ensure that all of the escaping routes of the targets are occupied by the robots. Work in [23] developed a leader-follower strategy based on the behavior of wolves to hunt a randomly moving target with unexpected behaviors. The authors of [24] proposed a limit cycle based algorithm using a neural oscillator to surround a target moving with unknown but constant velocity. The authors of [25] utilized rule-based mechanisms using only relative positions of neighbors and no direct communication within the swarm for surrounding an

escaping target by introducing a circulating behavior in the swarm.

Recent research in colloidal swarms has shown the capture of multiple randomly moving targets using self-organization control schemes. In [26], the authors designed a stochastic centralized controller for an intelligent colloidal micro-robotic swarm to capture multiple Brownian targets in a maze. In [27], the authors show via simulations, the feedback-controlled reconfigurability of colloidal particles that act as a swarm capable of capturing and transporting microscopic Brownian cargo.

To implement a distributed approach of searching and encircling targets in an inexpensive and efficient way, in [28] the authors developed a new dual-rotating proximity sensor to obtain relative position information of neighbors for tracking multiple targets with a minimalist swarm. Authors of [29] proposed a scheme to estimate the global quantities required by the controller in a decentralized way using only local information exchange between robots for the guaranteed encirclement of a 2D or 3D target.

While the above approaches successfully solve the target encirclement while avoiding collisions, most of them rely on the assumption that robots have knowledge of the exact relative location of both their neighbors and the target. Furthermore, it is a common assumption that the average speed of the agents in the swarm is greater than that of the moving target to guarantee encapsulation [30]. In contrast, in this work, we provide guarantees on the encapsulation of dynamic targets without the requirement of accurate (relative) location information and without direct communication within the swarm.

Contributions: This paper's contributions are: (i) a discrete-time decentralized control law for a minimalist robotic swarm that guarantees the encapsulation of dynamic targets, for different target motion models, without accurate detection of the relative location of either the targets or neighboring robots, given certain bounds are met (ii) sensor-placement dependent bounds on the ratio between the target and robot speeds to guarantee encapsulation, (iii) proof of stochastic convergence of our control law for scenarios when a target is moving faster than a robot, and (iv) simulations and analysis of emergent behavior of the swarm in the presence of sensor noise and different task parameters.

II. DEFINITIONS

In this section, we provide definitions from [5] that we use throughout.

Environment: We consider a 2D bounded environment $E \subseteq \mathbb{R}^2$. The environment has a fixed global frame $\mathcal{I}$.

Robot: We model a robot, $R = (\mathbf{c}_r, \gamma_r, r_r, p, Z)$, as a disk of radius r_r centered at $\mathbf{c}_r \in E$ with heading $\gamma_r \in \mathbb{S}$. The shape of a robot does not affect the analysis presented in the paper since the robot can always be circumscribed by a circle of radius r_r . The kinematics of a robot is given by Eq. (1),

which is a typical model for a differential drive robot. At each time step T , the robot is controlled in a *turn-then-move* scheme with control inputs $\theta_r \in \mathbb{S}$ and $d_r \in \mathbb{R}^+$. The maximum step-size of a robot is $d_r^{\max}$.

$$\begin{aligned}\gamma_{r,T} &= \gamma_{r,T-1} + \theta_r \\ \mathbf{c}_{r,T} &= \mathbf{c}_{r,T-1} + d_r [\cos \gamma_{r,T} \quad \sin \gamma_{r,T}] \end{aligned} \quad (1)$$

Each robot is reactive, memoryless, has no knowledge of the relative locations of other robots or targets, and cannot communicate with its neighbors. A robot has p isotropic sensors arranged on its boundary such that $\phi^k \forall k \in \{1 \dots p\}$ is the angle between the k^{th} sensor and the robot's heading direction. Z is the set of measurements from all sensors on a robot.

Signal Sources and Sensors: We consider three types of signal-emitting sources present in the environment that a robot can detect: s_g from a point source at the center of a target, s_r from a point source at the center of a robot, and s_e from a line source present on the entire environment boundary. For clarity in notation, we hereby denote the signal set $\{s_g, s_r, s_e\}$ by $\{g, r, e\}$.

The strength of any signal $s \in \{g, r, e\}$ located at a distance d from a signal source is given by the function $B_s(d)$. The influence distance of a source is limited to β_s , such that $B_s(d) = 0 \forall d \geq \beta_s$. Let N_s^k be the set of all the sources of type s in the sensing range of the k^{th} sensor and d_j^k be the distance of this sensor from a source $j \in N_s^k$. Then the sensor reading $z_s^k = \sum_{j \in N_s^k} B_s(d_j^k)$ is the sum of signal strengths from all sources in N_s^k . This summation becomes an integral over the boundary segment for a line source present inside the influence region β_e .

The tuple (z_g^k, z_r^k, z_e^k) corresponds to the measurements of the k^{th} sensor. Let $Z_g = \{z_g^1 \dots z_g^p\}$, $Z_r = \{z_r^1 \dots z_r^p\}$ and $Z_e = \{z_e^1 \dots z_e^p\}$, then the measurement set is $Z = Z_g \cup Z_r \cup Z_e$. We define $r_s^{\text{safe}} \forall s \in \{g, r, e\}$ as the user-specified minimum safety distance that a robot must maintain from a source at all times.

The isotropic source-sensor model in this paper is inspired by sensing constraints ranging from LED-photodiode implementations in multi-robotic systems [31] to nanomedicine applications where LC-based microscopic sensors can be used for guiding chemotaxis [32] to source localization in radioactive nuclear plants [33].

III. PROBLEM FORMULATION

We model a target $g = (\mathbf{c}_g, r_g)$ as a disk of radius r_g centered at $\mathbf{c}_g \in E$. $\mathcal{G}$ is the set of all targets contained in E . The kinematics of a target is given in Eq. (2). At any time step T , $d_g \in \mathbb{R}^+$ and $\theta_g \in \mathbb{S}$ are the distance moved and angle turned by the target, and $\gamma_{g,T} \in \mathbb{S}$ is the target heading.

$$\begin{aligned}\gamma_{g,T} &= \gamma_{g,T-1} + \theta_g \\ \mathbf{c}_{g,T} &= \mathbf{c}_{g,T-1} + d_g [\cos \gamma_{g,T} \quad \sin \gamma_{g,T}] \end{aligned} \quad (2)$$

The maximum distance that a target can move in a time step is limited to $d_g^{\max}$.

Target Motion Models: In this paper, we design controllers and analyze the swarm behavior for different types of target motion models. A target can exhibit one of the following motions:

- 1) Target moves randomly such that at any time step T , $\gamma_{g,T} \in [0, 2\pi)$, $d_g \in [0, d_g^{\max}]$ and $\mathbf{c}_{g,T} \in E$.
- 2) Target moves randomly as in motion model 1 until a robot is in its *escape domain* $= (\mathbf{c}_g, r_g^{\text{escape}})$ of radius r_g^{escape} centered at $\mathbf{c}_g$, in which case the target chooses a heading direction to escape from all the robots that satisfies $\|\mathbf{c}_{g,T} - \mathbf{c}_{r,T}\| \leq r_g^{\text{escape}}$, and moves the maximum step-size d_g .
- 3) Target follows an unknown motion pattern until a robot satisfies $\|\mathbf{c}_{g,T} - \mathbf{c}_{r,T}\| \leq r_g^{\text{escape}}$, in which case it chooses a heading direction to escape nearby robots.

Target Encapsulation: For each target $g \in \mathcal{G}$, we define an encapsulation ring $\mathcal{A}_{g,T} = (\mathbf{c}_{g,T}, r_g^{\text{safe}}, r_g^{\text{encap}})$ of inner radius r_g^{safe} and outer radius r_g^{encap} centered at $\mathbf{c}_{g,T}$. A robot R is considered to be in $\mathcal{A}_{g,T}$ if, $r_g^{\text{safe}} < \|\mathbf{c}_{r,T} - \mathbf{c}_{g,T}\| \leq r_g^{\text{encap}}$. A target is *encapsulated* if the total number of robots present in the encapsulation ring is n_g , which is a user-specified input as shown in Fig. 1.

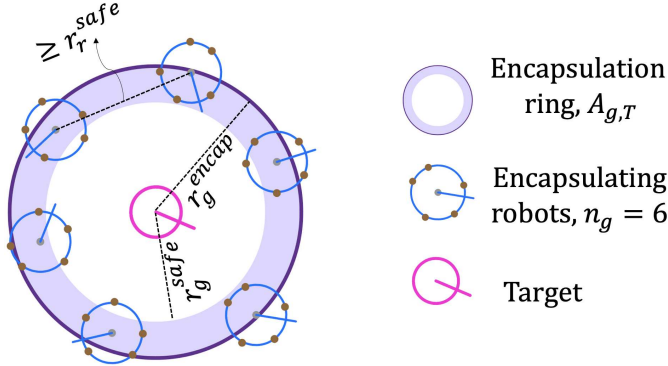


Fig. 1: A target is encapsulated if n_g robots are present simultaneously in the encapsulation ring while maintaining at least a distance of r_r^{safe} from each other.

Problem statement: Consider a bounded environment $E \subseteq \mathbb{R}^2$ with m dynamic targets where the initial distribution of the robots and targets is arbitrary. Given the total number of sensors p on a robot, the user-provided safe distance $r_s^{\text{safe}} \forall s \in \{g, r, e\}$, the encapsulation ring $\mathcal{A}_g$, and the number of robots n_g needed to encapsulate each target g such that the total number of robots $n \geq \sum_{g \in \mathcal{G}} n_g$, our objective is to find a real-time decentralized control law for encapsulating all targets while ensuring safety distances are always maintained. We make the following assumptions about the environment and the system:

Assumption 1. The sensors are arranged on a robot such that when a robot's center is r_s^{safe} away from a source s , at least one sensor is in the influence region of the source. For ease of exposition, we consider circular robots with a symmetric placement of sensors to explain our algorithm, and

show in simulations how asymmetric sensor placements and non-circular robots affect swarm behavior.

Assumption 2. The distance between any two moving targets is greater than $(2\beta_g + 2r_r)$. That is, a robot can sense at most one target at a time.

Assumption 3. We constrain a target to maintain a minimum distance of $(r_g^{\text{encap}} + r_e^{\text{safe}} + d_r^{\max})$ from the environment boundary. This ensures that robots will be able to encapsulate the target without colliding with the environment boundary.

Assumption 4. We place no restriction on the target's knowledge of the environment; it may be able to perfectly sense the relative location of any robot present in its user-specified *escape domain*, r_g^{escape} . However, if a target is encapsulated, we assume it emits a single burst of a shut-off signal and stops emitting any signal subsequently. The influence distance of this signal is limited to $\mathcal{A}_g$, and we assume that thereafter both the robots within the encapsulation ring and the target stop moving, i.e. $d_r = 0$ and $d_g = 0$, respectively. This assumption emulates applications in nano-medicine, for example, encapsulating and eventually destroying a tumor and applications in localizing and shutting off a chemical source in plants.

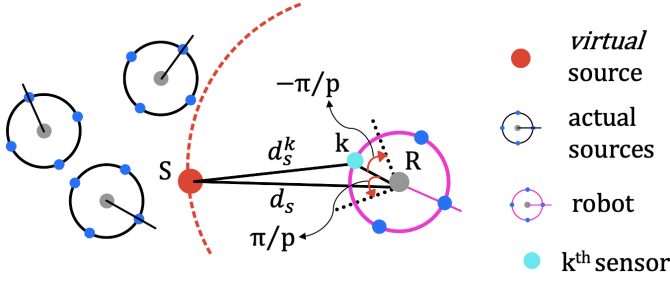
Assumption 5. The signal strength B_s strictly decreases with the radial distance, d from a source and the inverse of the signal function $B_s(d)$ exists and is known to the robots.

