

Asymptotic Boundary Shrink Control with Multi-robot Systems

Shaocheng Luo, Jonghoek Kim, Byung-Cheol Min*

Abstract—Harmful marine spills, such as algae blooms and oil spills, damage ecosystems and threaten public health tremendously. Hence, an effective spill coverage and removal strategy will play a significant role in environmental protection. In recent years, low-cost water surface robots have emerged as a solution, with their efficacy verified at small scale. However, practical limitations such as connectivity, scalability, and sensing and operation ranges significantly impair their large-scale use. To circumvent these limitations, we propose a novel asymptotic boundary shrink control strategy that enables collective coverage of a spill by autonomous robots featuring customized operation ranges. For each robot, a novel controller is implemented that relies only on local vision sensors with limited vision range. Moreover, the distributedness of this strategy allows any number of robots to be employed without inter-robot collisions. Finally, features of this approach including the convergence of robot motion during boundary shrink control, spill clearance rate, and the capability to work under limited ranges of vision and wireless connectivity are validated through extensive experiments with simulation.

Index Terms—Multi-robot coordination, Spill removal, Boundary shrink control, Distributed control, Artificial potential field

I. INTRODUCTION

COVERAGE operation has a wide application scope in environment searching and exploration. Currently, methods of coverage control have seen tremendous potential in solving environmental issues caused by side effects of urbanization and industrialization. One global issue of increasing importance is water pollution, including harmful algae blooms that result from eutrophication, and oil leakage in the oceans during transportation, due to its connections with potable water supplies and even marine ecosystems. It is a natural idea to deploy a team of water surface robots for cleaning such pollution, in order to protect human operators and to provide for immediate operation response before any ecological or economic damage results.

Researchers from MIT developed one approach for the removal of oil spills in the ocean: Seaswarm, a fleet of low-cost, oil-absorbing robots [1]. These robots are powered by solar cells and designed to move on the water surface, absorbing oil spills through the long edge of an attached conveyor belt, as shown in Fig. 1. The Seaswarm robot

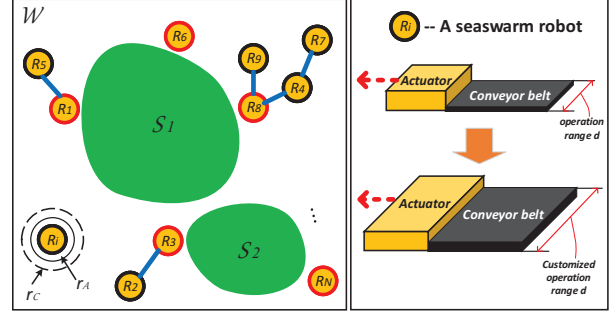


Fig. 1. A demonstrative figure showing the workspace $\mathcal{W}$ of spill coverage, where two spills, S_1 and S_2 , are surrounded by N number of distributed water surface robots such as MIT Seaswarm robots [1]. We consider to advance the Seaswarm robot by introducing a customized operation range d to adapt different operation scenarios, whose control and multi-robot coordination scheme is implemented in our proposed strategy. For each robot R_i , r_A and r_C represent its vision sensor range and the wireless communication range, respectively. The robots with red circles are in the vicinity of the spills, and the blue line represents a wireless connection. Note that the ranges and sizes of robot are not in real-scale.

has proven to be an effective prototype for spill cleaning operations; however, hardly any relevant research can be found that discusses control and collaboration strategies for this class of robot.

Outdoor coverage operations usually rely heavily on a global coordinate or external localization system, such as the Global Positioning System (GPS) [2]. However, robot control based on GPS sensors suffers from a low update rate (≤ 1 Hz), large and high-variance localization errors, and signal loss due to harsh weather. Moreover, in military scenarios, there may be cases where the GPS signal is jammed or not available due to an enemy's attack. Therefore, we develop a networked multi-robot system which does not rely on such external signals.

In this work, we propose an asymptotic boundary shrink control method using a team of robots that can collaboratively remove spills in diverse environments. We do not assume a global view of the workspace; instead, the robot team will first perform a random walk until each spill has been detected by at least one robot [3]. For each spill, the nearest robot becomes a rendezvous point for other robots by way of wireless connectivity mediated through on-board communication modules [4]. When any robot finishes migrating toward the rendezvous point to which it is allocated, this robot will begin boundary shrink control. When a boundary shrinks to zero and the robots meet each other, the associated spill has been covered and removed. During the boundary shrink process, we use local monocular vision sensors with a single line detection method to reduce reliance on GPS sensors and obtain more

Manuscript received...

This work was supported in part by NSF CAREER Award IIS-1846221.

S. Luo, and B.-C. Min are with SMART Lab, Department of Computer and Information Technology, Purdue University, West Lafayette, IN 47907, USA.

J. Kim is with Electrical and Electronic Convergence Department, Hongik University, Sejong-City, South Korea.

* Corresponding author email: minb@purdue.edu.

accurate control input.

To the best of our knowledge, the boundary shrink control method proposed in this work is the first to enable collective spill removal operation with multiple robots similar to Seaswarm [1], while also considering requirements such as adjustable robot operation ranges and constraints including limited vision and communication ranges, unavailability of global positioning systems, and collision avoidance. In addition to marine operations, the proposed coverage strategy can be applied in situations such as lawn mowing, forest fire-fighting, humanitarian de-mining, and more. As it does not rely on an external localization system, our strategy can even be applied for indoor coverage control, searching, and explorations.

Additionally, the distributedness of our method can tackle coverage situations where multiple discrete spills exist in the workspace. The efficacy of concurrently performing boundary shrink control on multiple spills is demonstrated in our simulations in Sec. VI. The proposed strategy achieves clearance rate of $> 99\%$ on multiple spills and with a variety of robot team settings.

The main contributions of this paper are threefold:

- This paper proposes a boundary shrink control method without path planning or localization;
- The proposed method can realize complete coverage of multiple spills in a distributed way, under practical constraints including customized robot operation ranges and limited communication and vision sensing ranges;
- The robot team allows strong scalability of the method.

The proposed coverage control strategy can also be applied in situations such as water quality monitoring [5], pollutant source searching [6], lawn mowing, forest fire-fighting, humanitarian de-mining, and others.

The paper is organized as follows: Sec. II reviews the relevant work and compares our method with the state-of-the-art research. Sec. III-A defines the research problem under the necessary assumptions and presents the proposed solution. Sec. III-B describes the control model for the robot, which is followed by a boundary shrink model, coverage manifold, and bounded moving speed during coverage based on robot processing capability. A multi-point multi-robot rendezvous strategy for boundary shrink control with limited wireless communication ranges is detailed in Sec. IV, while the boundary shrink control method is elaborated in Sec. V. Experiments with simulation and analyses of significant properties including boundary evolution, spill clearance rates, robot convergence, and operation time are elaborated in Sec. VI using the Robotarium testbed in MATLAB. Spills with non-convex coverage present challenges to the proposed solution, and these challenges are discussed in Sec. VII. This work is concluded in Sec. VIII.

II. RELATED WORK

Multi-robot systems have recently exhibited strong capability in environmental operations including cleaning oil spills [7] and fighting forest fires [8], which implicitly involves collective coverage of the area of interest.

The majority of research into multi-robot coverage operation focuses on path planning and robot distribution strategies, which are enabled by assuming a known environment or defined workspace. Coverage control using trapezoidal decompositions and Morse decompositions toward a defined area is discussed in [9] and [10], respectively, where a robot was designated to take care of a specific small area and travel in straight lines. Coverage of a water area was presented in [11], which required that each robot knows the planned path and its own accurate location. The work in [12] proposed to achieve target region coverage by way of motion planning based on triangulation. Multi-robot coverage with Boustrophedon decomposition is discussed in [13], which requires a pre-partitioned workspace for each robot as well as incessant localization. Strategies using pre-planned paths or motions can result in higher efficiency and less computational consumption for each robot, but in practice, the global coordinate frame and localization techniques for robot trajectory tracking are not always available, making tracking along the planned path unattainable. Moreover, the scale and shape of a real-world spill are usually unknown to the robot team, and coverage methods that depend on pre-planning are less adaptive to unknown environments or abrupt changes during the course of the operation.

In light of the aforementioned problems, our proposed method does not rely on any planning or global localization systems, such as [14], [15], and therefore can be applied in unknown or GPS-denied environments. We utilize a local monocular vision sensor with which each robot detects and tracks the spill boundary. However, the neighboring robot collision avoidance can be achieved using sonar [4], radar, or lidar sensors with 360 degrees scanning [16]. Therefore, the vision field or blind zone of vision sensors will not hinder the robot collision avoidance. A robot does not need to share its boundary detection information with others to construct a global map, as was done in [17]; instead, it can perform the coverage operation with a simple tracking model. Additionally, unlike other work involving the detection of a two-line boundary (left and right), similar to a driveway [18], [19], our method enables single line detection and so avoids massive computation.

Furthermore, many aforementioned coverage control methods feature centralized control schemes, such as those of [9], [10]. The Boustrophedon decomposition in [13] requires a pre-partitioned workspace for each robot, which limits its application scope. With centralized control, coverage performance depends largely on the master robot, whose decisions are based on feedback information from all slave robots. During operation, robots cannot be added to or removed from the task, meaning the system does not scale to different numbers of robots nor adapt to faulty robots. Our method advances the literature by nature of its complete distributedness, where each robot in a team can choose its own working state independent of its neighbours. Thus, the control scheme need not be updated according to the number of deployed robots.

Distributed coverage control methods based on Voronoi diagrams and Lloyd's algorithm have drawn attention in recent years. In papers such as [20], [21], a team of robots is

used to cover a specific area in a static manner once each robot is deployed to its goal position. A strategy based on a pivot robot is developed in [22]. The cooperative surveillance strategy from [23] can be used in spill removal, but it requires massive inter-robot communications and robust connections. The work in [24] proposed a moving coverage pattern where each agent i in the team has a specific sensor model and a defined sensory domain. Notably, all of these methods are primarily designed for information coverage, having multiple agents equipped with sensors, rather than for physical coverage such as is needed for spill removal. The works mentioned above, especially [7], [24], assumed each robot to be a point mass and did not elaborate the kinematics and control of the robots. Furthermore, they did not take into account practical limitations such as designated operation ranges, limited sensing ranges, and impediments to wireless connectivity. Our proposed boundary shrink control solution shows efficacy under these practical limitations, and is verified by extensive simulations referring to a Seaswarm robot prototype [1].

Our method also addresses problems such as the existence of multiple discrete spills in the workspace and collision avoidance for collaborating robots, which further strengthens the significance of our solution.

III. PROBLEM STATEMENT AND MATHEMATICAL FOUNDATION

This section structurally formulates the research problems in a mathematical way with reasonable assumptions.

A. Problem Statement

Given a workspace denoted as $\mathcal{W} \in \mathbb{R}^2$, as shown in Fig. 1, this study demonstrates how a certain number of randomly distributed robots can be controlled to implement coverage operation in the case of constraints such as limited vision sensing ranges and limited wireless signal ranges, while adapting to customized operation ranges. Meanwhile, this study reveals the optimal partitioned to the robots in a graph with several disconnected components and explains how they can collaborate in shrinking the spill boundary and complete coverage with only local image sensing.