IV. APPROACH

Our strategy for designing a local control law is based on geometry and the relative kinematics of the interaction of a robot with its neighboring robots and a dynamic target. We extend our previous work [5] where we only considered static targets; a robot's behavior there was to either move randomly in the bounded environment when it does not sense any target, or to move towards a target if sensing one while ensuring safety. Here, we introduce an additional robot behavior of orbital encirclement of a target, inspired by [17]. As we show in Section V, this behavior ensures the encapsulation of an escaping target. In Section IV-A we describe virtual sources as defined in [5] and use them to under-approximate the relative distance between a source and the robot's center as a function of the sensor placement. In Section IV-B we find the bounds on control parameters (d_r and θ_r) for a robot to ensure that it maintains r_s^{safe} distance from a source $s \in \{g, r, e\}$. In Section IV-C we introduce the concept of orbital encirclement of a moving target; we provide a summary of the overall reactive control law for a robot in the swarm in Section IV-D.

A. Virtual Source

Since we assume a robot is equipped with isotropic sensors, a sensor measurement corresponds to the aggregated signal strength from all the nearby sources. Hence, the same measurement could correspond to a single source nearby or a cluster of sources further away. Therefore, for each sensor reading, $z_s^k \forall s \in \{g, r, e\}$, we define a virtual source on a circle centered at the sensor k as shown in Fig. 2. It is shown in [5] that the closest possible location of the virtual

Fig. 2: Virtual source for the k^{th} sensor [5].

source with respect to the robot's center is given by Eq. (3). Furthermore, the range of possible directions of the location of the virtual source with respect to the robot's center is restricted to $[\phi^k - \pi/p, \phi^k + \pi/p]$ for symmetric sensor placement.

$$d_s = r_r \cos(\pi/p) + \sqrt{(d_s^k)^2 - r_r^2 \sin^2(\pi/p)} \quad (3)$$

For asymmetric sensor placement, we replace π/p with half of the maximum angle that the k^{th} sensor makes with either of its adjacent sensors. Similarly, for robots that are not circular in shape, we replace r_r by the distance between the k^{th} sensor and the robot's center in the above equation. We can see in (3) that as $p \rightarrow \infty$, $d_s \rightarrow d_s^k + r_r$. That is, the error in locating the source is dependent on the total sensors on a robot.

B. Collision Avoidance

We use the technique introduced in [5] for collision avoidance with nearby robots and the environment boundary. At each time step, the robot estimates the relative distance between its center and the nearby sources using Eq. (3) for the sensor with the maximum sensor reading $z_s^k \forall s \in \{g, r, e\}$. If this distance is less than or equal to $(r_s^{\text{safe}} + d_s^{\text{max}})$, the collision avoidance behavior is triggered for this robot to ensure safety.

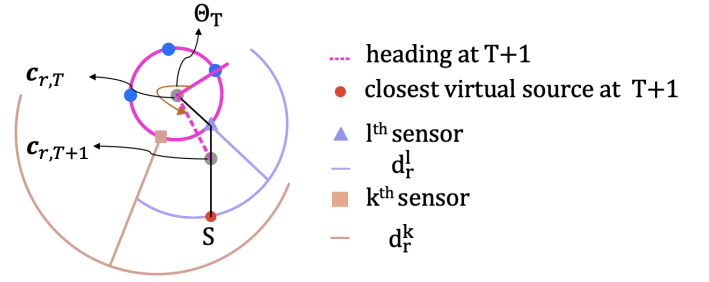
We have shown in [5] that to avoid collisions with static obstacles (such as environment boundary), the robot's heading direction θ_r must be chosen from the angular range given by Eq. (4).

$$\Theta_e^{\text{avo}} = [\phi^k + \pi/p + \pi/2, \phi^k - \pi/p + 3\pi/2] \quad (4)$$

Whereas to avoid the neighboring moving robots, the distance d_r that a robot moves at time step T in a given heading direction $\gamma_{r,T}$ must be chosen such that at $T+1$ it maintains at least a distance of r_r^{safe} from the closest neighboring robot. As shown in Fig. 3, let k and l be the indices of the sensors closest to the intended heading direction γ_r at time T and d_r^k and d_r^l are their radii of virtual sources respectively such that $d_r^k > d_r^l$. Then, we can compute the bounds on the step-size d_r that the robot can take in the heading direction $\gamma_{r,T}$ using Eq. (5).

$$0 \leq d_r \leq r_r \cos(\phi^l - \theta_r) + \sqrt{(d_r^l - r_r^{\text{safe}} - d_r^{\text{max}})^2 - r_r^2 \sin^2(\phi^l - \theta_r)} \quad (5)$$

To ensure that two robots never deadlock, the bounds on the maximum step size a robot can take, and the influence

Fig. 3: Computing d_r such that collision is avoided with nearby moving robots [5].

region of a robot's source, are given by Eq. (6) and Eq. (7), respectively. The proof is detailed in Lemma V.3 of [5].

$$d_r^{\text{max}} < \frac{r_r^{\text{safe}} + r_r \cos(\pi/p)}{2} - \frac{\sqrt{(r_r^{\text{safe}})^2 + r_r^2 - 2r_r r_r^{\text{safe}} \cos(\pi/p)}}{2} \quad (6)$$

$$\sqrt{(r_r^{\text{safe}})^2 + r_r^2 - 2r_r r_r^{\text{safe}} \cos(\pi/p)} + 2d_r^{\text{max}} < \beta_r < r_r^{\text{safe}} + r_r \cos(\pi/p) \quad (7)$$

C. Encirclement of a Target

In [5], our approach to encapsulate a static target, was for a robot to either move towards the target or move away from an obstacle between itself and the target in the direction of the sensor receiving the minimum reading from nearby moving robots. However, in order to surround a dynamic target, the behavior of a robot should be such that the swarm is able to disperse around the target in order to block off its escaping paths. Since we consider minimalist robots that can neither communicate with their neighbors nor know their exact relative position, we can not make use of formation control strategies, such as [18], [24].

Consider a scenario where all the robots in the swarm start on one side of a target. Then, for a swarm to disperse around a target, it is necessary that an individual robot be able to catch up with the escaping target, and once the robot reaches the encapsulation ring, it should be able to encircle the target so that the target is prevented from escaping.

To ensure encapsulation, we define primary and secondary orbits around each target, as shown in Fig. 4. For each orbit, we define a *tie-breaking orbital rotation* which can be either clockwise (denoted by a value of -1) or counter-clockwise (denoted by a value of 1). The primary orbit, $Or_0 = (\mathbf{c}_{g,T}, Or_0^{\text{inner}}, Or_0^{\text{outer}}, -1)$ is an annular ring centered at $\mathbf{c}_{g,T}$ with an inner radius of $Or_0^{\text{inner}} \geq r_g^{\text{safe}} + d_g^{\text{max}}$, an outer radius $Or_0^{\text{outer}} = r_g^{\text{encap}}$ and a clockwise orbital rotation (chosen arbitrarily). Let w be the width of a secondary orbit, then an i^{th} secondary orbit is given by, $Or_i = (\mathbf{c}_{g,T}, Or_0^{\text{outer}} + (i-1)w, Or_0^{\text{outer}} + iw, (-1)^{i-1})$, $\forall i > 0$. We consider a robot to be in i^{th} orbit if, $Or_i^{\text{inner}} < \|\mathbf{c}_{g,T} - \mathbf{c}_{r,T}\| \leq Or_i^{\text{outer}}$. Each robot in the swarm computes its *current orbit* using its estimate of $\|\mathbf{c}_{g,T} - \mathbf{c}_{r,T}\|$. At time-step T , let Or_i be the

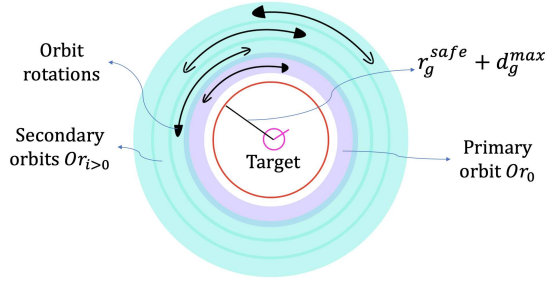


Fig. 4: Primary (purple ring) and secondary (cyan rings) orbits around a target, and the lower bound on the target's escape domain (red circle). A robot moves either clockwise or counter-clockwise in an orbit depending on its neighbors. The solid arrows denote the tie-breaking rotation for an orbit.

current orbit as estimated by a robot, then its control consists of one of the following behaviors:

- 1) if $i > 0$, the robot moves towards the target in a heading direction chosen from the line of sight angular range as estimated from the virtual source (Θ_g^{LOS}) while maintaining a safe distance from nearby robots.
- 2) else if $i > 0$ and the robot cannot move a non-zero distance towards the target, it moves tangentially in its current orbit while maintaining a safe distance from nearby robots. The direction of the tangent is chosen such that it maximizes the possible step-size d_r . In case of symmetry, the robot moves in the *orbital rotation* of the i^{th} orbit.
- 3) else if $i > 0$ and the robot can neither move in a direction from Θ_g^{LOS} nor tangential to the orbit, it chooses a direction of motion that maximizes the possible step-size d_r .
- 4) else if $i = 0$, the robot moves tangentially in its current orbit while maintaining a safe distance from the target.
- 5) else if the relative distance between the target and a robot is less than or equal to Or_0^{inner} , it moves away from the target.
- 6) else the robot performs a simple random walk while avoiding nearby moving robots.

In general, a robot moves toward the target until it reaches the primary orbit. If other robots are present between itself and the target, the robot moves tangentially in its current orbit until it can move toward the target. All the robots that place themselves in the primary orbit constantly move tangentially and eventually close off the target's escape routes. The width of a secondary orbit, w , must be less than β_r , so that a robot's neighbors in adjacent orbits lie within its sensing range. This ensures that a robot doesn't move towards a target when it senses other robots in the front and instead moves tangentially in its current orbit. Fig. 5 depicts how robots move in different orbits around a target, over time, while avoiding nearby robots.

Now, using the sensor readings and their corresponding virtual sources, we find the set of directions that a robot needs to choose from to move towards a target, away from a target, or tangentially in an orbit. Let k be the index of the sensor such that $z_g^k > z_g^l, \forall l \neq k$. Here we have ignored

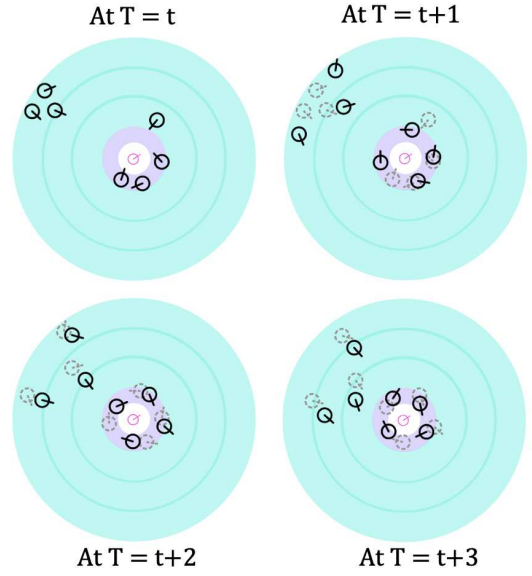


Fig. 5: Motion of robots in different orbits over four consecutive time-steps. Dotted grey robots denote the location of the robots in the previous time step.

the unlikely scenario where two sensors receive the same maximum intensity from a target. Then the angular range, Θ_g^{LOS} , for the possible location of the target with respect to the robot's center is given by Eq. (8).

$$\Theta_g^{\text{LOS}} = [\phi^k - \pi/p, \phi^k + \pi/p] \quad (8)$$

The angular range, Θ_g^{avo} (Eq. (9)), to move away from the target can be derived in a similar fashion to Eq. (4).

$$\Theta_g^{\text{avo}} = [\phi^k + \pi/p + \pi/2, \phi^k - \pi/p + 3\pi/2] \quad (9)$$

The angular range to move tangentially in an orbit in a clockwise or counter-clockwise direction is given by Eq. (10) and Eq. (11), respectively, where we define $\Theta_g^{\text{tan}} = \Theta_g^{\text{tan},+1} \cup \Theta_g^{\text{tan},-1}$.

$$\Theta_g^{\text{tan},-1} = [\phi^k - \pi/p + \pi/2, \phi^k + \pi/p + \pi/2] \quad (10)$$

$$\Theta_g^{\text{tan},+1} = [\phi^k - \pi/p + 3\pi/2, \phi^k + \pi/p + 3\pi/2] \quad (11)$$

Fig. 6 shows the different angular range sets for a target-robot interaction. It is worth mentioning that for noiseless sensors,

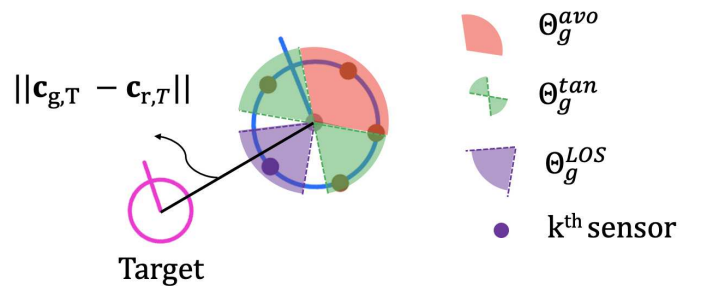


Fig. 6: The angular range set for a target-robot interaction. The robot is equipped with 5 sensors placed asymmetrically.

if $z_g^{k-1} > z_g^{k+1}$ then $\Theta_g^{\text{LOS}} = [\phi^k - \pi/p, \phi^k]$. This results

in a more accurate estimation of the location of a target and reduces the angular resolution error by half. The estimation of $\Theta_g^{\tan}$ and Θ_g^{avo} also changes accordingly.

As shown in our previous work [5], a heading direction in the angular ranges Θ_g^{LOS} and Θ_g^{avo} is guaranteed to make a robot move towards the target and away from the target, respectively. In contrast, a robot might end up moving towards or away from the target when it moves tangentially in an orbit. Since secondary orbits are at least at a distance of Or_0^{outer} from a target, a robot moving tangentially in these orbits will always maintain a safe distance from the target. However, if a robot is moving tangentially in the primary orbit, we need to make sure that it maintains at least a distance of Or_0^{inner} from the target after moving d_r units in the intended heading direction $\gamma_{r,T}$ such that $\theta_r \in \Theta_g^{\tan}$.

In Fig. 7, we can see that at $T+1$, the closest possible location of the target is at $S \in \mathcal{I}$. That is, if the heading direction $\theta_r \notin \Theta_g^{\text{LOS}}$, the closest possible location of the target with respect to the robot's center at $T+1$ would be along one of the extremes of the angular range Θ_g^{LOS} . To ensure safety,

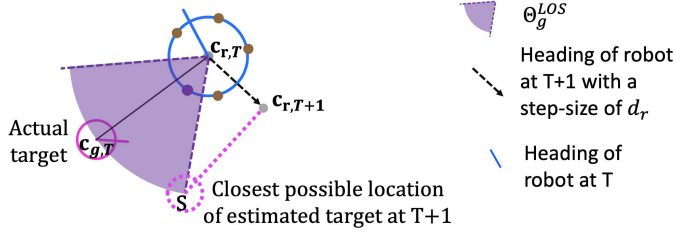


Fig. 7: The distance d_r that the robot can move in the intended heading is computed using the geometry of $\Delta \mathbf{Sc}_{r,T}\mathbf{c}_{r,T+1}$

$\|\mathbf{c}_{r,T+1} - \mathbf{S}\| \geq Or_0^{\text{inner}}$. Using the cosine rule of triangle for $\Delta \mathbf{Sc}_{r,T}\mathbf{c}_{r,T+1}$, the bounds on the control parameter d_r can be computed using the quadratic inequality given by Eq. (12).

$$d_r^2 - 2d_r \|\mathbf{c}_{r,T} - \mathbf{S}\| \cos(\angle \mathbf{Sc}_{r,T}\mathbf{c}_{r,T+1}) + \|\mathbf{c}_{r,T} - \mathbf{S}\|^2 \geq (Or_0^{\text{inner}})^2 \quad (12)$$

In the above inequality, $\|\mathbf{c}_{r,T} - \mathbf{S}\|$ is under-approximated by the robot using the virtual source, as described in Section IV-A.

D. Local Control Law

Algorithm 1 encodes the local reactive control law for a robot in the swarm that is tasked with searching and encapsulating targets while avoiding collisions.

The algorithm describes the computation that happened at each time step T . Each robot in the swarm has: Z —the tuple of sensor measurements, B_s —the function describing the signal source strength as a function of radial distance from $s \in \{g, r, e\}$, the maximum step-size of a robot d_r^{max} and a target d_g^{max} , the user-specified safety constraints for each source r_s^{safe} , and the set `orbits` defined by an inner and outer radius of each orbit.

The control synthesis proceeds as follows: First, the robot estimates its distance, `DistToEnvBound` (Section IV-A), from the environment boundary. If the robot is too close to the boundary, it computes the allowed set of heading directions,

Algorithm 1: Control algorithm for a robot

Input : $Z, B_s, p, d_r^{\text{max}}, d_g^{\text{max}}, r_s^{\text{safe}}, \forall s \in \{r, g, e\}$,
orbits $Or_{i \geq 0}$

Output: d_r, θ_r
// compute $\Theta_e^{\text{avo}}, \Theta_g^{\text{LOS}}, \Theta_g^{\text{avo}}, \Theta_g^{\tan}$,
currentOrbit

```

1 if DistToEnvBound ≤ r_e^safe + d_r^max then
2   θ_r = arg max_{θ ∈ Θ_e^avo} DistAvoRob(Z_r, B_r, θ, r_r^safe)
3   d_r = DistAvoRob(Z_r, B_r, θ_r, r_r^safe)
4 else if max(Z_g) = 0 then
5   θ_r = randsample([0, 2π))
6   d_r = DistAvoRob(Z_r, B_r, θ_r, r_r^safe)
7   if d_r = 0 then
8     k = argmin(Z_r)
9     θ_r = ϕ^k
10    d_r = DistAvoRob(Z_r, B_r, θ_r, r_r^safe)
11 else if DistToTar < Or_0^inner then
12   θ_r = arg max_{θ ∈ Θ_g^avo} DistAvoRob(Z_r, B_r, θ, r_r^safe)
13   d_r = DistAvoRob(Z_r, B_r, θ_r, r_r^safe)
14   d_r^req = ||DistToTar - Or_0^inner||
15   if d_r > d_r^req then
16     d_r = d_r^req
17 else if currentOrbit = Or_0 then
18   θ_r = arg max_{θ ∈ Θ_g^tan} min(DistAvoRob(Z_r, B_r, θ, r_r^safe),
                               DistAvoTar(Z_g, B_g, Θ_g^LOS, θ))
19   d_r = DistAvoRob(Z_r, B_r, θ_r, r_r^safe)
20 else
21   θ_r = arg max_{θ ∈ Θ_g^LOS} DistAvoRob(Z_r, B_r, θ, r_r^safe)
22   d_r = DistAvoRob(Z_r, B_r, θ_r, r_r^safe)
23   if d_r = 0 then
24     θ_r = arg max_{θ ∈ Θ_g^tan} DistAvoRob(Z_r, B_r, θ, r_r^safe)
25     d_r = DistAvoRob(Z_r, B_r, θ_r, r_r^safe)
26     if d_r = 0 then
27       k = argmin(Z_r)
28       θ_r = ϕ^k
29     d_r = DistAvoRob(Z_r, B_r, θ_r, r_r^safe)

```

Θ_e^{avo} . The direction of motion, θ_r is then chosen such that the robot moves away from the boundary with a maximum possible step size, d_r while avoiding nearby robots (lines 1-2). The function **DistAvoRob** computes this maximum possible value of d_r from Eq. (5), as described in Section IV-B.

Once the robot is at a safe distance from the boundary, it then estimates the relative distance `DistToTar` (Section IV-A) from a target. If no target is sensed the robot performs a random walk while maintaining a safe distance from nearby robots (lines 4-10). If, on the other hand, the robot is inside the influence region of a target, it computes its current orbit, `currentOrbit` based on the estimated `DistToTar` and the input orbits. If the robot estimates that the relative distance between itself and the target is less than Or_0^{inner} , it computes the set of allowed heading direction,

Θ_g^{avo} , and chooses a direction of motion from this set while maximizing the step size to avoid nearby robots (lines 11-16). The distance to move away from a target is capped at $d_r^{\text{req}} = \|\text{DistToTar} - Or_0^{\text{inner}}\|$ (lines 15-16) to ensure that the robot doesn't move outside the primary orbit.

If the robot is in the primary orbit (line 17), it moves tangentially to the orbit Or_0 (heading direction chosen from the computed set Θ_g^{tan}) with a step size d_r such that it maintains a safe distance from the nearby robots and the target (lines 18-19). The function **DistAvoTar** computes the maximum possible value of d_r from Eq. (12), as described in Section IV-C.

When a robot is in a secondary orbit, $Or_{i>0}$ it chooses a heading direction from the set Θ_g^{LOS} to move towards the target while avoiding nearby robots (line 21-22). In case the robot cannot find a direction of motion to move a non-zero distance toward the target (line 23), it either moves a non-zero distance tangentially in its current orbit (lines 24-25) or moves a safe distance in a heading direction based on the reading from the sensor receiving the minimum signal strength z_r^k , i.e. the direction where the virtual source corresponding to other robots is the farthest (lines 26-29).

A robot's local control, as summarized in Algorithm 1, is agnostic to the motion type of the targets. This, together with our convergence guarantees in the following section, implies that our algorithm guarantees the encapsulation of multiple targets moving in the bounded environment with different types of motion models, as described in Section III.

V. CONVERGENCE GUARANTEES

We use the Lyapunov stability theory to provide guarantees on the emergent behavior of the swarm. In this section, for clarity, we consider circular robots with noiseless sensors. In practice, the desired behavior emerges for non-circular robots as well, which we demonstrate in simulations in Section VI.