As to the workspace, it is obstacle-free except for robots, and it is possible that many spills exist in the area of interest. Avoiding collision between robots coming from different spills can be handled by letting each robot use simple local collision avoidance controller, e.g., [25] or applying techniques such as stigmergic communication [26]. Thus, it is not within the research scope of this paper.

We assume a sufficient number of robots deployed in the coverage operation. However, if only a few robots are available, for instance less than the number of spills, they can be redeployed to cover the remaining spills by applying the boundary shrink control method. For each robot in a team, it begins a random walk from the same dock as the others and stops once detecting the boundary. The dock cannot be inside a spill, otherwise no robot can move outward to detect other spills. Furthermore, at the time when the random walk is ceased, no robot can be inside a spill, because it should have

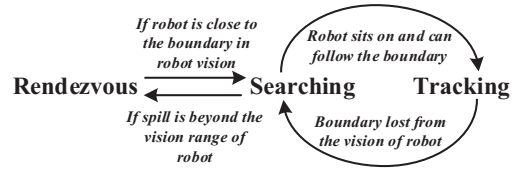


Fig. 2. The state diagram of the proposed multi-robot boundary shrink control method, and the state transition of robots in workspace $\mathcal{W}$. The collision avoidance hierarchy is prioritized as *Tracking* > *Searching* > *Rendezvous*.

stopped before traversing the boundary. We also assume that those robots that cannot see the spill are wirelessly connected to at least one other robot that can see the boundary.

As a solution, this paper first introduces a unicycle model for robot control, shown in Fig. 3(a). Based on this model, a hybrid control strategy is proposed for the robot team. The motion of each robot is represented as a finite state machine consisting of three individual states, as shown in Fig. 2, with the collision avoidance hierarchy being prioritized as *Tracking* > *Searching* > *Rendezvous*. A more detailed illustration of these states for one spill scenario is found in Fig. 4. The task is completed once there is no spill within the view of the robot.

B. Mathematical Foundation for Boundary Shrink Control Method and Robot

In order to show the performance of the robot in coverage, we first need to elaborate a robot control model and then robot control laws with a bounded speed. In this sub-section, the boundary shrink control model for spill coverage is presented.

1) *Vehicle Model*: The driving system that actuates the robot is composed of two unidirectionally placed parallel actuators (e.g., thrusters), which give the robot mobility with three degrees of freedom (DOF): x- and y-translation, and z-rotation. The robot control model plays an important role in determining the motion control laws. Assuming the center of buoyancy coincides with the center of gravity, and they are balanced with each other, it is concluded that all the robots move in a two-dimensional space $\mathbb{R}^2$. R_i denotes the i -th robot, here $i \in \{1, \dots, N\}$. Let the generalized coordinate of R_i be $\mathbf{q}_i = (x_i, y_i, \theta_i)$, we introduce a unicycle model [27], which has been validated previously through field tests using surface vehicles [28], [29], as below for the robot actuators shown in Fig. 3(a).

$$\dot{\mathbf{q}}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos \theta_i & 0 \\ \sin \theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \mathbf{J}(\theta_i) \mathbf{Q}_i \quad (1)$$

$$\mathbf{Q}_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(v_{ri} + v_{li}) \\ \frac{1}{L}(v_{ri} - v_{li}) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} v_{ri} \\ v_{li} \end{bmatrix} \quad (2)$$

where $\mathbf{Q}_i$ is the velocity vector and control objective of robot R_i consisting of linear velocity v_i and angular velocity ω_i with respect to the centroid. L in (2) denotes the tread width of the robot. The angular velocity ω_i is bounded by ω_{max} , which will be determined later in Sec. V-A. Furthermore, the angular controller for the robot is discrete in time and its resolution is $\phi_{ci} = \omega_i T_c$, where T_c is the system cycle. Since the robot is driven by the two parallel actuators, the control objective

space of $\mathbf{Q}_i$ has to be transformed into the space of v_{li} and v_{ri} . Additionally, we formulate control input as $[u_i, w_i]$, to differentiate from the velocity $[v_i, \omega_i]$. $[u_i, w_i]$ is the control input set as a “high level control”, and $[v_i, \omega_i]$ represents the actual motion of the robot due to the control input.

2) Boundary Shrink Model:

Proposition 1. *When a team of robots performs coverage of a spill collectively via tracking the boundary, each robot with an effective range d , an asymptotic boundary shrink control can be realized.*

Proof. Trivial. As shown in Fig. 3(b), when robot R_i travels along the boundary counterclockwise (CCW) and covers the spill on its left hand side, the unit area that is covered in time Δt is ΔC . When the robot first penetrates into a closed spill and moves from point $p(t_0)$ to $p(t_1)$, as illustrated in Fig. 3(b), the boundary length changes from $s_i(t) = \Delta\varphi \cdot r_i(t)$ to $s'_i(t) + 2d = \Delta\varphi \cdot (r_i(t) - d) + 2d$, assuming the radius of curvature r_i of the new position $p(t_i)$ is available. Even if the length of boundary increases aperiodically at some point, it shrinks to zero when the entire spill is removed, which leads to an asymptotic boundary shrink effect. Assuming N robots are involved in the boundary shrink control, the boundary will shrink asymptotically with a greater gradient while the robot team performs coverage. This ends the proof. $\square$

3) *Bounded Speed and Acceleration:* When a robot is tracking the boundary of a spill denoted as ∂S , and visits points $A(x_1, y_1)$ and $B(x_2, y_2)$ in sequence, we propose an approximated coverage manifold $\Delta C'$, which is shown in Fig. 3(b) and formulated as below in (3) and (4), to decide the maximal speed of robots. A similar model is used in [22].

$$\Delta C' = \{M(x, y) : (0 < \mathbf{AM} \cdot \mathbf{AB} < \mathbf{AB} \cdot \mathbf{AB}) \cap (0 < \mathbf{AM} \cdot \mathbf{AD} < \mathbf{AD} \cdot \mathbf{AD})\}, \quad (3)$$

$$D(x_4, y_4) = \left(-\sqrt{\|\mathbf{AD}\|^2 - y_4^2}, \frac{\|\mathbf{AB}\|^2 + \|\mathbf{AD}\|^2 - \|\mathbf{BD}\|^2}{2\|\mathbf{AD}\|}\right) \quad (4)$$

where M is a point in covered area $\Delta C'$, and $\|\mathbf{AD}\| = d$ is the effective range. Additionally, assuming a limit $\mathcal{V}$ for processing capacity in spill removal, which is similar to centrifugal volume and filtering system capacity in collecting contaminants, by having $\Delta C' = p_i \cdot d = v d \Delta t = \mathcal{V} \Delta t$, we determine the bounded speed for robots as below:

$$v_{max} = f(\mathcal{V}, d) = \frac{\mathcal{V}}{d}. \quad (5)$$

Here, we further set a maximum acceleration a_{max} which is bounded by actuator output power. When the robot moves, its linear acceleration will follow:

$$|a_i| \leq a_{max}. \quad (6)$$

The coverage model (3) and the effective range d can be validated with existing robots such as [1], which is also shown in Fig. 1. A left hand side coverage style leads to a uniform counterclockwise robot team moving direction which helps avoid inter-robot collision in a distributed way. The elaborated details can be found in Sec. V.

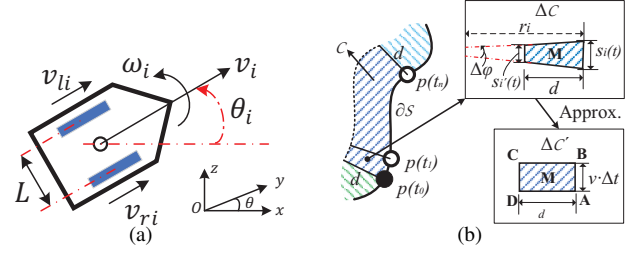


Fig. 3. (a) The mobility chassis of the robot driven by two parallel actuators. The robot has three degrees of freedom, i.e., $\dot{x}_i$, $\dot{y}_i$, and ω_i . (b) Coverage manifold for a robot maneuvering by tracking ∂S . A unit area covered is shown in a subfigure named ΔC , and its approximation is shown in another subfigure named $\Delta C'$ with unit time Δt and speed v .

IV. MULTI-ROBOT RENDEZVOUS

This paper handles the scenario where multiple separate spills exist in the workspace. With a limited vision sensing range, each spill can be recognized by robots using our previous multi-point rendezvous algorithm in [30], provided that every spill has at least one robot at its vicinity at first. As long as we deploy sufficiently many robots initially, this assumption is feasible.

At the beginning of the operation, every robot that can see spill boundary is set as a rendezvous point $D_m \in D$ ($m \leq \mathcal{M}$). Here, $\mathcal{M}$ is the number of rendezvous points, and D is the set of all rendezvous points. Then, multi-point rendezvous algorithm in [30] can be applied and every robot can rendezvous to its associated rendezvous point. Such rendezvous process is distributed and assures collision avoidance among robots.¹

When the rendezvous process is completed under the algorithm in [30], every robot will sit near its corresponding rendezvous point and therefore be able to detect the boundary. Once a robot meets its associated rendezvous point, the robot will switch to boundary searching state and move toward the nearest point on the boundary, as shown in Fig. 2 and 4.

Let $A(k) = (V(k), E(k), W(k))$ denote the undirected graph for wireless connectivity at time step k , where $V(k)$ represents the vertex set and $E(k)$ the edge set. The weight $w_{ij}(k)$ of each edge in the weight matrix $W(k) : [w_{ij}(k)]$ is approximately proportional to the Euclidean distance between two robots (i.e., $w_{ij}(k) \propto \|\mathbf{q}_i(k) - \mathbf{q}_j(k)\|$), which can be determined by wireless radio signal strength (RSS) measurement [4]. Note that the graph $A(k)$ is established based on local interaction of each robot by running *Algorithm 1* in [30]. Since *Algorithm 1* in [30] is distributed, every robot can build $A(k)$ in a distributed manner.

According to [30], each robot in the team $R_i \in \mathcal{R}$ will rendezvous to the leader robot $P(R_i) \in D$ with the least amount of time and energy, by following the optimization function below:

$$P(R_i) = \arg \min_{D_m \in D} d_{A(k)}(R_i, D_m), \quad (7)$$

¹We acknowledge that this paper uses the multi-robot rendezvous algorithm in our previous paper [30]. Thus, introducing a new multi-robot rendezvous algorithm is not within the scope of this paper.

where $d_{A(k)}(R_i, R_j)$ is the function denoting the shortest path between two nodes R_i and R_j in the graph $A(k)$ using Dijkstra's algorithm. Ultimately, $\mathcal{M}$ subgraphs ($A_m \subset A(k)$) where $m \in \{1, 2, \dots, \mathcal{M}\}$ are constructed for each rendezvous point, which is the node in the vicinity of the spills.