Lemma V.1. [34] *A disc robot with a non-zero radius performing a random walk in a bounded 2D environment will almost surely approach the exploration of the entire area as time approaches infinity.*

Corollary V.1.1. *We consider a robot to be in the influence region of a target at time step T when $\|\mathbf{c}_{g,T} - \mathbf{c}_{r,T}\| < \beta_g + r_r$. From Lemma V.1 we can say with a probability of 1 that a robot performing a random walk will enter the influence region, $\beta_g > r_r$, of a static target as time goes to infinity. In the case of a dynamic target, we assume that the influence region of a target is large enough to ensure that a robot performing a random walk will enter its influence region within a finite time.*

Lemma V.2. *For any arbitrary initial condition such that a robot is at least r_g^{safe} away from a target, a necessary condition to ensure a collision-free target's motion is that the escape radius of the target, $r_g^{\text{escape}} \geq r_g^{\text{safe}} + d_g^{\text{max}}$ and the inner radius of the primary orbit, $Or_0^{\text{inner}} \geq r_g^{\text{safe}} + d_g^{\text{max}}$*

Proof. The lower bound on r_g^{escape} ensures that a target gets enough margin to escape an approaching robot. As described

in Section IV-C, a robot's behavior is such that it moves away from the target if the robot crosses the inner ring, Or_0^{inner} , of the primary orbit. Hence the above lower bound on Or_0^{inner} ensures that collision avoidance behavior for a robot is triggered before the distance between a target and a robot becomes r_g^{safe} . $\square$

Lemma V.3. *For any arbitrary initial condition such that a robot is at least r_g^{safe} away from a target the following are the necessary conditions to ensure a target's encapsulation:*

- 1) *the outer radius of the encapsulation ring A_g satisfies*

$$r_g^{\text{encap}} \geq d_r^{\text{max}} + r_r + \sqrt{(Or_0^{\text{inner}})^2 + r_r^2 - 2r_r Or_0^{\text{inner}} \cos(\pi/p)} \quad (13)$$

- 2) *the number of robots n_g specified for encapsulation satisfies,*

$$n_g \leq n_0 = \frac{2\pi}{\cos^{-1}\left(1 - \frac{(\beta_r + r_r)^2}{2(r_g^{\text{encap}})^2}\right)} \quad (14)$$

Proof. Each robot's estimate of the relative distance from a target depends on the sensor with the maximum reading, $\max(Z_g)$. Given that a virtual source is always either closer or at the radial location of an actual source, it is possible that even if a robot is present in the primary orbit, the robot estimates itself to be present at a relative distance of less than Or_0^{inner} with respect to the target. This will trigger the collision avoidance behavior for the robot and it will move away from the target. To successfully encapsulate a target g , it is required that the outer radius of the encapsulation ring, r_g^{encap} incorporate the robot with the worst possible estimate of the target's location. This will ensure that the robot remains in the primary orbit even after being over-cautious in moving away from the target.

At each time step, a robot chooses its control parameters such that it maintains at least a distance of Or_0^{inner} from a target, that is, $\|\mathbf{c}_{g,T} - \mathbf{c}_{r,T}\| \geq Or_0^{\text{inner}}$. Since Or_0^{inner} is defined between a robot's center and the target, we set $d_s = Or_0^{\text{inner}}$ in Eq. (3) to obtain $d_g^k = \sqrt{(Or_0^{\text{inner}})^2 + r_r^2 - 2r_r Or_0^{\text{inner}} \cos(\pi/p)}$. A robot will start to move away from the target when $\max(Z_g) \geq B_g(d_g^k)$. At this point, the upper bound on $\|\mathbf{c}_{g,T} - \mathbf{c}_{r,T}\|$ is given by Eq. (15). We can see that as $p \rightarrow \infty$, $\|\mathbf{c}_{g,T} - \mathbf{c}_{r,T}\| \rightarrow Or_0^{\text{inner}}$ and for a finite p , collision avoidance behavior is triggered before the robot is at a distance of Or_0^{inner} from the target.

$$\|\mathbf{c}_{g,T} - \mathbf{c}_{r,T}\| \leq r_r + \sqrt{(Or_0^{\text{inner}})^2 + r_r^2 - 2r_r Or_0^{\text{inner}} \cos(\pi/p)} \quad (15)$$

For asymmetric sensor placement, we replace π/p with half of the maximum angle between two adjacent sensors on the robot.

To incorporate the robot with the worst estimate of a target's location, we set a lower bound on the outer radius of the encapsulation ring using Eq. (15) as $r_g^{\text{encap}} \geq r_r + \sqrt{(Or_0^{\text{inner}})^2 + r_r^2 - 2r_r Or_0^{\text{inner}} \cos(\pi/p)}$. Since the robot may chatter in the encapsulation ring due to constant attraction and repulsion from the target and nearby robots, we add d_r^{max}

to the lower bound on r_g^{encap} (condition 1). This will ensure that a robot remains in the encapsulation ring when there are other robots nearby.

Furthermore, the maximum number of robots that can be specified for target encapsulation (condition 2) is bounded by the total number of robots that can be physically placed in the encapsulation ring such that the encapsulating robots are outside each other's influence region to ensure no chattering that can be caused by repulsion from each other. When $n_g > n_0$ a dynamic equilibrium exists around a target such that there are always almost n_0 robots present in the primary orbit [5]. $\square$

As described in Section III, we consider three types of motion patterns that a target can exhibit. For each of these target motion patterns, we provide guarantees for liveness (eventual encapsulation) based on the Lyapunov stability theory and stochastic analysis (Lemmas V.4 - V.6).

Lemma V.4. Consider a target $g \in \mathcal{G}$ moving randomly in the bounded environment until it senses any robot in its escape domain (as described by motion model 2 in Section III). If,

- 1) the maximum step size of the target, $d_g^{\max} \leq \lambda d_r^{\max}$ where

$$\begin{aligned} \lambda &= \min\left(\frac{\pi}{2}, \frac{\alpha}{\sin(\pi - \alpha)}\right) \frac{\sin\varphi}{\varphi} \cos\varphi \\ \alpha &= \cos^{-1}\left(1 - \frac{(\beta_r + r_r)^2}{2(r_g^{\text{escape}})^2}\right) \\ \varphi &= \frac{\max(\phi^k - \phi^{k+1})}{2}, \quad k = \{1 \dots p\} \\ &= \pi/p, \quad \text{for symmetric sensor placement} \end{aligned}$$

the target g will be encapsulated eventually.

Proof. Consider a target $g \in \mathcal{G}$. Let $\mathbf{u}_g = [d_g \cos(\gamma_{g,T} + \theta_g) \quad d_g \sin(\gamma_{g,T} + \theta_g)]$ and $\mathbf{u}_r = [d_r \cos(\gamma_{r,T} + \theta_r) \quad d_r \sin(\gamma_{r,T} + \theta_r)]$ be the control input of a target and robot respectively at time T , and η be the total robots currently present in the escape domain of the target, that satisfy $\|\mathbf{c}_{g,T} - \mathbf{c}_{r,T}\| \leq r_g^{\text{escape}}$. As outlined in Section IV-C, the motion strategy of a robot can be broken down as follows:

Case I: Robot is in an orbit $Or_{i \geq 0}$ such that $\eta = 0$

We use the definition of stochastic stability in the sense of Lyapunov [35] to show that a robot eventually reaches the primary orbit Or_0 . Let $V = \|\mathbf{c}_{g,T} - \mathbf{c}_{r,T}\|^2$ be the candidate Lyapunov function defined on the domain $D_r \subseteq \mathbb{R}^2$ such that $\|\mathbf{c}_{g,T} - \mathbf{c}_{r,T}\| \geq r_g^{\text{encap}}$. Using Eq. (2) and Eq. (1) we have, $\Delta V = \|(\mathbf{c}_{g,T} + \mathbf{u}_g) - (\mathbf{c}_{r,T} + \mathbf{u}_r)\|^2 - \|\mathbf{c}_{g,T} - \mathbf{c}_{r,T}\|^2$. For ease of exposition, we will drop the subscript T in the following analysis. On simplifying, $\Delta V = \|\mathbf{u}_g - \mathbf{u}_r\|^2 + 2(\mathbf{c}_g - \mathbf{c}_r) \cdot (\mathbf{u}_g - \mathbf{u}_r)$. Fig. 8 depicts the relative kinematics model between the target and a robot where ω is the angle that the LOS vector, $(\mathbf{c}_g - \mathbf{c}_r)$ makes with the x-axis. Let $\hat{\mathbf{l}} = [\cos\omega \quad \sin\omega]$ be the vector along $(\mathbf{c}_g - \mathbf{c}_r)$ and $\hat{\mathbf{t}} = [-\sin\omega \quad \cos\omega]$ be the vector tangential to it. Then,

$$\Delta V = d_g^2 + d_r^2 - 2\mathbf{u}_g \cdot \mathbf{u}_r + 2\|\mathbf{c}_g - \mathbf{c}_r\| \hat{\mathbf{l}} \cdot (\mathbf{u}_g - \mathbf{u}_r) \quad (16)$$

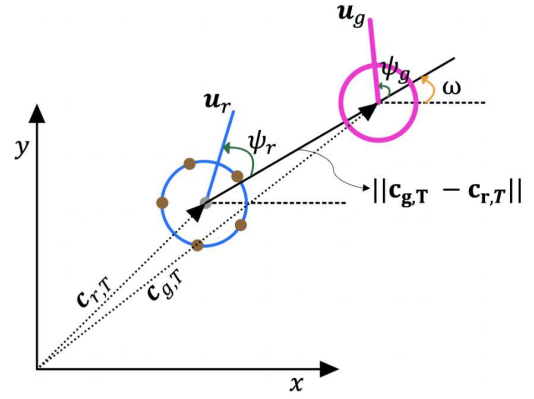


Fig. 8: Relative kinematics of a robot-target interaction.

(a) Robot is in a secondary orbit: For this case, a robot would move towards the target, that is $\theta_r \in \Theta_g^{\text{LOS}}$ given in Eq. (8). If $\eta = 0$, that is, there are no robots in the target's escape domain, the target moves randomly. Hence, $\theta_g \in [0 \ 2\pi)$. That is, both θ_g and θ_r are stochastic. Moreover, the control inputs $\mathbf{u}_g$ and $\mathbf{u}_r$ are independent random vectors and their corresponding expected values are given by,

$$\begin{aligned} \mathbb{E}[\mathbf{u}_g] &= d_g \mathbb{E}[\cos(\gamma_g + \theta_g) \quad \sin(\gamma_g + \theta_g)] \\ &= d_g \left[\int_0^{2\pi} \cos(\gamma_g + \theta_g) \frac{1}{2\pi} d\theta_g \quad \int_0^{2\pi} \sin(\gamma_g + \theta_g) \frac{1}{2\pi} d\theta_g \right] \\ &= 0 \end{aligned} \quad (17)$$

$$\begin{aligned} \mathbb{E}[\mathbf{u}_r] &= d_r \mathbb{E}[\cos(\gamma_r + \theta_r) \quad \sin(\gamma_r + \theta_r)] \\ &= d_r \left[\int_{\phi_k - \frac{\pi}{p}}^{\phi_k + \frac{\pi}{p}} \cos(\gamma_r + \theta_r) \frac{1}{2\pi/p} d\theta_r \quad \int_{\phi_k - \frac{\pi}{p}}^{\phi_k + \frac{\pi}{p}} \sin(\gamma_r + \theta_r) \frac{1}{2\pi/p} d\theta_r \right] \\ &= d_r \frac{\sin\varphi}{\varphi} [\cos(\gamma_r + \phi_k) \quad \sin(\gamma_r + \phi_k)] \\ &= d_r \frac{\sin\varphi}{\varphi} \hat{\mathbf{u}}_r^k \end{aligned} \quad (18)$$

where $\varphi = \pi/p$ for symmetric sensor placement and $\hat{\mathbf{u}}_r^k$ is the unit vector in the direction of k^{th} sensor. Intuitively, this means that on an average the robot moves in the direction of the k^{th} sensor (receiving maximum intensity from the target) with a step-size reduced by the factor $\sin\varphi/\varphi$. Furthermore, as $p \rightarrow \infty$, $\mathbb{E}[\mathbf{u}_r] \rightarrow d_r \hat{\mathbf{u}}_r^k$. That is, if the robot knows the exact relative location of the target, it moves towards the target along the line of sight vector with the maximum possible step size.

Using Eq. (17) and Eq. (18), the expected value of change in the Lyapunov function (as given by Eq. (16)) between two

consecutive time steps is,

$$\begin{aligned}\mathbb{E}[\Delta V] &= d_g^2 + d_r^2 + 2\mathbb{E}[\|\mathbf{c}_g - \mathbf{c}_r\| \hat{\mathbf{l}} \cdot (\mathbf{u}_g - \mathbf{u}_r)] \\ &= d_g^2 + d_r^2 + 2\|\mathbf{c}_g - \mathbf{c}_r\| (\mathbb{E}[\mathbf{u}_g] - \mathbb{E}[\mathbf{u}_r]) \cdot \hat{\mathbf{l}} \\ &= d_g^2 + d_r^2 - 2\|\mathbf{c}_g - \mathbf{c}_r\| d_r \frac{\sin\varphi}{\varphi} \hat{\mathbf{u}}_r^k \cdot \hat{\mathbf{l}}\end{aligned}\quad (19)$$

The maximum deviation of the unit vector in the direction of the k^{th} sensor, $\hat{\mathbf{u}}_r^k$ from the LOS vector $\hat{\mathbf{l}}$ is limited to $\varphi = \pi/p$ (refer to Fig. 2), that is $\hat{\mathbf{u}}_r^k \cdot \hat{\mathbf{l}} \geq \cos\varphi$. Furthermore, when a robot is in a secondary orbit $\|\mathbf{c}_g - \mathbf{c}_r\| \geq r_g^{\text{encap}}$. Substituting these bounds in Eq. (19) we have,

$$\mathbb{E}[\Delta V] \leq d_g^2 + d_r^2 - 2r_g^{\text{encap}} d_r \frac{\sin\varphi}{\varphi} \cos\varphi$$

For stability, we require that $\mathbb{E}[\Delta V] \leq 0$. That is

$$d_r^2 - (2r_g^{\text{encap}} \frac{\sin\varphi}{\varphi} \cos\varphi) d_r + d_g^2 \leq 0 \quad (20)$$

The necessary condition to satisfy the above inequality is that $\cos\varphi > 0$. That is, for symmetric sensor placement, the total number of sensors on a robot must be greater than or equal to three. For asymmetric sensor placement, this condition is a design parameter to choose the maximum distance φ between any two adjacent sensors. The roots of the above quadratic inequality in d_r give us an upper bound on the maximum step-size of a robot d_r^{max} which is always larger than the bound determined in Eq. (6) and hence Eq. (20) is always satisfied.

(b) Robot is in the primary orbit: Once a robot reaches the primary orbit, it moves tangentially to the orbit while ensuring that $\|\mathbf{c}_g - \mathbf{c}_r\| \geq Or_0^{\text{inner}}$. To analyze this, we look at how the LOS vector between a target and robot changes between two time steps, which is given by $\Delta(\mathbf{c}_g - \mathbf{c}_r) = \mathbf{u}_g - \mathbf{u}_r$. In Eq. (21) and Eq. (22) we define u_{gr}^{rad} and u_{gr}^{tan} representing the polar coordinates corresponding to the radial and tangent component of the change in LOS vector in global frame $\mathcal{I}$.

$$u_{gr}^{\text{rad}} = (\mathbf{u}_g - \mathbf{u}_r) \cdot \hat{\mathbf{l}} \quad (21)$$

$$u_{gr}^{\text{tan}} = (\mathbf{u}_g - \mathbf{u}_r) \cdot \hat{\mathbf{t}} \quad (22)$$

As explained earlier, to ensure a target's encirclement, it is necessary that a robot is able to complete a revolution around the target in the primary orbit Or_0 . We evaluate this stochastically by computing the expected value of the change in the tangential component of the relative LOS vector, $\mathbb{E}[u_{gr}^{\text{tan}}] = \mathbb{E}[\mathbf{u}_g] \cdot \hat{\mathbf{t}} - \mathbb{E}[\mathbf{u}_r] \cdot \hat{\mathbf{t}}$. For a clockwise orbital rotation, $\theta_r \in \Theta_g^{\text{tan}, -1}$. Since $\eta = 0$ for this case, $\theta_g \in [0, 2\pi)$. Simplifying and substituting Eq. (17) in the above equation we have

$$\mathbb{E}[u_{gr}^{\text{tan}}] = -d_r \left[\int_{\phi_k - \frac{\pi}{p} + \frac{\pi}{2}}^{\phi_k + \frac{\pi}{p} + \frac{\pi}{2}} \cos(\gamma_r + \theta_r) \frac{1}{2\pi/p} d\theta_r \right. \\ \left. \int_{\phi_k - \frac{\pi}{p} + \frac{\pi}{2}}^{\phi_k + \frac{\pi}{p} + \frac{\pi}{2}} \sin(\gamma_r + \theta_r) \frac{1}{2\pi/p} d\theta_r \right] \cdot \hat{\mathbf{t}}$$

$$\begin{aligned}&= -d_r \frac{\sin\varphi}{\varphi} [-\sin(\gamma_r + \phi^k) \quad \cos(\gamma_r + \phi^k)] \cdot \hat{\mathbf{t}} \\ &= -d_r \frac{\sin\varphi}{\varphi} \hat{\mathbf{u}}_r^k \cdot \hat{\mathbf{l}}\end{aligned}\quad (23)$$

Eq. (23) shows that a robot moves clockwise in the primary orbit with an expected tangential step-size of $d_r \frac{\sin\varphi}{\varphi} \cos(\varphi)$ with respect to the target.

Case II: $\|\mathbf{c}_{g,T} - \mathbf{c}_{r,T}\| \leq r_g^{\text{escape}}$

At a timestep T , it is possible that a robot is marginally outside the escape domain of the target but is unable to move in due to the presence of a nearby robot at a distance of r_r^{safe} .

This behavior could lead to $\|\mathbf{c}_{g,T+1} - \mathbf{c}_{r,T+1}\| < r_g^{\text{escape}}$, resulting in $\eta > 0$. The target would then move such that it can escape from all the robots present in its escape domain. Let ψ_g be the angle between a target's intended heading and the LOS vector $(\mathbf{c}_{g,T} - \mathbf{c}_{r,T})$, as shown in Fig. 8. If $\eta = 1$, then $\psi_g \in [3\pi/2, \pi/2]$. If $\eta = 2$ and α is the angle that these robots subtend at the center of the target, as shown in Fig. 9, then $\psi_g \in [3\pi/2 + \alpha, \pi/2]$ or $\psi_g \in [3\pi/2, \pi/2 - \alpha]$, depending on which robot ψ_g is measured with respect to. Without loss of generality, we can consider one of these. That is, if $\eta > 1$, the available angular range of the target for escaping decreases from π to $(\pi - (\eta - 1)\alpha)$.

To determine α , we use the fact that robots in an orbit disperse such that, on average, they are outside each other's influence region. Then, using geometry shown in Fig. 9, $\alpha = \cos^{-1} \left(1 - \frac{(\beta_r + r_r)^2}{2(r_g^{\text{escape}})^2} \right)$. Note that, the maximum escaping angular range of a target is limited to π (when $\eta = 1$). Hence, for $\eta > \pi/\alpha$, the target can no longer escape with the maximum possible step size. For **Case II**, we have to

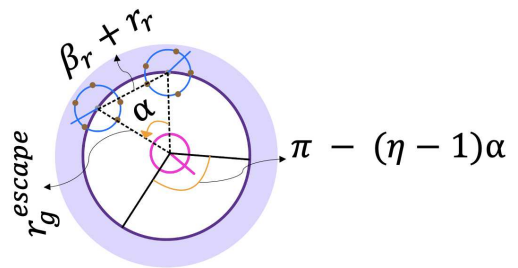


Fig. 9: Geometric configuration for computing bounds on the ratio between target and robot step-sizes.

ensure that (i) robots implementing tangential control law in the primary orbit are able to encircle the target, that is $\mathbb{E}[u_{gr}^{\text{tan}}] \leq 0$ if $\theta_r \in \Theta_g^{\text{tan}, -1}$ and $\mathbb{E}[u_{gr}^{\text{tan}}] \geq 0$ if $\theta_r \in \Theta_g^{\text{tan}, 1}$, and (ii) robots in secondary orbits are able to move towards the target, that is $\mathbb{E}[u_{gr}^{\text{rad}}] \leq 0$ for $\theta_r \in \Theta_g^{\text{LOS}}$.

Using Eq. (22) and the tangential control input ($\theta_r \in$

$\Theta_g^{\tan,-1}$) for robots present in Or_0 we have,

$$\begin{aligned} \mathbb{E}[u_{gr}^{\tan}] &= d_g \mathbb{E}[\sin\psi_g] - \mathbb{E}[\mathbf{u}_r] \cdot \hat{\mathbf{t}} \\ &= d_g \int_{\frac{3\pi}{2} + (\eta-1)\alpha}^{\frac{\pi}{2}} \sin\psi_g \frac{1}{\pi - (\eta-1)\alpha} d\psi_g - \mathbb{E}[\mathbf{u}_r] \cdot \hat{\mathbf{t}} \\ &= d_g \frac{\sin((\eta-1)\alpha)}{(\pi - (\eta-1)\alpha)} - d_r \frac{\sin\varphi}{\varphi} \hat{\mathbf{u}}_r^k \cdot \hat{\mathbf{t}} \end{aligned}$$

To ensure clockwise orbital rotation, $\mathbb{E}[u_{gr}^{\tan}] \leq 0$, that is,

$$\begin{aligned} \max_{\eta \leq \pi/\alpha} \left(d_g \frac{\sin((\eta-1)\alpha)}{(\pi - (\eta-1)\alpha)} \right) &\leq \min_{\hat{\mathbf{u}}_r^k \cdot \hat{\mathbf{t}} \geq \cos\varphi} \left(d_r \frac{\sin\varphi}{\varphi} \hat{\mathbf{u}}_r^k \cdot \hat{\mathbf{t}} \right) \\ d_g &\leq \frac{\alpha}{\sin(\pi - \alpha)} d_r \frac{\sin\varphi}{\varphi} \cos\varphi \end{aligned} \quad (24)$$

Intuitively, $\mathbb{E}[\mathbf{u}_g \cdot \hat{\mathbf{t}}]$ is maximal when the target has the least freedom in choosing its motion, that is $\eta = \pi/\alpha$. Similarly, $\mathbb{E}[\mathbf{u}_r \cdot \hat{\mathbf{t}}]$ is minimal when the average heading direction, perpendicular to $\hat{\mathbf{u}}_r^k$, is deviated the most from $\hat{\mathbf{t}}$.