For each subgraph $A_m \subset A(k)$, a shortest-path-tree T_m is constructed with D_m as the rendezvous point. A hierarchical rendezvous algorithm, i.e., *Algorithm 2* in [30], is utilized to enable such rendezvous in each A_m such that all the nodes in this graph can be gathered around the associated spill. Such hierarchical rendezvous is validated through real experiments in [4].

In the hierarchical rendezvous algorithm (*Algorithm 2*) of [30], the child robot updates its parent from the current to the one in the upper level, when the child robot is very close to the current parent. This fact implies that the leaf robot (i.e., a node without any child) can switch to boundary searching and tracking state when it meets the rendezvous point D_m , which therefore closes the loop connecting multi-robot multi-point rendezvous and the proposed boundary shrink control.

In case that the spill is moving, the rendezvous points also move accordingly to avoid losing the vision contact to the boundary. Then, all the associate child robots will have to move following the rendezvous point to preserve wireless connectivity. This feature, namely herding in the literature [30], guarantees the success of the boundary shrink control under dynamic spill situation while avoiding creating an additional state to those shown in Fig. 2.

Compared with the work in [30], this work advances in extending the system distributedness. Instead of assuming a connected robot team in the workspace as described in [30], we validate cases where multi-point rendezvous succeeds in a graph with several disconnected components, as shown in Sec. VI-B2. A disconnected graph is more natural and practical for distributed robotic systems with limited wireless communication range.

V. BOUNDARY SHRINK CONTROL SCHEME

This section discusses how robots perform boundary shrink control by searching and tracking the boundary of a spill, and what controllers are designed in order to both avoid collision and achieve a complete coverage over the spill. Boundary searching and tracking happen simultaneously, thus they are discussed together here.

A. Boundary Searching Controller and Algorithms

When each robot is in place, it will then move along the spill boundary with the interior of the spill on the left hand side. In other words, all the robots move counterclockwise around the boundary, as demonstrated in Fig. 4, in the boundary tracking state. For a robot in rendezvous to begin tracking state and perform coverage operation, it has to switch to searching state first and align itself to the direction of the boundary. The boundary tracking control strategy is elaborated in Algorithm 2, while its initialization is detailed in Algorithm 1.

The goal position on the spill boundary can be determined via local vision sensors such as cameras, where the traveled

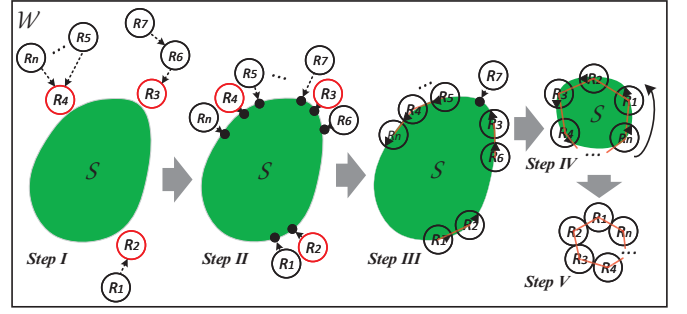


Fig. 4. A representative figure showing our proposed collective boundary shrink control strategy. The multi-robot hierarchical rendezvous is demonstrated in *Step I* where robots serving as rendezvous points D_m are highlighted with a red circle and the rendezvous hierarchy is indicated with dashed arrows. The robots deployed shall start boundary searching and navigate to the goals when they are in the vicinity of the spill and have no child robot associated, as shown in *Step II*. *Step III* demonstrates boundary tracking state on the completion of boundary searching, and robots maneuver along the boundary to shrink the spill. When robots are moving sequentially and getting closer, the proposed strategy prevents collision, as depicted in *Step IV*. The task is finished in *Step V*, where boundary shrinks to zero and no spill remains. The black dots denote the nearest deployment positions that robots are aiming for in the searching state. The triangles indicate robot moving direction, while the red lines mean the leading and trailing robots reside within APF effective range.

distance of robots is minimized. The shortest distance $\|d_i\|_{min}$ between $q_{s,i}$, the start position of robot R_i , $i \in 1, 2, \dots, N$, and $q_{g,i}$, the goal position which lays on the edge of spill S , can be estimated after the spill boundary is identified by image processing techniques [31]. Given the coordinates of goal positions for robots in the workspace, our study applies the artificial potential field (APF) method to enable navigation until robots reaches the goal positions. The navigation can be achieved with a robot onboard vision sensor that measures the current distance to the goal. An attractive potential field is introduced to achieve navigation and a repulsive potential field is introduced for collision avoidance. The robots working in the tracking state are assumed to have a higher avoidance priority than those maneuvering for rendezvous or searching. The avoidance priority is reflected in Fig. 2. If the workspace consists of multiple spills that need to be covered by the robots, an appropriate partition to the robot team has to be decided in completing multi-point rendezvous algorithm. The searching and tracking to the boundary remain the same for multiple spills cases as a single spill scenario.

Let $\mathbf{q}_{g,j} \in \mathbb{R}^2$ denote the goal position of a robot R_j . Let $d(\mathbf{A}, \mathbf{B})$ denote the distance between two positions $\mathbf{A} \in \mathbb{R}^2$ and $\mathbf{B} \in \mathbb{R}^2$. The attractive potential for a robot R_j moving toward the goal is formulated as

$$U_d(\mathbf{q}_j) = \frac{1}{2} \xi_1 d(\mathbf{q}_j, \mathbf{q}_{g,j})^2 \quad (8)$$

where ξ_1 is a scaling parameter. Meanwhile, a repulsive potential formulated as (9) is exerted on the robots working in searching state by the robots in tracking state, in order to avoid collision.

$$U_i(\mathbf{q}_j) = \begin{cases} \frac{\xi_2}{2} \left(\frac{1}{d(\mathbf{q}_j, \mathbf{q}_i)} - \frac{1}{d_0} \right)^2, & \text{if } d(\mathbf{q}_j, \mathbf{q}_i) \leq d_0, \\ 0, & \text{if } d(\mathbf{q}_j, \mathbf{q}_i) > d_0, \end{cases} \quad (9)$$

$i \in \mathcal{N}_t$

where ξ_2 is a positive scaling factor. $\mathcal{N}_t$ denote the robots working in tracking state. Here d_0 is the effective range, and we further assume that all the robots have equal $d_0 = r_A$.

Due to this, for any robot R_j , the final constructed potential field for searching navigation is:

$$U^*(\mathbf{q}_j) = U_d(\mathbf{q}_j) + \sum_{i \in \mathcal{N}_t} U_i(\mathbf{q}_j) \quad (10)$$

From the potential field function (8), and assuming no other robot within scope d_0 , we obtain the control input for the robots as below:

$$\mathbf{u} = k_u \tanh(\|\mathbf{q}_g - \mathbf{q}\|^2), \quad (11a)$$

$$\mathbf{w} = -k_w(\boldsymbol{\theta} - \boldsymbol{\varphi}) + \dot{\boldsymbol{\varphi}}. \quad (11b)$$

Here, $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_N]$, and $\boldsymbol{\varphi} = [\varphi_1, \varphi_2, \dots, \varphi_N]$. $\varphi_i \triangleq \arctan\left(\frac{U_y^*}{U_x^*}\right)$ is the direction of the collective potential field at a point (x, y) . Here, $U_x^* = \frac{\partial U^*}{\partial x}$ and $U_y^* = \frac{\partial U^*}{\partial y}$. In (11b), $k_u > 0$ and $k_w > 0$ are control gains, $\mathbf{u} = [u_1, u_2, \dots, u_N]$ and $\mathbf{w} = [w_1, w_2, \dots, w_N]$ are the input linear and angular velocity vectors of all the robots in searching state $\mathcal{N}_s$, respectively. $\dot{\varphi}_i$, the time derivative of φ_i , is shown as below considering the nonlinear model (1):

$$\dot{\varphi}_i = \frac{1}{U_x^{*2} + U_y^{*2}} \left(\left(\frac{\partial U_y^*}{\partial x} \cos \theta_i + \frac{\partial U_y^*}{\partial y} \sin \theta_i \right) U_x^* - \left(\frac{\partial U_x^*}{\partial x} \cos \theta_i + \frac{\partial U_x^*}{\partial y} \sin \theta_i \right) U_y^* \right) u_i$$

where u_i is provided in (11a). Here, the singularity happens when $U_x^{*2} + U_y^{*2} = 0 \Leftrightarrow U_x^* = 0 \wedge U_y^* = 0$. The singularity can be reached only if robot R_i is located at the local minimum of the potential field, and R_i will be stuck. However, we conclude in Proposition 2 that such local minimum does not exist if applying our proposed hybrid control scheme.

Noticeably, the input velocity $\|\mathbf{u}_j\|$ for the searching state does not need to be bounded by the maximum speed in (5), because the robot has not yet started coverage operation. But in analysis, the velocity is made to be bounded by v_{max} just to simplify the analysis process. Meanwhile, the robot acceleration is bounded mechanically according to (6) for searching state.

The linear control input (11a) is to motivate the robot R_i to approach the goal $q_{g,i}$, while the angular control (11b) is to align the orientation of the robot to the direction of the collective potential field U^* , which is φ_i . The control law (11) makes the robot with a unicycle model converge to the goal asymptotically from almost all initial conditions; its stability is proved in Proposition 1 of [32]. Considering that the robot may not arrive exactly at the designated goal position, or an oscillation may happen when the robot passes the goal position due to disturbance or inertia, we propose a dead zone value ϵ to terminate the searching state when $\|\mathbf{q}_{g,j} - \mathbf{q}_j\| < \epsilon$.

The reason why the repulsive potential component (9) does not include those robots in rendezvous state $\mathcal{N}_r$ or searching state $\mathcal{N}_s$ is because the trajectory of the robot in boundary searching will not intersect with robots in rendezvous or

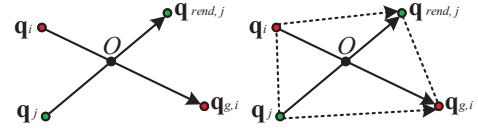


Fig. 5. A representative figure showing an intersection between trajectories of two robots R_i and R_j .

the same searching state. This statement can be proved by contradiction in Proposition 2.

Proposition 2. For a specific spill in the workspace, no intersection exists between the trajectories of any two robots if neither one is working under boundary tracking state.