Now, we need to ensure that the robots in the secondary orbits move toward an escaping target. Using Eq. (21) and the LOS control input ($\theta_r \in \Theta_g^{\text{LOS}}$) for robots present in $Or_{i>0}$ we have,

$$\begin{aligned} \mathbb{E}[u_{gr}^{\text{rad}}] &= \mathbb{E}[\mathbf{u}_g \cdot \hat{\mathbf{t}}] - \mathbb{E}[\mathbf{u}_r] \cdot \hat{\mathbf{t}} \\ &= d_g \int_{\frac{3\pi}{2} + (\eta-1)\alpha}^{\frac{\pi}{2}} \cos\psi_g \frac{1}{\pi - (\eta-1)\alpha} d\psi_g - \mathbb{E}[\mathbf{u}_r] \cdot \hat{\mathbf{t}} \\ &= d_g \frac{1 + \cos((\eta-1)\alpha)}{(\pi - (\eta-1)\alpha)} - d_r \frac{\sin\varphi}{\varphi} \hat{\mathbf{u}}_r^k \cdot \hat{\mathbf{t}} \end{aligned}$$

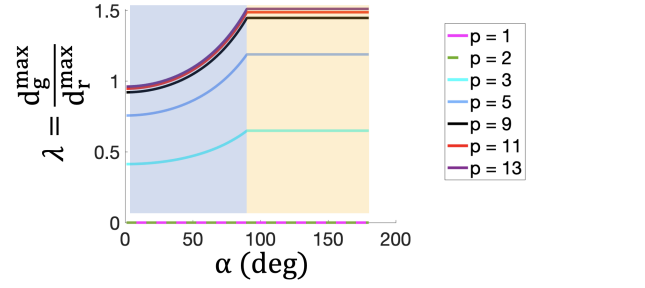
The distance between a target and robot will decrease if $\mathbb{E}[u_{gr}^{\text{rad}}] \leq 0$. That is,

$$\begin{aligned} \max_{\eta \leq \pi/\alpha} \left(d_g \frac{1 + \cos((\eta-1)\alpha)}{(\pi - (\eta-1)\alpha)} \right) &\leq \min_{\hat{\mathbf{u}}_r^k \cdot \hat{\mathbf{t}} \geq \cos\varphi} \left(d_r \frac{\sin\varphi}{\varphi} \hat{\mathbf{u}}_r^k \cdot \hat{\mathbf{t}} \right) \\ d_g &\leq \frac{\pi}{2} d_r \frac{\sin\varphi}{\varphi} \cos\varphi \end{aligned} \quad (25)$$

Eq. (24) and Eq. (25) determine an upper bound on the maximum step size of a target as given by condition (1).

In Fig. 10 we show how the ratio of the step-size between a target and robot, λ , changes with the number of sensors p and $\angle\alpha$ (which is proportional to how well the robots disperses in the primary orbit). As derived above, $\lambda = \min\left(\frac{\pi}{2}, \frac{\alpha}{\sin(\pi-\alpha)}\right) \frac{\sin(2\pi/p)}{2\pi/p}$. For a given number of sensors on a robot, λ increases with an increase in $\angle\alpha$ until $\pi/2 > \frac{\alpha}{\sin(\pi-\alpha)}$. We can also see that with an increase in the number of sensors on a robot, p , the ratio of the step size of a target to a robot increases and tends to $\pi/2 > 1$, that is, we can guarantee convergence (encapsulation) even when the target moves faster than the robots in the swarm. Previous approaches in the literature typically assume that the target moves slower than the robots [9]–[13].

The increased accuracy in the estimation of the relative location of nearby robots and target enables the robot to



$$\lambda = \frac{\alpha}{\sin(\pi - \alpha)} \frac{\sin(2\pi/p)}{2\pi/p} \quad \lambda = \frac{\pi \sin(2\pi/p)}{2} \frac{1}{2\pi/p}$$

Fig. 10: For a given number of sensors on a robot p , λ increases with an increase in $\angle\alpha$ until $\pi/2 > \frac{\alpha}{\sin(\pi-\alpha)}$. The ratio between the step-size of a target and robot tends to $\pi/2$ with an increasing p indicating that the swarm can encapsulate a target moving faster than the individual robots in the swarm.

disperse quickly in the primary orbit with less chattering. Apart from the more accurate estimation, with an increase in total sensors on a robot, the sensing radius β_r of a robot increases (Eq. (7)), which enables quick dispersion because of a robot's behavior of remaining outside other robots' influence region. This results in blocking the escaping paths of the target efficiently. The ratio between the target and robot step-sizes is zero when the robot has less than three sensors, for all values of $\angle\alpha$, implying that a minimum of three sensors are required to encapsulate a moving target.

Absence of livelocks and encapsulation of the target g : Similar to Lemma V.3, we can compute the total number of robots, n_i , that can be simultaneously present in an i^{th} orbit. If at a time step T there are less than n_i robots in Or_i , empty spots that could potentially be occupied by nearby robots, will be present in this orbit. As discussed in Section IV-C, the robots in the influence of a target either move toward the target or move, typically, in opposite tangential directions in adjacent orbits. This ensures that a dynamic empty spot present in an orbit Or_i and the robots present in Or_{i+1} move so as to align with each other. As we proved above, the robots present in the encapsulation ring (or the primary orbit) are guaranteed to continuously orbit the target. So, when there are less than n_g robots in the encapsulation ring, a dynamic empty spot is present in the primary orbit which will be eventually occupied by a robot orbiting in the secondary orbit Or_1 . When either all the empty spots in the primary orbit are filled by the robots or there are at least n_g robots in it, a target will be encapsulated. Once that happens we have from assumption (4) that the target will stop emitting its signal and set its control parameters to zero thereafter. All the robots that were in secondary orbits and in the influence of this target will transition into random walk behavior. Hence assumption (4) ensures that the robots would not be stuck in the secondary orbits of an encapsulated target and can transition into target-searching behavior after one target is encapsulated.

□

Lemma V.5. Consider a swarm with a total of n robots and a target $g \in \mathcal{G}$ moving randomly in the bounded environment (as described by motion model 1 in Section III). If,

- 1) the inner radius of the primary orbit $Or_0^{\text{inner}} \geq r_g^{\text{safe}} + d_r^{\text{max}}$
- 2) the maximum step size of the target, $d_g^{\text{max}} \leq \lambda d_r^{\text{max}}$ where

$$\lambda = \left(n - \left\lfloor \frac{2\pi}{\cos^{-1}\left(1 - \frac{(\beta_r + r_r)^2}{2(Or_0^{\text{inner}})^2}\right)} \right\rfloor + 1 \right)^{-1}$$

the target g will be encapsulated eventually.

Proof. The challenge in encapsulating a randomly moving target is in ensuring that robots avoid colliding with a target. For example, say at time T a robot i is in the primary orbit sandwiched between a target on one side at a distance of $r_g^{\text{safe}} + d_g^{\text{max}}$ and a robot j at a distance of r_r^{safe} on the other side, along the target's LOS vector. Furthermore, consider that for time steps T until $T+2$, the randomly moving target acts adversarial by trying to collide with the robot. That is, at every time step it moves towards robot i .

To ensure that the i^{th} robot avoids colliding with the target at $T+1$, it must choose a heading direction $\gamma_i \in \Theta_g^{\text{avo}}$. However, due to the presence of robot j , it cannot move a nonzero distance at time T in the intended heading direction. Hence the robot would violate the target-robot safety specification at $T+1$. Now, at T , say the j^{th} robot had moved away from robot i , implying that at $T+1$, the i^{th} robot will not sense the j^{th} robot and will be free to move away from the target. That is, it took a minimum of two time steps for the i^{th} robot to move away from the target. Hence, to ensure safety for this scenario, $d_g^{\text{max}} \leq d_r^{\text{max}}/2$ and $Or_0^{\text{inner}} \geq r_g^{\text{safe}} + d_r^{\text{max}}$ (condition 1).

Generalizing this, let the total robots present in the environment be n and $\tilde{n}_0 = \left\lfloor \frac{2\pi}{\cos^{-1}\left(1 - \frac{(\beta_r + r_r)^2}{2(Or_0^{\text{inner}})^2}\right)} \right\rfloor$ be the number

of robots that can be simultaneously present in the primary orbit marginally outside Or_0^{inner} without repelling each other. Then, in the worst case scenario, the total time steps for which a robot present on Or_0^{inner} may have to remain idle ($d_r = 0$) is $(n - \tilde{n}_0 + 1)$. This follows from the fact that only $(n - \tilde{n}_0)$ number of robots contribute to the idle waiting time of a robot in the primary orbit present on Or_0^{inner} . This constraint on the idle time of a robot in the primary orbit gives us an upper bound λ (condition 2) on how slow an adversarial target needs to be with respect to a robot to ensure safety. The analysis for stability and encapsulation follows from Lemma V.4. $\square$

Lemma V.6. Consider a target $g \in \mathcal{G}$ moving in an unknown pattern until it senses a robot in its escape domain (as described by motion model 3 in Section III). If,

- 1) the maximum step size of the target when moving in an unknown pattern, $d_g^{\text{max}} < \lambda d_r^{\text{max}}$ where

$$\lambda = \frac{\sin\varphi}{\varphi} \cos(\varphi),$$

$$\varphi = \pi/p, \quad \text{for symmetric sensor placement}$$

- 2) the maximum step-size of the target when escaping nearby robots,

$$\lambda = \min\left(\frac{\pi}{2}, \frac{\alpha}{\sin(\pi - \alpha)}\right) \frac{\sin\varphi}{\varphi} \cos\varphi$$

$$\alpha = \cos^{-1}\left(1 - \frac{(\beta_r + r_r)^2}{2(r_g^{\text{escape}})^2}\right)$$

$$\varphi = \frac{\max(\phi^k - \phi^{k+1})}{2}, \quad k = \{1 \dots p\}$$

$$= \pi/p, \quad \text{for symmetric sensor placement}$$

the target g will be encapsulated eventually.

Proof. This scenario is comparable to hunting problems [22], [23] where the target moves at slower speeds in some unknown motion pattern. But as soon as it detects (target's sensing limited to r_g^{escape}) a predator (robot) in its domain, it escapes at a faster speed than the predator. To ensure that robots in the secondary orbits move toward the target, we require that,

$$\mathbb{E}[\mathbf{u}_g \cdot \hat{\mathbf{l}}] - \mathbb{E}[\mathbf{u}_r \cdot \hat{\mathbf{l}}] < 0$$

$$d_g(\hat{\mathbf{u}}_g \cdot \hat{\mathbf{l}}) < d_r \frac{\sin\varphi}{\varphi} \cos\varphi \quad (26)$$

Eq. (26) is always satisfied if $d_g < d_r \frac{\sin\varphi}{\varphi} \cos\varphi$. It is trivial to show using Eq. (22) and Eq. (18) that the constraint $d_g < d_r \frac{\sin\varphi}{\varphi} \cos\varphi$ also ensures that robot in a primary orbit will encircle the target. As shown in Lemma V.4, the robots can successfully encapsulate an escaping target as long as a target step-size is within the bounds given by Eq. (25) and Eq. (24). $\square$

VI. SIMULATION RESULTS

In this section, we study the effect of the total number of sensors p , target-robot step-size ratio λ , and noisy sensors on the global behavior of the swarm. For each case, we consider three different target motion models: (i) target performs random walk in the environment, (ii) target perform random walk until there exists a robot such that $r_{gr} \leq r_g^{\text{escape}}$, (iii) target moves with constant velocity until there exists a robot such that $r_{gr} \leq r_g^{\text{escape}}$. The simulation environment consists of one moving target and ten robots. The total time is capped at 4000 time-steps. Due to the inherent randomness in the motion of the robots and targets, we ran 50 simulations for each data point with the same initial conditions. All the robots were initialized arbitrarily such that they lie in a sector of $\pi/4$ with respect to the target's center. This is done to show the ability of the swarm to successfully encapsulate a target when it has all the escape paths open.

Effect of the total number of sensors: For a given p , with an increase in the radius of the escape domain of a target, the bound on the maximum step size of a target decreases as shown in Fig. 11. This is because, with an increase in r_g^{escape} , a target gets a higher margin for escaping. From Lemmas V.4 - V.6, the ratio λ , and hence the target's step size, is dependent on the total number of sensors on a robot, p , and r_g^{escape} . When the escape domain of the target, $r_g^{\text{escape}} \leq r_g^{\text{safe}} + \frac{\pi}{2} d_r^{\text{max}}$, an increase in the total number

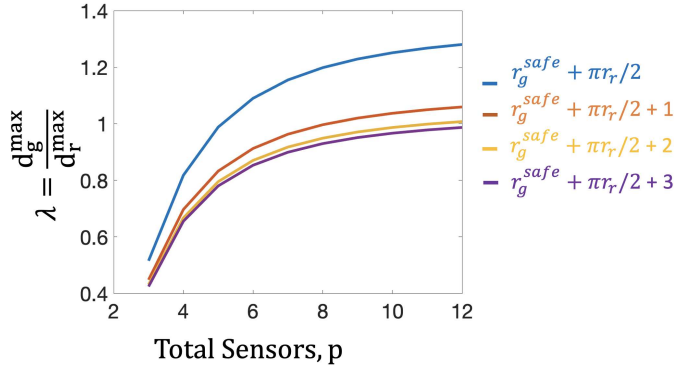


Fig. 11: For a given p , the ratio between the step-size of a target and robot decreases with an increase in r_g^{escape} , indicating that as the target is able to detect robots sooner, to ensure encapsulation it must also move slower.

of sensors on a robot enables the swarm to capture a faster moving target. This can be seen in Fig. 11, where the blue line corresponds to $r_g^{\text{escape}} = r_g^{\text{safe}} + \frac{\pi}{2}r_r$. As p increases, d_r^{max} tends to r_r and λ becomes greater than 1. If the escape domain is further increased, that is $r_g^{\text{escape}} > r_g^{\text{safe}} + \frac{\pi}{2}d_r^{\text{max}}$, then d_g^{max} decreases proportionally because a robot's step size is limited to d_r^{max} . For a given n_g , r_g^{escape} and r_g^{encap} , Fig. 12 shows how varying the total number of sensors, and hence the target-robot step-size ratio λ , affects the total time taken for target encapsulation for each type of target motion model.

Effect of noisy sensors: To study the effect of noise we added Gaussian noise to each sensor reading, $z_s^k = (1 - n_s^k) \sum_{j \in N_s^k} B_s(d_j^k)$, $n_s^k \sim \mathcal{N}(0, \sigma^2)$ and $n_s^k \leq 1$. Similar to the results obtained in our previous work [5], for all noise levels, we did not observe any collision within the swarm. However, to ensure that a robot does not collide with a moving target or the environment boundary, we increase the radius of Or_0^{inner} in proportion to the standard deviation of the noise. Fig. 13a shows the total time taken by the swarm to encapsulate a target with $p = 7$ and $\lambda = 1.1549$. With an increase in noise level, a robot's estimate of the target's location becomes less accurate, leading to an increase in the total time taken for encapsulation. Furthermore, as can be seen in Fig. 13b for noise levels greater than 50%, the probability of success for target encapsulation drops to 40% when a target moves with constant velocity.

Effect of initial swarm coverage: We next study how the initial distribution of the swarm plays a role in encapsulating a faster-moving target. Given that our control strategy for a robot in the swarm is to encapsulate the target by moving in orbits and shutting off its escaping directions, we expect that the initial placement of the swarm around a target would affect how quickly encapsulation happens. We consider a single target in the environment with 10 robots, each equipped with 7 sensors such that $\lambda = 1.1549$ and run 50 simulations for each condition. Fig. 14b shows the initial swarm distribution with varying sector coverage around a

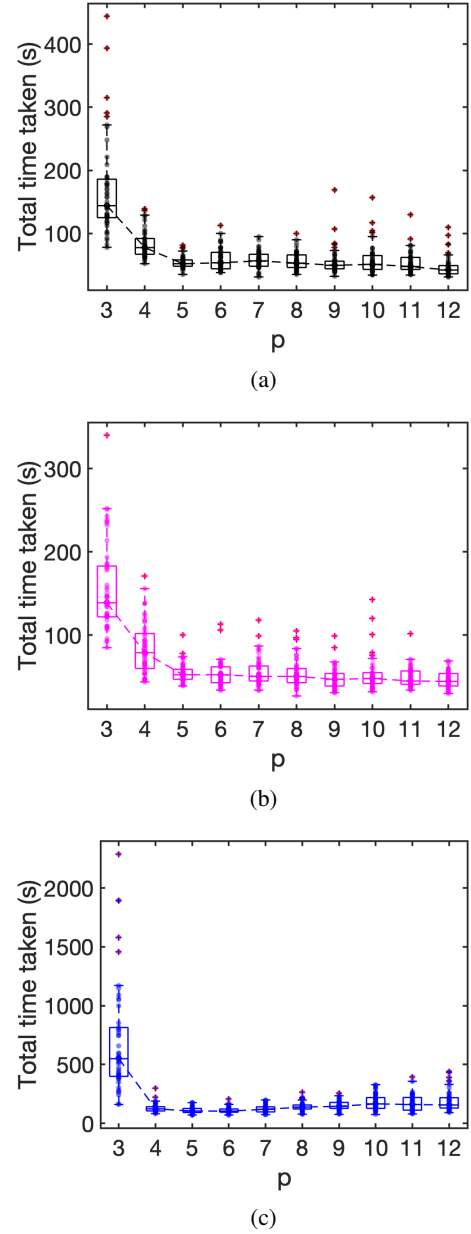


Fig. 12: The total time taken for task completion as a function of p such that (a) target performs random walk in the environment (b) target performs random walk while escaping from nearby robots (c) target moves with a constant velocity while escaping from nearby robots. The box plot shows median, 25th and 75th percentiles and the min/max values. The line connects the medians.

target with respect to its center. Fig. 14a shows that with an increase in initial coverage around a faster-moving target, encapsulation time decreases quickly. On increasing the initial swarm distribution from a sector of $\pi/8$ to 2π we observed a drop of 50% in the total time taken for encapsulation.

Comparison with algorithm in [5]: The algorithm we proposed in this paper is more efficient in terms of the total time taken by the swarm to encapsulate a static target as

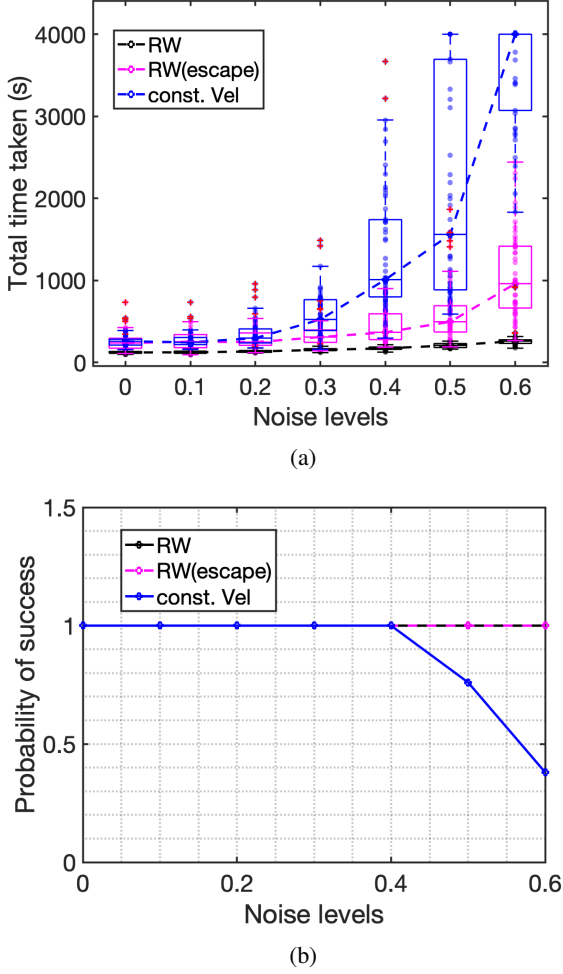


Fig. 13: (a) The total time taken for task completion as a function of noise levels for $p = 7$, $\lambda = 1.1549$ (total time capped at 4000 time-steps) and different target motion models (b) probability of success for task completion

compared with our previous method in [5]. This is due to the orbiting behavior of the swarm when a robot cannot move toward the target which results in a faster occupancy of empty spots in the encapsulation ring. This is shown in Fig. 15.

Scalability: In the supplementary video, we run additional simulations to show the effect of asymmetric sensor placement, the validity of our algorithm for non-circular robots and demonstrate the scalability of our algorithm with a large-scale simulation of 120 robots and 15 targets moving with different motion models.

Effect of targets with overlapping influence regions: In the supplementary video, we demonstrate the consequences of relaxing Assumption 2 without making any adjustments to the current algorithm. We observed instances of collisions between a robot and a dynamic target. This occurs because a robot determines the nearest possible location of a target based on the sensor receiving the maximum signal strength from that target (refer to Fig. 7). This approach is effective only when

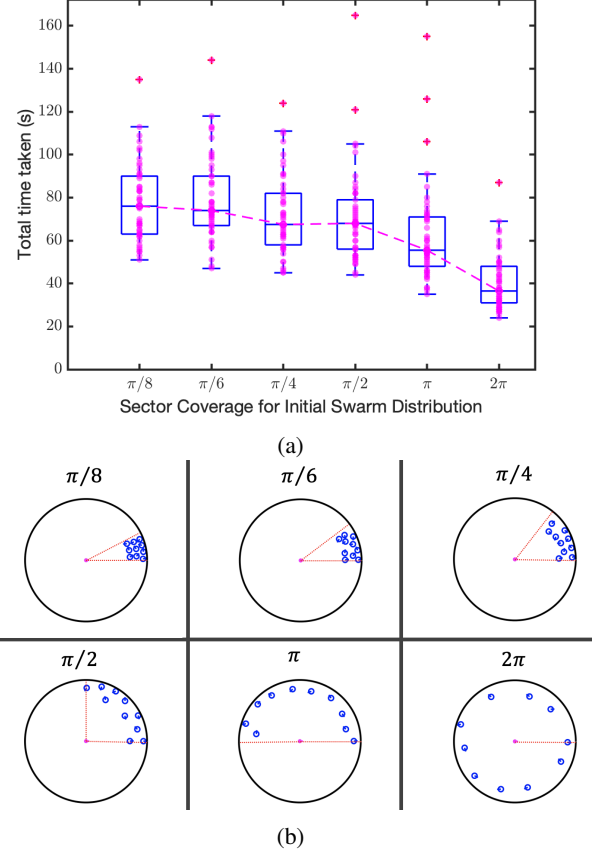


Fig. 14: (a) The total time taken for task completion decreases as the initial coverage of the swarm around a target increases. (b) Initial swarm distribution for different sector coverage around a target with respect to its center.