Proof. This proposition is proved by contradiction. Without loss of generality, we assume that there is one intersection between the trajectories of robots $R_i \in \mathcal{N}_s$ and $R_j \in \mathcal{N}_r$, while R_i is starting from its current positions $\mathbf{q}_i$ and moving toward its goal position $\mathbf{q}_{g,i}$, and R_j is moving from its current position $\mathbf{q}_j$ toward its rendezvous point $\mathbf{q}_{rend,j}$, respectively. The intersection is denoted as O , shown in Fig. 5 (left). If connected with line segments, the four points form a quadrilateral. We can show that $\overline{O\mathbf{q}_i} + \overline{O\mathbf{q}_{g,i}} > \overline{\mathbf{q}_i\mathbf{q}_{rend,j}}$ and $\overline{O\mathbf{q}_j} + \overline{O\mathbf{q}_{rend,j}} > \overline{\mathbf{q}_j\mathbf{q}_{g,i}}$, thus we show $\overline{\mathbf{q}_i\mathbf{q}_{g,i}} + \overline{\mathbf{q}_j\mathbf{q}_{rend,j}} > \overline{\mathbf{q}_i\mathbf{q}_{rend,j}} + \overline{\mathbf{q}_j\mathbf{q}_{g,i}}$. If this conclusion stands, the better rendezvous point for R_j will be $\mathbf{q}_{g,i}$, which is on the boundary. Moreover, R_j will not remain in rendezvous state for long as it can detect the boundary (i.e., point $\mathbf{q}_{g,i}$). Meanwhile, since R_i does not rendezvous to point $\mathbf{q}_{rend,j}$, they do not reside in the same hierarchical rendezvous spanning tree which is described in [30], meaning that they do not have an intersected trajectories for being in different rendezvous spanning trees. Similarly, it can be proved that two robots working under searching state will not collide by replacing $\mathbf{q}_{rend,j}$ with $\mathbf{q}_{g,j}$. Those two robots can have less total displacement if their goal positions are swapped, which obviously violates Algorithm 2. Finally, no collision will happen between robots even if they are within the same rendezvous tree, according to [4]. Conclusively, there is no intersection of the trajectories of any two robots if none of them is in boundary tracking state. $\square$

Deadlock equilibrium is a common issue that hinders robot movement in a multi-robot situation. Although robots working in different states can avoid collision by eliminating a saddle point, as discussed above, they may still face a deadlock issue for unicycle models. Specifically, a robot with a unicycle model has to steer from the initial orientation while approaching to the goal position, whose trajectories can be seen in Fig. 10. This steering may exert conflicting forces upon adjacent robots towards opposite directions and hence causes a deadlock equilibrium. To eliminate such deadlock, we can increase the scope range d_0 in (9) to allow more room for steering, or stop robots before entering a computed collision zone and resume motion once the zone is clear [33]. Alternatively, strategies based on robot trajectory re-planning such as [34] or reciprocal velocity obstacle method [35] can also be utilized.

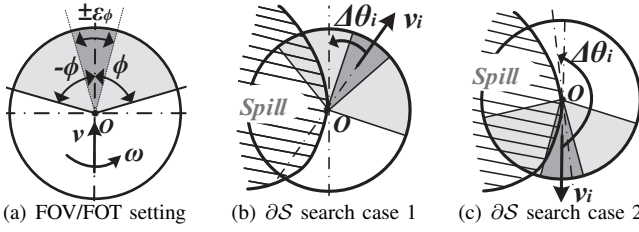


Fig. 6. (a) Definition of agent's field of view (FOV), which is a section spanning $[-\phi, \phi]$ and highlighted with light shadow, and field of tracking (FOT), a section spanning $[-\varepsilon_\phi, \varepsilon_\phi]$ being highlighted with dark shadow. (b) and (c) are two common spill boundary ∂S searching cases with angular changes produced by control laws and algorithms.

For an agent R_i with a field of view (FOV) of span $(-\phi, \phi)$, an angular velocity control law needs to be determined such that the agent is aligned with the direction of the boundary, especially when the robot is already on the boundary but needs re-orientation, such as the cases shown in Fig. 6(b) and Fig. 6(c). If the boundary does not initially reside in the FOV, this angular velocity control law is applied and the robot starts searching and aligning to the boundary. The searching state continues until the boundary is maintained within the field of tracking (FOT), a small view angle denoted $\pm\varepsilon_\phi$ and located within the FOV of robot, as shown in Fig. 6(a). The FOT is an axial dead zone designed for angular control that avoids frequent oscillation when robot is advancing. Algorithm 2 presents the control algorithm of boundary searching in detail. Before boundary searching starts, few system internal states that may affect Algorithm 2 are decided beforehand as initialization in Algorithm 1.

Persistent tracking of spill boundary ∂S is accomplished by Algorithm 3 and lasts until the coverage work is done. If a robot R_i worked in tracking boundary but suddenly misses the boundary from its vision possibly due to an overshoot or large disturbance, it needs to switch back to the searching state and detect the boundary again, as shown in Fig. 2. Note that for a drastically changing environment where the spill boundary may escape from the vicinity of the robot and cannot be detected by vision sensors any longer, the robot team needs to enter rendezvous state and move to the boundary again. Under certain circumstance, the robot can remain in tracking state despite of a dynamic spill. More details can be found in Sec. V-D.

A special case exists that the initial boundary may be shrunk by robots rooted at rendezvous D_i beyond it can be sensed by robots from D_j , ($j \neq i$); thus, rendezvous robot D_j loses its vicinity of the boundary. To circumvent this issue, we design two modes for a robot in tracking, namely *normal mode* and *idle mode*. The robot performs boundary shrink control only under normal mode, while tracking without shrinking the boundary in idle mode. One robot R_i switches to idle mode when it resides within an effective range of a static rendezvous robot D_j (with child robots associated), i.e., $\|\mathbf{q}_i - \mathbf{q}_{D_j}\| \leq \alpha \cdot r_c$, $\alpha \in (0, 1)$. Therefore, the boundary detected by D_j will remain until rendezvous completes. This fact is as well reflected in Line 7~10 of Algorithm 3.

Special consideration should be given in determining the ϕ_{cm} in Algorithm 2. On one hand, it should be as large as

Algorithm 1: Boundary ∂S Searching Initialization

```

1 Initialization /*All flags clear to 0 */
2 repeat
3   if  $R_i$  detects itself sitting on the spill boundary  $\partial S$  then
4     Set on_the_boundary_flag  $\leftarrow 1$ 
5     if boundary  $\partial S$  resides in FOT then
6       Set tracking_enable_flag  $\leftarrow 1$ 
7       break
8   if  $\partial S$  can be detected and is within the FOV then
9     Set boundary_within_FOV_flag  $\leftarrow 1$ 
10    break
11  if no boundary  $\partial S$  is detected then
12    Set no_boundary_detected_flag  $\leftarrow 1$ 
13    Set  $u_i = 0, w_i = 0$ 
14    end /*Terminate this program, as no spill is
        detected within vision range  $r_A$ */
15 until robot  $R_i$  rotates in the same place for  $2\pi$ 

```

Algorithm 2: Boundary ∂S Searching Algorithm

```

1 repeat
2   if on_the_boundary_flag = 0 then
3     if boundary_within_FOV_flag = 1 then
4       repeat
5         Identify the nearest point  $\mathbf{q}_{g,i}$  on  $\partial S$  to the
          robot;
6         Apply control law (11) to drive robot to
           $\mathbf{q}_{g,i}$ ;
7       until on_the_boundary_flag = 1;
8   else if on_the_boundary_flag = 1 then
9     The robot linear velocity control input  $u_i = 0$ ;
10    if boundary_within_FOV_flag = 1 then
11      if spill is on the right hand side of  $R_i$  then
12         $R_i$  rotates counterclockwise with respect to
          its centroid for a degree of
13         $\phi_{temp} = \omega_{max} \cdot T_c$  such that  $\partial S$  is outside
          the FOV;
14        Clear boundary_within_FOV_flag  $\leftarrow 0$ ;
15      else
16        repeat
17          Apply control laws (14) and (15) for
            direction alignment /* Since  $\partial S$  is
              within FOV and spill is on the left
              hand side of  $R_i$  */;
18        until  $\partial S$  is within the FOT;
19        Set tracking_enable_flag  $\leftarrow 1$ 
20      else if boundary_within_FOV_flag = 0 then
21        repeat
22           $R_i$  begins rotating counterclockwise with
            respect to its centroid for a degree of
23           $\phi_{temp}$ ;
24        until  $\partial S$  is within the FOV;
25        Set boundary_within_FOV_flag  $\leftarrow 1$ ;
26  until tracking_enable_flag = 1;
27 return Switch to the Tracking state /* End the program,
    and switch to the Tracking state in Algorithm 3 */;

```

possible in order to enlarge the step size, making rotation complete in fewer operation cycles. But on the other hand, it should not be so large that the robot misses the FOV in one rotation. Evidence for this is found in Algorithm 2, where the robot has to stop moving and indicate 'search fails' if failing

Algorithm 3: Boundary ∂S Tracking Algorithm

```

1 The robot  $R_i$  is located on the boundary of an associated
  spill
2 The velocity control input of  $R_i$  follows constraints (5) and
  (6)
3 repeat
4   if  $\partial S$  is within the FOV then
5     if  $\partial S$  is within the FOT then
6       Apply control law (22) and (20)
7       if  $\|q_i - q_{D_j}\| \leq \alpha \cdot r_c, D_j \in D$  then
8         Enable idle mode
9       else
10        Enable normal mode
11     else
12       Apply control laws (14) and (15)
13   else
14     return Switch to the Searching state /* End the
      program, tracking failed, switch to the Searching
      state in Algorithm 2*/
15 until forever

```

to detect the boundary in one full rotation. Therefore, we conclude that $\phi_{temp} = \omega_{max} \cdot T_c \leq 2\phi$. Thus, ω_i is bounded by:

$$\omega_{max} = \frac{2\phi}{T_c}. \quad (12)$$

If the boundary ∂S is in the FOV but not in the FOT, with the heading angle of agent being θ_i , the angle error with respect to the desired angle $\theta_d (= 0)$ is defined as $\theta_{ei} = \theta_d - \theta_i$. θ_d is the angle of spill boundary, which is detected using vision sensors of robots. [19] presented a method to transform an input vision image into a bird-eye view image. Then, using the bird-eye view image, we can estimate θ_d . How to estimate θ_d is beyond the scope of this paper. One can apply the following proportional-derivative (PD) control law to align the forward direction of the robot with the desired angle θ_d :

$$w_i = \begin{cases} -K_p \cdot \theta_{ei} - K_d \cdot \dot{\theta}_{ei}, & \text{if } \partial S \text{ appears in sector } [-\phi, -\varepsilon_\phi) \cup (\varepsilon_\phi, \phi], \\ 0, & \text{if } \partial S \text{ appears in sector } [-\varepsilon_\phi, \varepsilon_\phi], \end{cases} \quad (13)$$

where K_p and K_d are proportional and derivative gains, respectively.

Let T denote the current time when the control is applied. Considering the fact that the robot control is discrete in time, the following discrete control law is adopted instead of (13):

$$w_i = \begin{cases} -K_p \cdot \theta_{ei}(T) - \frac{K_d}{T_c} (\theta_{ei}(T) - \theta_{ei}(T - T_c)), & \text{if } \partial S \text{ appears in sector } [-\phi, -\varepsilon_\phi) \cup (\varepsilon_\phi, \phi], \\ 0, & \text{if } \partial S \text{ appears in sector } [-\varepsilon_\phi, \varepsilon_\phi]. \end{cases} \quad (14)$$

Finally, by considering the upper limit of angular speed shown in (12), we update the controller to be:

$$\|w_i\| = \begin{cases} \|K_p \cdot \theta_{ei}(T) + \frac{K_d}{T_c} (\theta_{ei}(T) - \theta_{ei}(T - T_c))\|, & \text{if } \|w_i\| \leq \omega_{max}, \\ \omega_{max}, & \text{otherwise.} \end{cases} \quad (15)$$

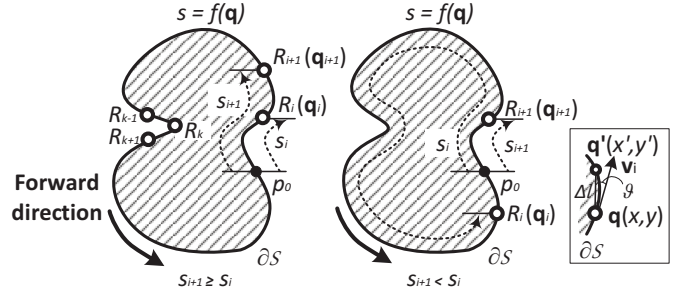


Fig. 7. The left figure shows two subgroups of adjacent robots, one subgroup consisting of R_{k-1} , R_k , and R_{k+1} , and the other consisting of R_i and R_{i+1} . Both the left and middle figures illustrate the definition of s_i and s_{i+1} for the trailing robot R_i and its leading robot R_{i+1} , respectively. The right figure in the box shows the symbols and their geometrical relationship when a robot is maneuvering along the boundary ∂S .

Specifically, if the boundary ∂S is outside the FOV, the proposed control laws are able to rotate the robot until ∂S falls into the FOV. However, if the ∂S resides in FOV but the spill is on the right hand side of the robot, that means the robot is heading the wrong direction. Hence, the robot has to rotate counterclockwise until it catches sight of ∂S again, as illustrated in Fig. 6(c). This procedure is presented at Line 10~13 of Algorithm 2. Note the rotation process outside FOV is open-loop based and aims for no specific goal. The proposed ω_{max} guarantees the minimum possible steps to finish searching. When the ∂S is once again within the FOV, control laws (14) and (15) can be applied. Both (14) and (15) are based on a vision sensor feedback available at low cost in terms of implementation. If more accurate and consistent control is needed for more complex environments, one can refer to the visual servoing methods proposed in [36], [37].

B. Boundary Tracking Controller and Convergence Proof

When a group of robots is performing boundary shrinking for spill coverage along the boundary, a proper control method needs to be proposed to avoid collision between robots but still motivate them to move forward. Such a controller based on the APF method is provided and discussed in this section.

The robots are re-labeled after the completion of allocation. For a set of sequential indices, they are assumed to be associated with a specific spill. Two indices of robots are sequential means they are adjacent, such as R_{i+1} and R_i . Such re-labeling is considered as well in the analysis of Sec. V-C.

Since workspace $\mathcal{W}$ is assumed to be obstacle-free, attractive potential U_{att} exerted on R_i by its leading robot R_{i+1} is sufficient to drive R_i . When R_i is tracking the boundary ∂S , it can measure the arc length in boundary rather than line distance between itself and other agents within its FOV. It uses arc length in boundary because a closer line distance between two robots does not mean that they are adjacent if tracking a serpentine boundary. For instance, in Fig. 7, the adjacent robot of R_{k-1} is R_k rather than R_{k+1} , although R_{k+1} is closer to R_{k-1} in a line distance. Practically, such measurement can be performed with LIDAR, high resolution laser range-finders, or vision sensors equipped with imaging techniques proposed in [38], [39]. Suppose that a function $s = f(\mathbf{q})$ can represent the spill boundary ∂S , where $s \in [0, \|\partial S\|]$ demonstrates the arc

length between a reference point and an agent located at $\mathbf{q}$ in the counterclockwise direction. Here, $\|\partial S\|$ indicates the total length of ∂S . Note that the definition of s and a starting point p_0 are obtained from a global view and used only for stability and convergence analyses. We can show a pair of adjacent robots R_i and R_{i+1} in Fig. 7, where R_{i+1} is the leading robot of R_i . Once R_{i+1} falls in the FOV of its following robot R_i , arc length distance between these two adjacent robots, namely $l_i = \|\mathbf{q}_i, \mathbf{q}_{i+1}\|_{\partial S}$, can be computed by

$$l_i = \begin{cases} s_{i+1} - s_i, & \text{if } s_{i+1} \geq s_i, \\ s_{i+1} - s_i + \|\partial S\|, & \text{if } s_{i+1} < s_i. \end{cases} \quad (16)$$

s_i is defined as the s in the view of R_i , and p_0 in Fig. 7 is the reference point for s . Recall that r_A is the vision sensor range and is illustrated in Fig. 1. $s_{i+1} \leq s_i$ can happen depending on where the reference point is, which is shown in Fig. 7. If the leading robot of R_i is beyond the radius r_A of its FOV, (16) does not apply. Therefore it defines a virtual distance l_i^* which includes both situations as below

$$l_i^* = \begin{cases} l_i, & \text{if } l_i \leq r_A, \\ r_A, & \text{if } l_i > r_A \text{ or } l_i \text{ is unknown.} \end{cases} \quad (17)$$

The attractive potential function is defined as

$$U_{att} = \frac{1}{2} \xi_3 l_i^{*2} \quad (18)$$

where ξ_3 is a positive scaling factor. The velocity $\mathbf{v}_i$ should be in the direction of negative gradient of U_{att} with respect to s_i such that

$$\mathbf{v}_i = -\nabla_{s_i} U_{att} = \xi_3 l_i^*. \quad (19)$$

However, extra factors need to be considered when applying (19) as the control input.

1) *Consider the maximum linear speed*: It is worth noticing that the robot cannot exceed the maximum linear speed v_{max} . Thus, we should have

$$\mathbf{v}_i = \begin{cases} \xi_3 l_i^*, & \text{if } \|\mathbf{v}_i\| \leq v_{max}, \\ \frac{\xi_3 l_i^* v_{max}}{\|\xi_3 l_i^*\|}, & \text{otherwise.} \end{cases} \quad (20)$$

2) *Control design*: Controllers similar to (20) are developed in the works such as [27]; however, they do not consider or fully discuss the control design under unicycle model (1). Here, we provide a strategy inspired by [40] to determine the velocity control input tailored for unicycle model. Note that the robot acceleration is bounded mechanically according to (6) for boundary searching and tracking states.

Let (x, y) present the current position of R_i . Assuming a point (x', y') lying on ∂S and in front of the robot R_i with a small length offset of Δl , we have its global coordinate as $x' = x + \Delta l \cos(\vartheta + \theta_i)$ and $y' = y + \Delta l \sin(\vartheta + \theta_i)$, having ϑ as the angle between orientations of the robot and the potential field $U_{att}(x, y)$. Recall that θ_i is the orientation direction of the robot R_i . The aforementioned symbols and their geometrical relationship can be found in Fig. 7 (in the box), where the orientation of the potential field is represented as $\mathbf{v}_i$. Practically, (x', y') can be sensed by local vision sensors

of R_i . Let $\mathbf{Q} = [u_i, w_i]^T$ be the velocity control input and $\mathbf{v}_i = [v_x^i, v_y^i]^T$ in (20), the velocity becomes as follow:

$$\begin{bmatrix} \dot{x}' \\ \dot{y}' \end{bmatrix} = \begin{bmatrix} v_x^i \\ v_y^i \end{bmatrix} = \begin{bmatrix} \cos \vartheta & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \Delta l \end{bmatrix} \begin{bmatrix} u_i \\ w_i \end{bmatrix} = \mathbf{G}(\vartheta, \Delta l) \mathbf{Q}. \quad (21)$$

Then the control input can be obtained by $\mathbf{Q} = \mathbf{G}^{-1} \mathbf{v}_i$, i.e.,

$$u_i = v_x^i \cos \vartheta + v_y^i \sin \vartheta, \quad (22a)$$

$$w_i = \frac{1}{\Delta l} (-v_x^i \sin \vartheta + v_y^i \cos \vartheta). \quad (22b)$$

Note that the controller (22) is bounded using (20), where the the velocity set $\mathbf{v}_i = [v_x^i, v_y^i]^T$ in (22) is derived from (20). In the case of coarse measurement or irregular spill shapes such that (16) cannot be obtained with precision, cooperative Kalman filters presented in [17] can be utilized to estimate the boundary curvature and determine the control law, when a noisy scalar field is built for the spill.

3) *Speed convergence*: When $l_i^* \rightarrow 0$, we need to show the convergence of linear speed. By differentiating $U_{att}(\mathbf{q}_i)$ in (18) with respect to time t and considering (20), we have

$$\dot{U}_{att} = -\xi_3 \dot{l}_i^* = -\xi_3^2 l_i^* = -2\xi_3 U_{att}. \quad (23)$$

From (23), it shows

$$U_{att} = U_{att}(0) e^{-2\xi_3^2 t}. \quad (24)$$

Obviously, the maximum potential $U_{att} = \frac{1}{2} \xi_3 r_A^2$ exists if the distance of boundary between robot R_i and its leader R_{i+1} are far enough. Therefore, it is preferable if R_{i+1} travels beyond the FOV of R_i . Moreover, the greater ξ_3 is, the faster U_{att} and l_i^* converge to 0. One last significant remark is, from control laws (20) and (15), we can see that one trailing robot R_i cannot wrap up its leading robot R_{i+1} , because along the boundary, the attractive potential decreases to zero as R_i is approaching R_{i+1} , thus linear velocity input also decreases to zero.

4) *Robot convergence after boundary shrink*: When coverage is coming to an end and boundary shrinks to almost a point, all the robots associated with a specific spill should then form a loop and converge around a place, as long as the spill remains convex during coverage. One might ask whether the robots will reduce their speed while moving to the endpoint and eventually stop when they have arrived. A positive conclusion is given in the discussion in Sec. V-B3, which shows that the linear speed $\|u_i\|$ for every agent converges to zero when l_i^* approaches zero. The whole process stops when the robot cannot find any more spill, or, more specifically, when robot FOV still misses boundary ∂S after a full rotation under searching state. This implies that the spill have been fully cleaned. Practically, a complete coverage can be achieved when the conveyor belts of robots, as shown in Fig. 1, are crowding toward each other. The dimensions of the robot may affect the convergence result, since the robots' dimensions prevent them from getting closer to each other. Nevertheless, this issue can be tackled by improving robot design, for instance, a retractable or flexible conveyor belt can be adopted for robots in Fig. 1. The stability of multi-robot coordination control in coverage is detailed in Sec. V-C. The information about nonconvex spills is provided in Sec. VII.