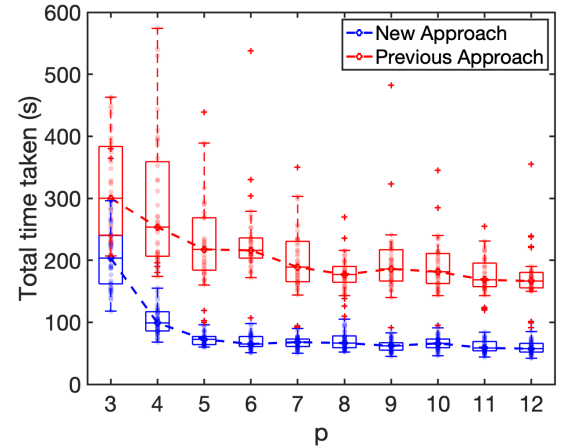


Fig. 15: For a given p , the total time taken to encapsulate a static target is lower for the new approach introduced in this paper as compared to our previous approach in [5].

Assumption 2 remains valid. However, if a robot can sense more than one target simultaneously, there is a possibility that in addition to the sensor receiving the maximum intensity from the targets, other sensors may detect signal strengths exceeding

the safety threshold. In such cases, the robot may collide with a target if readings from all sensors are not considered.

Conversely, we observed instances of deadlocks when we incorporated readings from all the sensors, particularly in scenarios where all sensors received readings exceeding the safety threshold. When utilizing Algorithm 1, we can address deadlocks in the case of dynamic robots by setting bounds on the influence region of a robot's source and leveraging the robots' active avoidance mechanisms. However, upper-bounding an adversarial target's influence region does not guarantee the absence of deadlocks, as we have no control over the targets' behavior.

An additional challenge that arises when a robot can sense multiple targets simultaneously, is estimating the direction of the target using the strategy outlined in this paper and subsequently computing orbits around it. For instance, if a robot is sensing two targets, the orbits will be adjusted towards a center position between these two targets, depending on the robot's proximity to them. This can lead to the robot moving towards a primary orbit that does not align with any specific target, resulting in no encapsulation. In our future work, we will investigate how the control algorithm should be modified to address deadlocks when relaxing Assumption 2.

VII. DISCUSSION AND CONCLUSIONS

In this paper, we propose a decentralized scalable algorithm for a minimalist swarm to encapsulate dynamic targets with unknown motion without requiring the exact knowledge of the relative positions or memory of the previous control inputs. We consider different scenarios of target motion and compute bounds on the target-robot step-size ratio to provide convergence guarantees. We observed the emergence of robots maintaining an approximate phase difference of $2\pi/n_g$ in the encapsulating ring, resulting in uniform distribution around the target and hence closing off its escaping directions. Furthermore, using extensive simulations we studied the effect of noisy sensors and showed the validity of our algorithm for non-circular robots. Our controller can be generalized for robots equipped with non-isotropic sensors which are not accurate in measuring the relative distances between two entities. If the bounds on the measurement error are known, our analysis can be used to compute bounds on the target-robot step-size ratio to ensure guaranteed target encapsulations. In this paper, we do not consider the presence of obstacles. However, the control algorithm we propose includes collision avoidance with both moving robots and targets in close proximity as well as static obstacles like the environment boundary. By conducting a similar analysis, this algorithm can be expanded to encompass other dynamic and static obstacles existing in the environment, as long as a robot's sensor can detect the signals emitted by these obstacles.

In our future work, we will explore how the control algorithm should change when we relax Assumption 2. Furthermore, we will implement our algorithm on physical robots and will also study the trade-off between incorporating the memory of previous states on the desired emergent behavior and providing timing bounds on task completion.

REFERENCES

- [1] L. Bayındır, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, pp. 292–321, 2016.
- [2] I. Slavkov, D. Carrillo-Zapata, N. Carranza, X. Diego, F. Jansson, J. Kaandorp, S. Hauert, and J. Sharpe, "Morphogenesis in robot swarms," *Science Robotics*, vol. 3, no. 25, p. eaau9178, 2018.
- [3] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 3293–3298.
- [4] L. Feola and V. Trianni, "Adaptive strategies for team formation in minimalist robot swarms," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4079–4085, 2022.
- [5] H. Sinhar and H. Kress-Gazit, "Decentralized control of minimalistic robotic swarms for guaranteed target encapsulation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 7251–7258.
- [6] A. Babić, I. Lončar, B. Arbanas, G. Vasiljević, T. Petrović, S. Bogdan, and N. Mišković, "A novel paradigm for underwater monitoring using mobile sensor networks," *Sensors*, vol. 20, no. 16, p. 4615, 2020.
- [7] C. Coquet, C. Aubry, A. Arnold, and P.-j. Bouvet, "A local charged particle swarm optimization to track an underwater mobile source," in *OCEANS 2019-Marseille*. IEEE, 2019, pp. 1–7.
- [8] C. Moeslinger, T. Schmickl, and K. Crailsheim, "A minimalist flocking algorithm for swarm robots," in *Advances in Artificial Life. Darwin Meets von Neumann*, G. Kampis, I. Karsai, and E. Szathmáry, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 375–382.
- [9] Z. Xue, J. Zeng, C. Feng, and Z. Liu, "Swarm target tracking collective behavior control with formation coverage search agents & globally asymptotically stable analysis of stochastic swarm," *J. Comput.*, vol. 6, no. 8, pp. 1772–1780, 2011.
- [10] G. Hollinger, S. Singh, J. Djgash, and A. Kehagias, "Efficient multi-robot search for a moving target," *The International Journal of Robotics Research*, vol. 28, no. 2, pp. 201–219, 2009.
- [11] X. Li, X. Liu, G. Wang, S. Han, C. Shi, and H. Che, "Cooperative target enclosing and tracking control with obstacles avoidance for multiple nonholonomic mobile robots," *Applied Sciences*, vol. 12, no. 6, p. 2876, 2022.
- [12] P. D. Hung, T. Q. Vinh, and T. D. Ngo, "A scalable, decentralised large-scale network of mobile robots for multi-target tracking," in *Intelligent Autonomous Systems 13*. Springer, 2016, pp. 621–637.
- [13] Q. B. Diep, I. Zelinka, and R. Senkerik, "An algorithm for swarm robot to avoid multiple dynamic obstacles and to catch the moving target," in *Artificial Intelligence and Soft Computing*, L. Rutkowski, R. Scherer, M. Korytkowski, W. Pedrycz, R. Tadeusiewicz, and J. M. Zurada, Eds. Cham: Springer International Publishing, 2019, pp. 666–675.
- [14] H. Zhao, D. Straub, and C. A. Rothkopf, "The visual control of interceptive steering: How do people steer a car to intercept a moving target?" *Journal of Vision*, vol. 19, no. 14, pp. 11–11, 12 2019. [Online]. Available: <https://doi.org/10.1167/19.14.11>
- [15] F. Belkhouche and B. Belkhouche, "A method for robot navigation toward a moving goal with unknown maneuvers," *Robotica*, vol. 23, no. 6, p. 709–720, nov 2005. [Online]. Available: <https://doi.org/10.1017/S0263574704001523>
- [16] J.-F. Xiong and G.-Z. Tan, "Virtual forces based approach for target capture with swarm robots," in *2009 Chinese Control and Decision Conference*. IEEE, 2009, pp. 642–646.
- [17] A. Benzerrouk, L. Adouane, and P. Martinet, "Dynamic obstacle avoidance strategies using limit cycle for the navigation of multi-robot system," in *2012 IEEE/RSJ IROS'12, 4th Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, 2012.
- [18] M. Wu, F. Huang, L. Wang, and J. Sun, "A distributed multi-robot cooperative hunting algorithm based on limit-cycle," in *2009 International Asia Conference on Informatics in Control, Automation and Robotics*, 2009, pp. 156–160.
- [19] Z. Sun, H. Sun, P. Li, and J. Zou, "Cooperative strategy for pursuit-evasion problem with collision avoidance," *Ocean Engineering*, vol. 266, p. 112742, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S002980182202025X>
- [20] M. V. Ramana and M. Kothari, "A cooperative pursuit-evasion game of a high speed evader," in *2015 54th IEEE Conference on Decision and Control (CDC)*, 2015, pp. 2969–2974.
- [21] X. Fang, C. Wang, L. Xie, and J. Chen, "Cooperative pursuit with multi-pursuer and one faster free-moving evader," *IEEE Transactions on Cybernetics*, vol. 52, no. 3, pp. 1405–1414, 2022.

- [22] L. Wu, "Hunting in unknown environments with dynamic deforming obstacles by swarm robots," *International Journal of Control and Automation*, vol. 8, no. 11, pp. 385–406, 2015.
- [23] O. Hamed and M. Hamlich, "Improvised multi-robot cooperation strategy for hunting a dynamic target," in *2020 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*. IEEE, 2020, pp. 1–4.
- [24] S. Manzoor, S. Lee, and Y. Choi, "A coordinated navigation strategy for multi-robots to capture a target moving with unknown speed," *Journal of Intelligent & Robotic Systems*, vol. 87, no. 3, pp. 627–641, 2017.
- [25] L. Blazovics, K. Csorba, B. Forstner, and C. Hassan, "Target tracking and surrounding with swarm robots," in *2012 IEEE 19th International Conference and Workshops on Engineering of Computer-Based Systems*. IEEE, 2012, pp. 135–141.
- [26] K. Xu, Y. Yang, and B. Li, "Brownian cargo capture in mazes via intelligent colloidal microrobot swarms," *Advanced Intelligent Systems*, vol. 3, no. 11, p. 2100115, 2021.
- [27] Y. Yang and M. A. Bevan, "Cargo capture and transport by colloidal swarms," *Science Advances*, vol. 6, no. 4, p. eaay7679, 2020. [Online]. Available: <https://www.science.org/doi/abs/10.1126/sciadv.aay7679>
- [28] G. Lee, N. Y. Chong, and H. Christensen, "Tracking multiple moving targets with swarms of mobile robots," *Intelligent Service Robotics*, vol. 3, no. 2, pp. 61–72, 2010.
- [29] A. Franchi, P. Stegagno, and G. Oriolo, "Decentralized multi-robot encirclement of a 3d target with guaranteed collision avoidance," *Autonomous Robots*, vol. 40, no. 2, pp. 245–265, 2016.
- [30] V. J. Lumelsky and K. Harinarayan, "Decentralized motion planning for multiple mobile robots: The cocktail party model," *Autonomous Robots*, vol. 4, no. 1, pp. 121–135, 1997.
- [31] J. A. Defay, A. Q. Nilles, and K. Petersen, "Characterization of the design space of collective braitenberg vehicles," in *16th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, 2022.
- [32] S. Maiti, S. Roh, I. Cohen, and N. L. Abbott, "Non-equilibrium ordering of liquid crystalline (lc) films driven by external gradients in surfactant concentration," *Journal of Colloid and Interface Science*, vol. 637, pp. 134–146, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S002197972202272X>
- [33] K. Henderson, K. Stadnikia, A. Martin, A. Enqvist, and S. Koppal, "Tracking radioactive sources through sensor fusion of omnidirectional lidar and isotropic rad-detectors," in *2017 International Conference on 3D Vision (3DV)*, 2017, pp. 97–106.
- [34] S. Popov, *Recurrence of two-dimensional simple random walk*, ser. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2021, p. 8–32.
- [35] Y. Li, W. Zhang, and X. Liu, "Stability of nonlinear stochastic discrete-time systems," *Journal of Applied Mathematics*, vol. 2013, 2013.