C. Stability Analysis

We investigate the stability of robot coordination in coverage control. At a certain time $t = T_+$ after long operation, we assume all robots deployed are in the tracking state, and each of them has $l_i \leq r_A$, meaning that it has a leading robot within its view and resides within the view of a trailing robot. Defining the n_j number of robots allocated for spill $\mathcal{S}_j$ as $R_1^j, R_2^j, \dots, R_{n_j}^j$, the boundaries for spills $[\|\partial\mathcal{S}_1\|, \|\partial\mathcal{S}_2\|, \dots, \|\partial\mathcal{S}_M\|]_{t=T_+}$, and $\mathbf{l}^* = [l_1^*, l_2^*, \dots, l_N^*]_{t=T_+}^T$ at the current time, the system collective potential energy is provided by

$$V(\mathbf{s}) = V(\mathbf{l}^*) = \sum_{j=1}^M \left(\sum_{i=1}^{n_j} U_{att}(i) \right). \quad (25)$$

Here, $\mathbf{s} = [s_1, s_2, \dots, s_N]^T$, and $U_{att}(i)$ is the attractive potential shown in (18).

We define a sparse matrix $Z = [z_{ij}]_{M \times N}$ to represent the association of each robot to a specific spill, where $z_{ij} = 1$ means robot R_i is associated to spill $\mathcal{S}_j$. The matrix Z also satisfies constraint (26).

$$\sum_{i=1}^N z_{ij} \geq 1, \quad \sum_{j=1}^M z_{ij} = 1, \quad \forall j \in \{1, 2, \dots, M\}, \quad (26)$$

Obviously, the collective potential (25) has to satisfy a boundary constraint as shown below

$$Z \cdot \mathbf{l}^* = [\|\partial\mathcal{S}_1\|, \|\partial\mathcal{S}_2\|, \dots, \|\partial\mathcal{S}_M\|]_{t=T_+}^T. \quad (27)$$

Using (25) and the result of Sec. V-B3, we have the time derivative of $V(\mathbf{s})$ as

$$\begin{aligned} \dot{V}(\mathbf{s}) &= \frac{dV_1}{ds_1} \cdot \frac{ds_1}{dt} + \frac{dV_2}{ds_2} \cdot \frac{ds_2}{dt} + \dots + \frac{dV_N}{ds_N} \cdot \frac{ds_N}{dt} \\ &= -\xi_3 \mathbf{l}^* \cdot \mathbf{v} \leq 0 \end{aligned}$$

where $\mathbf{v} = [v_1, v_2, \dots, v_N]^T$. Thus, we show that the Lyapunov function $V(\mathbf{s})$ is negative semi-definite, and $\dot{V} = 0$ iff $\mathbf{l}^* = 0$, i.e., $s_i = s_{i+1}$ for any two adjacent robots tracking the same spill and $\|\partial\mathcal{S}_i\| = 0$, according to (16), (17), and (27). Meanwhile, the velocity v_i is bounded by (20) and it decays with l_i^* . Therefore, according to LaSalle's invariance principle [41], those robots deployed to a specific spill will asymptotically converge to a place when the spill boundary shrinks to a point, i.e., the spill is removed.

D. Dynamic Spill Applicability

In real cases, the boundary of a spill cannot be always static during the operation. In this section, we prove that our proposed boundary shrink control strategy can also handle a case when the spill boundary is moving.

The multi-robot rendezvous algorithm deals with moving boundary intrinsically by motivating the rendezvous robot D_m to move according to the boundary. Therefore, the child robots will eventually gather in the vicinity of the changing boundary. Such behavior that has a dynamic rendezvous point is named as robotic *herding* and validated in [30].

In boundary searching state, our solution can handle moving boundary case by simply remaining in this state and rerunning

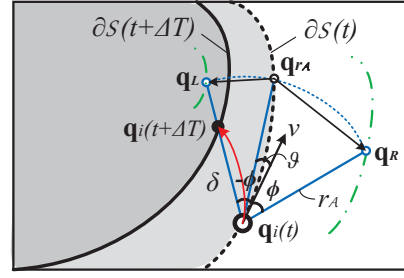


Fig. 8. The trajectory (red line) of robot R_i working in the boundary tracking state when the boundary is changing. The dashed black curve is the boundary at time t while the solid black curve is the boundary in $t + \Delta T$. The blue lines with a span $(-\phi, \phi)$ indicates the field of view (FOV) of robot. The dash-dotted lines in green represent the critical location of boundary escaping the robot vision range from left and right side, respectively.

Algorithm 2, provided that the boundary does not move faster than robot maximal speed v_{max} in the case that the boundary escapes from robot vision range r_A when the robot is chasing the goal position on the boundary.

If boundary moving happens in the boundary tracking state, then our Algorithm 3 can tackle this issue, even without switching to boundary searching state if the changing boundary appears inside the FOV, as shown in Fig. 8. Since the changing boundary appears inside robot FOV, the trajectory of robot working in the boundary tracking state and a dynamic boundary is shown in the red line in Fig. 8. In the same figure, robot's current position is $\mathbf{q}_i(t)$, and the goal position on the dynamic boundary for the robot to track denotes $\mathbf{q}_i(t + \Delta T)$, whose coordinate can be calculated as

$$\mathbf{q}_i(t + \Delta T) = \begin{bmatrix} x(t + \Delta T) \\ y(t + \Delta T) \end{bmatrix} = \begin{bmatrix} \delta \cos(\phi + \theta_i) + x(t) \\ \delta \sin(\phi + \theta_i) + y(t) \end{bmatrix}, \quad (28)$$

where $[x(t), y(t)]^T$ is the coordinate of $\mathbf{q}_i(t)$, and δ is the distance between $\mathbf{q}_i(t)$ and $\mathbf{q}_i(t + \Delta T)$ which can be determined by image processing techniques such as [42]. Using control law (11) by replacing $\mathbf{q}_g$ in it with $\mathbf{q}_i(t + \Delta T)$, the robot can track the dynamic boundary.

For a robot working under boundary tracking state, as shown in Fig. 8, the spill boundary can escape the robot vision range from either left or right side if $\mathbf{q}_{rA}$, the furthest point on the boundary that is within robot vision range, moves over point $\mathbf{q}_L$ or $\mathbf{q}_R$. Given this fact, we can decide the maximum speed for the moving spill, $v_{max}^{\partial\mathcal{S}}$, such that the boundary always resides in robot's FOV. $v_{max}^{\partial\mathcal{S}}$ can be determined as $v_{max}^{\partial\mathcal{S}} = \min\left\{\frac{\|\mathbf{q}_L - \mathbf{q}_{rA}\|}{T_c}, \frac{\|\mathbf{q}_R - \mathbf{q}_{rA}\|}{T_c}\right\} = \min\left\{\frac{(\phi - \vartheta)r_A}{T_c}, \frac{(\phi + \vartheta)r_A}{T_c}\right\}$. Here, recall that T_c is the robot control system cycle time.

VI. SIMULATION RESULTS

To evaluate the proposed boundary shrink control strategy, extensive experiments using simulation were conducted in Robotarium MATLAB platform [43] which is an open source swarm robotics experiment testbed. Robotarium also supports hardware experiments with GRITsbot robots. Robotarium has a 2D arena of size $3\text{m} \times 3\text{m}$, which can be used to simulate a scale-down aquatic workspace such as lakes and oceans where spills are patched. In the experiments, four spills with different geometric features are presented, including two circles with

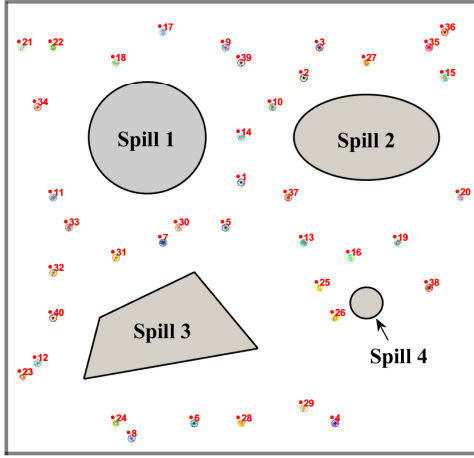


Fig. 9. The workspace for experiments with 4 spills featuring different shapes and areas. Randomly distributed robots are also indexed.

TABLE I
GEOMETRY OF THE SPILLS IN THE WORKSPACE $\mathcal{W}$.

Geometry	Spill1	Spill2	Spill3	Spill4
Area(m ²)	0.4301	0.4046	0.4200	0.0314
boundary(m)	2.3247	2.3926	2.8415	0.6283

different areas, one ellipse and one quadrilateral, as shown in Fig. 9. The diverse geometry of presented spills will help to validate the capability of the proposed method. In addition, Table I shows the index and area of the spill patches, and v_{max} in (5) is made equal to 0.01 m/s to cope with this testbed.

A. Evaluation Metrics

The following three metrics were used to evaluate the performance of the proposed coverage control strategy in our experiments. Note that the units in simulations are in accord with Robotarium testbed, which is a scaled-down environment.

- 1) Lyapunov candidate function (Convergence):

$$L_x = \sum_{R_i, R_j \in S_x, i \neq j} \|\mathbf{q}_i - \mathbf{q}_j\|. \quad (29)$$

If there is only one robot allocated for a spill S_x , the Lyapunov candidate function is revised as:

$$L_x = \sum_{R_i \in S_x} \|\mathbf{q}_i - \mathbf{c}_x\|, \quad (30)$$

where $\mathbf{c}_x$ is the computational centroid of the patch.

- 2) Total distance traveled by all robots after covering each spill (Distance):

$$D_{sum} = \sum_{S_x \in \mathcal{W}} \sum_{k=1}^{k_{max}} \|\mathbf{q}_i(k+1) - \mathbf{q}_i(k)\|. \quad (31)$$

- 3) Number of iterations ($k_{stop} \leq k_{max}$) to reach the following stop condition:

Stop at k_{stop} if the current area which is bounded by instant boundary satisfies

$$A_x \leq A_{min}, \forall S_x \in \mathcal{W}. \quad (32)$$

Here A_{min} is defined to be 1% of the original area of each spill.

B. Experiment Cases and Results

We validate our proposed solution in terms of efficacy, efficiency, scalability, and convergence. First, we select a case where a relatively large vision sensing range $r_A = 1\text{m}$ is adopted such that every robot can detect the boundary within its vision range. Therefore, every robot itself becomes a rendezvous point $D_i, i \leq N$ and is designated to a specific spill by following a nearest-neighbor method. This case is to show that a larger vision range helps reduce rendezvous hierarchy and hence the total operation time. The robots association to a spill is shown in Table II. Meanwhile, as a contrast, in the second case, a smaller vision sensing range and a certain wireless communication range are chosen. In the second case, we use the same number of robots as the first case. We verify the performance of the proposed boundary shrink control method where a hierarchical rendezvous takes effect before boundary shrink control starts.

We validate our proposed strategy with many other experiments in terms of different operation ranges, different numbers of robots, etc. To be concise, the detailed results such as robot initial distributions and trajectories for each spill are omitted. Instead, the effects resulted by these factors are discussed in a comprehensive way in Sec. VI-C. All the experiments can be seen in complementary videos.

1) *Case 1* ($N = 40, r_A = 1\text{m}$): In this case, the boundary shrink control with 40 robots deployed was tested. Fig. 10(a) shows the random initial distribution of the 40 robots. Since the large vision sensing range r_A allows each robot being the rendezvous point, they have no child robot associated and can directly switch to boundary searching and tracking states. No hierarchical rendezvous or inter-robot communication is needed. Assume each spill in the workspace is detected by at least one robot, all the spills can be covered eventually. After removing the spill, the robots ultimately should converge at a point near the centroid of the spill as the boundary shrinks to zero. As documented in the top field of Table. II, every spill was associated with many robots, except for Spill 4. The area evolution chart in Fig. 10(c) shows that the completion progress of coverage was steadily declining. Since Spill 4 was detected with only one robot, its Lyapunov candidate function was defined according to (30) while other spills have Lyapunov candidate function defined according to (29). The convergence figure in Fig. 10(d) also shows the sum of the Lyapunov candidate values of all the spills, i.e., $\sum_{S_x \in \mathcal{W}} L_x$, and therefore describes the overall performance of robot convergence in the workspace. According to the convergence figure, all the robots converged to their endpoints while covering, which was confirmed by the Lyapunov candidate function L_x approaching to zero. In addition, because of the physical dimensional restraint of robots, the sum of convergence in this case did not amount to zero when the task approached its end. As depicted in Fig. 11(a), the boundary, evaluated as perimeter, was shrinking asymptotically while spill coverage is in progress. Fig. 10(b) shows the trajectories of the 40 robots during boundary shrink control. From Fig. 10(b), we can see that twisted trajectories developed in R_{13} and R_{32} during the searching period. This happened because R_{13} and

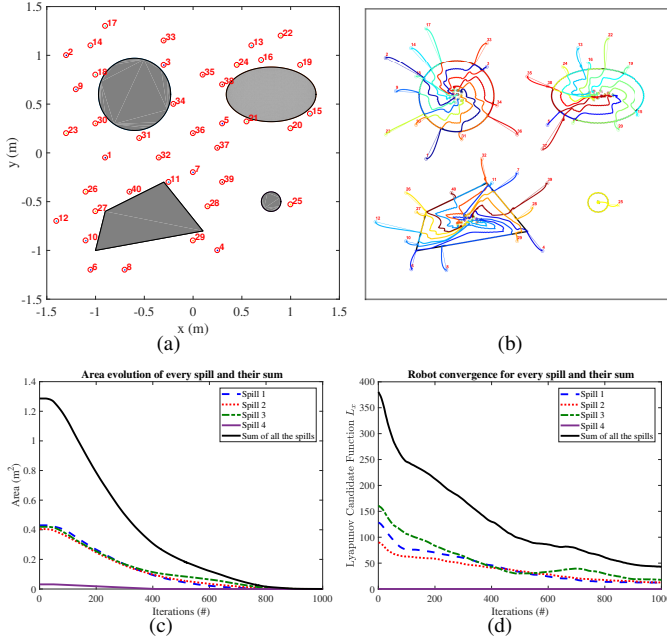


Fig. 10. (a) shows the initial random distribution of robots while (b) illustrates the coverage trajectories for Case 1 with 40 working robots. The area evolution and convergence with respect to time (iterations) are shown in (c) and (d), respectively.

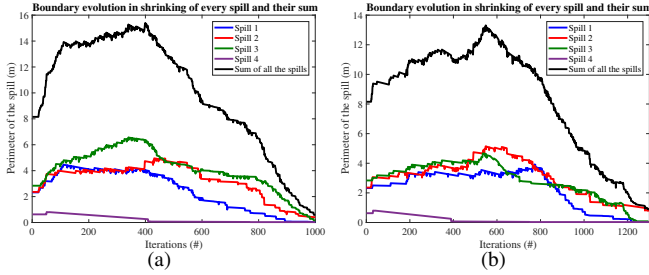


Fig. 11. The perimeter evolution is depicted separately of Case 1 (a) and Case 2 (b).

R_{32} were trying to avoid colliding with approaching robots R_{16} and R_{11} , which were working in the tracking state and hence prioritized over the searching state in collision. This validated the effectiveness of our proposed boundary shrink control strategy in terms of collision avoidance.

2) *Case 2* ($N = 40, r_A = 0.13m, r_C = 0.3m$): With a small vision range but relatively wide wireless communication range, the robots that are far from the spill can maneuver to its vicinity and catch sight of the boundary with the help of hierarchical rendezvous based on wireless connectivity. The initial robot distribution is set to be identical to Case 1, which can be found in Fig. 10(a). In this case, 18 shortest path trees are constructed according to Sec. IV with rendezvous points being $D_1(R_3), D_2(R_{18}), \dots$, and $D_{18}(R_{25})$, representing the robots that already detect the boundary at first. The rendezvous path trees are shown in Fig. 12. The numerical results including robot association to these spills, rendezvous points D_m , and experiment data are shown at the second row of Table II.

The convergence curve was smoother but less sharp at the beginning in this scenario than in Case 1, as shown in Fig. 13(d), since multi-point rendezvous was taking effect but

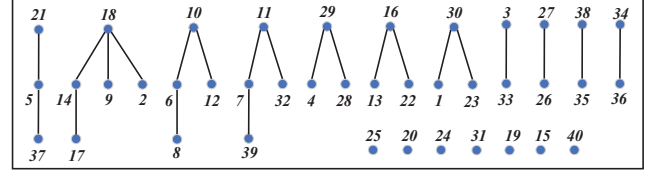


Fig. 12. The shortest path trees that depict the hierarchical rendezvous, where the root node for each tree is a rendezvous point.

did not yield a faster convergence in general. Meanwhile, since part of the operation time was spent on rendezvous, which ended up until iteration $t = 298$ in the experiment, the total coverage operation took longer time than that of Case 1. The boundary shrink control started after each robot moved and saw the boundary within its vision range, which happened right after robot completed rendezvousing to its corresponding rendezvous point. The spill areas were decreasing monotonically as shown in Fig. 13(c) during the boundary shrink control, with operation range being $d = 0.090m$. The robot trajectories are shown in Fig. 13(b), the child robot such as R_{28} switched to boundary searching state at the time it met the spill boundary while approaching its rendezvous points R_{29} .

The rendezvous hierarchy at the beginning of rendezvous is illustrated in Fig. 12, which can be updated with local sensing. In Fig. 12, robots such as R_{37} , R_{17} , etc., are leaf robots, meaning that they have no child robot associated. Noticeably, robots can switch to boundary searching state whenever they see the boundary within vision range, even if it happens before completion of rendezvous. This fact was verified by agents such as R_{23} , which first saw the spill boundary when it was moving toward R_{30} and switched to boundary searching state as it had no child robot. The parent robots except for rendezvous points (e.g. R_6 and R_{14}) can move while keeping the connectivity with their child robots (i.e., R_{17} and R_8). Robots that become rendezvous points start moving to the boundary and perform boundary shrink control unless they have no child robot, such as R_{25} and R_{40} , or all their child robots have switched away from rendezvous state.

C. Further Discussion

Since the completion time of boundary shrink control can be reduced if having more working robots, we test a total of four scenarios in which robot numbers are increasing with the same interval, i.e., $N = \{30, 40, 50, \text{ and } 60\}$, provided that r_C and r_A are the same as Case 2. Our analysis of these four scenarios reveals some significant outcomes. First, the total operation time decreased significantly by having more robots involved. More robots may lead to more rendezvous points and a fewer number of hierarchy levels, which accelerates rendezvous process. This result further substantiates the significance of our proposed multi-point rendezvous strategy. The area evolutions of the four spills as a whole for each scenario are depicted in Fig. 14(a). Second, the operation time decreased much slower as we employed more than 50 robots, mostly because inter-robot collision avoidance has become an dominating issue that resulted in a drop in efficiency.

Furthermore, we validate our solution for different operation ranges: $d = \{0.045m, 0.09m, 0.135m, \text{ and } 0.18m\}$, with the

TABLE II

PERFORMANCE OF BOUNDARY SHRINK CONTROL IN CASE 1 AND 2 HAVING $N = 40$ ROBOTS, WITH CORRESPONDING MAXIMUM ITERATIONS k_{max} .

Case	Metrics	Spill1	Spill2	Spill3	Spill4
Case 1 $r_A = 1\text{m}$ $k_{max} = 1,000$	Residual area at $k_{max}(\text{m}^2)$	7.85×10^{-5}	7.97×10^{-5}	7.19×10^{-4}	7.85×10^{-5}
	Completeness at $k_{max}(\%)$	99.96	99.98	99.83	99.75
	Number of allocated robots	13	12	14	1
	k_{stop} at $A_{min}(\#)$	750	842	841	410
	$D_{sum}(m)$	2319.1	2838.5	2712.8	23.6
Case 2 $r_A = 0.13\text{m}, r_C = 0.3\text{m}$ $k_{max} = 1,300$	Residual area at $k_{max}(\text{m}^2)$	2.85×10^{-4}	7.94×10^{-5}	7.85×10^{-5}	7.85×10^{-5}
	Completeness at $k_{max}(\%)$	99.93	99.98	99.98	99.75
	Number of allocated robots	13	12	14	1
	Rendezvous point D_m (indices)	5 (#3, #18, #30, #31, #34)	7 (#15, #16, #19, #20, #21, #24, #38)	5 (#10, #11, #27, #29, #40)	1 (#25)
	k_{stop} at $A_{min}(\#)$	924	1093	1115	390
	$D_{sum}(m)$	2420	2283.2	2494.9	24.7

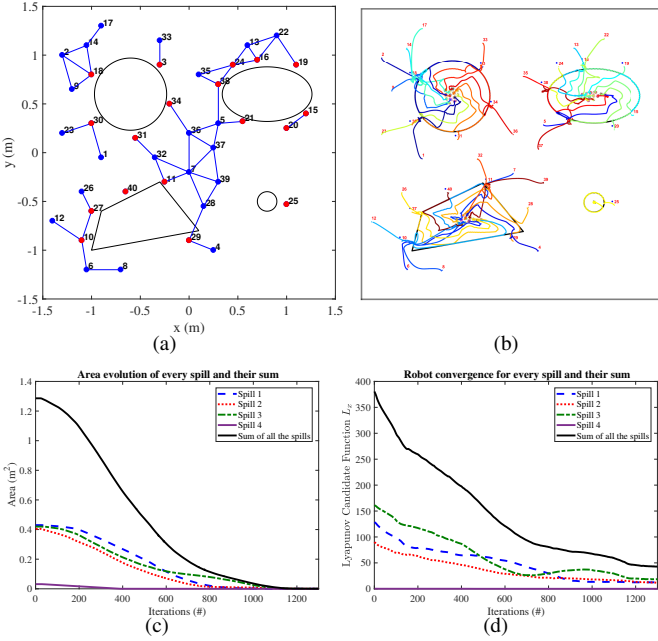


Fig. 13. (a) The same initial distribution of robots as Case 1, with wireless connectivity represented as a graph in solid blue lines and rendezvous points D_m highlighted in red. (b) shows boundary shrink control trajectories with multi-point rendezvous. The area evolution and convergence of Case 2 are shown in (c) and (d), respectively.

same initial robot distribution as Case 2 and appropriate range values. The area evolution of the entire workspace for these operation ranges is shown in Fig. 14(b). The videos of these experiments can be found in the paper video available at <https://youtu.be/XUrKICak4SQ>.

Through these experiments, we validated the efficacy, efficiency, and scalability of our proposed solution.

VII. NON-CONVEX SPILL

The covered spill mentioned above which boundary shrink control performs on is assumed to be convex during operation. However, as can be seen from the experiment videos, Spill 3 failed to remain convex while robots were collecting the spill. Fortunately, the task was still completed because the spill was not completely broken. It is possible to overlook this situation and apply our solution to non-convex situations. However, the challenge is how to make sure every separate piece of spill has at least one robot associated with it when an original spill is

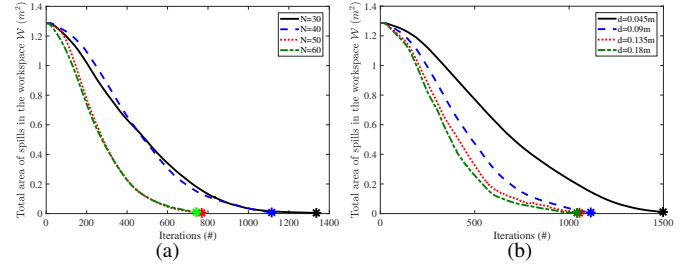


Fig. 14. (a) The change of residual areas in the workspace with respect to the iteration number for having different number of working robots N . The star marker at the end of every curve indicates its corresponding $\max\{k_{stop}\}$. (b) The evolution of residual areas for having different operation range d . The star markers indicated $\max\{k_{stop}\}$ for each scenario.

broken into several pieces, since the narrowest place is eroded with coverage. Robot rendezvous can be executed to reallocate the robots or create allocation methods that promote robot dispersion and prevent many robots from gathering at one piece [44]. Enabling ad hoc networking among robots may also help to achieve convergence while the robots are assembling for coverage. Research on this topic is to be further explained in our future works.

VIII. CONCLUSIONS AND FUTURE WORK

This paper proposes a concrete boundary shrink control strategy that can be used in scenarios requiring collective coverage such as oil spill cleanup, complete exploration of an environment, and cooperative de-mining. The system does not rely on any method of global localization; its distributed mechanism promises scalability and robustness to varied scenarios. The boundary shrink control is implemented as a hybrid solution consisting of three states: *rendezvous*, *searching*, and *tracking*. The convergence and stability of this hybrid control are proved mathematically. Extensive experiments in simulation have verified the effectiveness of our solution in concurrently covering multiple spills under many practical constraints, and the adoption of different parameter configurations demonstrated the system's scalability and flexibility.

Future work will investigate how the proposed boundary shrink control method can be adapted to non-convex spills. We will also examine how to decide the optimal numbers of robots for given spill shapes and distributions. In addition, more features such as fault tolerance will be introduced with

the primary goal of improving the system's robustness and efficiency in complex environments.

REFERENCES

- [1] MIT Senseable City Lab, <http://senseable.mit.edu/seaswarm/>, accessed: July 31, 2018.
- [2] J. Clark and R. Fierro, "Cooperative hybrid control of robotic sensors for perimeter detection and tracking," in *Proceedings of the 2005, American Control Conference, 2005*. IEEE, 2005, pp. 3500–3505.
- [3] A. Dhariwal, G. S. Sukhatme, and A. A. Requicha, "Bacterium-inspired robots for environmental monitoring," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 2. IEEE, 2004, pp. 1436–1443.
- [4] S. Luo, J. Kim, R. Parasuraman, J. H. Bae, E. T. Matson, and B.-C. Min, "Multi-robot rendezvous based on bearing-aided hierarchical tracking of network topology," *Ad Hoc Networks*, vol. 86, pp. 131–143, 2019.
- [5] F. D. V. B. Luna, E. de la Rosa Aguilar, J. S. Naranjo, and J. G. Jagüey, "Robotic system for automation of water quality monitoring and feeding in aquaculture shadehouse," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1575–1589, 2017.
- [6] Z. Liu, P. Smith, T. Park, A. A. Trindade, and Q. Hui, "Automated contaminant source localization in spatio-temporal fields: A response surface and experimental design approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 3, pp. 569–583, 2016.
- [7] X. Jin and A. Ray, "Navigation of autonomous vehicles for oil spill cleaning in dynamic and uncertain environments," *International Journal of Control*, vol. 87, no. 4, pp. 787–801, 2014.
- [8] K. A. Ghamry, M. A. Kamel, and Y. Zhang, "Multiple uavs in forest fire fighting mission using particle swarm optimization," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*. IEEE, 2017, pp. 1404–1409.
- [9] H. M. Choset, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.
- [10] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks," *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 331–344, 2002.
- [11] B. Li, B. Moridian, and N. Mahmoudian, "Underwater multi-robot persistent area coverage mission planning," in *OCEANS 2016 MTS/IEEE Monterey*. IEEE, 2016, pp. 1–6.
- [12] V. An, Z. Qu, and R. Roberts, "A rainbow coverage path planning for a patrolling mobile robot with circular sensing range," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.
- [13] C. S. Kong, N. A. Peng, and I. Rekleitis, "Distributed coverage with multi-robot system," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006.*, May 2006, pp. 2423–2429.
- [14] Y. Zhou, H. Hu, Y. Liu, S. W. Lin, and Z. Ding, "A real-time and fully distributed approach to motion planning for multirobot systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. PP, no. 99, pp. 1–15, 2017black.
- [15] R. Cui, Y. Li, and W. Yan, "Mutual information-based multi-aup path planning for scalar field sampling using multidimensional rrt," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 993–1004, 2015black.
- [16] D. Göhring, M. Wang, M. Schnürmacher, and T. Ganjineh, "Radar/lidar sensor fusion for car-following on highways," in *The 5th International Conference on Automation, Robotics and Applications*. IEEE, 2011, pp. 407–412.
- [17] F. Zhang and N. E. Leonard, "Cooperative filters and control for cooperative exploration," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 650–663, 2010.
- [18] N. Mechat, N. Saadia, N. M'Sirdi, and N. Djelal, "Lane detection and tracking by monocular vision system in road vehicle," in *5th International Congress on Image and Signal Processing, USA, 2012*, pp. 1276 – 1282.
- [19] Y. Seo and R. R. Rajkumar, "Detection and tracking of boundary of unmarked roads," in *17th International Conference on Information Fusion (FUSION), Spain, 2014*, pp. 1–6.
- [20] M. Schwager, J.-J. Slotine, and D. Rus, "Unifying geometric, probabilistic, and potential field approaches to multi-robot coverage control," in *Robotics research*. Springer, 2011, pp. 21–38.
- [21] J. Alonso-Mora, A. Breitenmoser, M. Rufli, R. Siegwart, and P. Beardsley, "Multi-robot system for artistic pattern formation," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4512–4517.
- [22] S. Luo, J. H. Bae, and B.-C. Min, "Pivot-based collective coverage control with a multi-robot team," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2018, pp. 2367–2372.
- [23] D. W. Casbeer, D. B. Kingston, R. W. Beard, and T. W. McLain, "Cooperative forest fire surveillance using a team of small unmanned air vehicles," *International Journal of Systems Science*, vol. 37, no. 6, pp. 351–360, 2006.
- [24] Y. Qu, S. Xu, C. Song, Q. Ma, Y. Chu, and Y. Zou, "Finite-time dynamic coverage for mobile sensor networks in unknown environments using neural networks," *Journal of the Franklin Institute*, vol. 351, no. 10, pp. 4838–4849, 2014.
- [25] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: a collision cone approach," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 28, pp. 562 – 574, 1998.
- [26] B. Ranjbar-Sahraei, G. Weiss, and A. Nakisae, "A multi-robot coverage approach based on stigmergic communication," in *German Conference on Multiagent System Technologies*. Springer, 2012, pp. 126–138.
- [27] G. Zhang, G. K. Fricke, and D. P. Garg, "Spill detection and perimeter surveillance via distributed swarming agents," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 1, pp. 121–129, 2013.
- [28] J. H. Bae, S. Luo, S. S. Kannan, Y. Singh, B. Lee, R. M. Voyles, M. Postigo-Malaga, E. G. Zenteno, L. P. Aguilar, and B. Min, "Development of an unmanned surface vehicle for remote sediment sampling with a van veen grab sampler," in *OCEANS 2019 MTS/IEEE SEATTLE*, Oct 2019, pp. 1–7.
- [29] W. Jo, J.-H. Park, Y. Hoashi, and B.-C. Min, "Development of an unmanned surface vehicle for harmful algae removal," in *OCEANS 2019 MTS/IEEE SEATTLE*. IEEE, 2019, pp. 1–7.
- [30] R. Parasuraman, J. Kim, S. Luo, and B.-C. Min, "Multipoint rendezvous in multirobot systems," *IEEE transactions on cybernetics*, 2018.
- [31] K. Konolige, M. Agrawal, R. C. Bolles, C. Cowan, M. Fischler, and B. Gerkey, "Outdoor mapping and navigation using stereo vision," in *Experimental Robotics*. Springer, 2008, pp. 179–190.
- [32] L. Valbuena and H. G. Tanner, "Hybrid potential field based control of differential drive mobile robots," *Journal of intelligent & robotic systems*, vol. 68, no. 3–4, pp. 307–322, 2012.
- [33] D. E. Soltero, S. L. Smith, and D. Rus, "Collision avoidance for persistent monitoring in multi-robot systems with intersecting trajectories," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 3645–3652.
- [34] S. Qutub, R. Alami, and F. Ingrand, "How to solve deadlock situations within the plan-merging paradigm for multi-robot cooperation," in *Intelligent Robots and Systems, 1997. IROS'97., Proceedings of the 1997 IEEE/RSJ International Conference on*, vol. 3, 1997, pp. 1610–1615.
- [35] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008, pp. 1928–1935.
- [36] J. Sattar, P. Giguere, G. Dudek, and C. Prahacs, "A visual servoing system for an aquatic swimming robot," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 1483–1488.
- [37] J.-F. Lots, D. M. Lane, E. Trucco, and F. Chaumette, "A 2d visual servoing for underwater vehicle station keeping," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 3. IEEE, 2001, pp. 2767–2772.
- [38] S. Goyal, A. Prorok, and A. Martinoli, "Two-phase online calibration for infrared-based inter-robot positioning modules," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 3313–3319.
- [39] A. Mitiche and J. K. Aggarwal, *Computer vision analysis of image motion by variational methods*. Springer, 2014.
- [40] D. Pickem, M. Lee, and M. Egerstedt, "The gritsbot in its natural habitat—a multi-robot testbed," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4062–4067.
- [41] J. P. LaSalle, *The stability of dynamical systems*. Siam, 1976, vol. 25.
- [42] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Advances in neural information processing systems*, 2006, pp. 1161–1168.
- [43] D. Pickem, P. Grottel, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The robotarium: A remotely accessible swarm robotics research testbed," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1699–1706.
- [44] M. A. Batalin and G. S. Sukhatme, "Spreading out: A local approach to multi-robot coverage," in *Distributed Autonomous Robotic Systems 5*. Springer, 2002, pp. 373–382.