Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Sensor-based Exploration of an Unknown Area with Multiple Mobile Agents

Ertug Olcay [*,**] **Jens Bodeit** [*] **Boris Lohmann** [*]

[*] *Technical University of Munich, Department of Mechanical Engineering, Chair of Automatic Control, Munich, Germany (e-mail: {ertug.olcay, jens.bodeit, lohmann}@tum.de).*
[**] *Technical University of Munich, TUM School of Life Sciences, Chair of Agrimechatronics, Munich, Germany.*

Abstract: The research field of effective coverage of a certain area has received considerable attention, especially in exploration tasks. The ability of robots to localize themselves in a map and to plan elaborated motions are the basics of many coverage approaches. Cooperative, multiple robots can be employed in order to accelerate exploration missions. Over the past years, many methods have been investigated for this purpose. However, either the robots know the obstacle locations or they are not capable of identifying their environment completely. In this study, we propose a sensor-based framework to cover a given space simultaneously with multiple mobile agents in a cooperative fashion without any prior knowledge of the environment. With our approach, the agents are capable of avoiding collisions with different shaped obstacles and autonomously constructing a map of the whole area by identifying inaccessible domains in the map.

Keywords: Multi-agent systems, area exploration, cooperative control, motion planning.

1. INTRODUCTION

Efficient area coverage with multiple, autonomous robots (areal, underwater or ground vehicles) or mobile sensor networks is significant for various applications such as environmental monitoring, floor cleaning, data collection in certain environments, surveillance and reconnaissance (Ma et al., 2018). In addition, space mining, map creation of unknown areas, discovery and exploration of certain regions on a planet are other tasks requiring area coverage.

Deployment of multi-agent systems for area exploration increases the efficiency and success in many missions. However, decentralized coordination of multiple agents requires the ability to communicate, algorithms for cooperative motion and decision-making through information exchange.

There are several approaches for area coverage with single or multiple agents. Some of these methods are based on optimal control (Nguyen et al., 2016), game theory (Ramaswamy and Marden, 2016) and reinforcement learning (Adepegba et al., 2016). In dynamic area coverage, mobile agents have limited sensor ranges. Therefore, efficient motion planning is the key for mission success. In anti-flocking algorithms, agents try to move away from each other to improve coverage and explore new spaces in contrast to flocking behavior of swarming agents (Miao et al., 2010).

In order to maximize area coverage, the so-called anti-flocking algorithms for mobile sensor networks were proposed (Yuan et al., 2018b,a; Semnani and Basir, 2014). These algorithms distribute agents spatially in a systematic way. Combination of anti-flocking with potential field approach enables agents to detect targets in a large area with local communication and to avoid obstacles (Ganganath et al., 2016, 2018). Due to battery limitations, energy efficiency is an issue in area coverage tasks that were considered in studies (Yuan et al., 2018c; Koval et al., 2019).

Area coverage can be performed with decomposition-based approaches through partitioning of the exploration domain. These methods include Voronoi diagram (Kennedy et al., 2019; Papatheodorou et al., 2018; Miah et al., 2018), Boustrophedon decomposition (Koval et al., 2019) and Morse decomposition (Acar and Choset, 2002). Furthermore, there are ergodicity-based frameworks that consider target probability distribution (Salman et al., 2017; Ayvali et al., 2017; Ivić et al., 2016).

Autonomous agents utilize different kinds of path planning techniques for area coverage. One approach is global path planning, which requires prior knowledge of the environment. Another method is local path planning, which obligates the robot to plan its motion solely based on its sensor measurements. The advantage of this method is that the agent can be operated in any unknown environment without having prior information.

In our study, we consider a group of mobile agents with limited communication bandwidths and sensing performance. As a result, they can communicate only with the agents within their communication range and detect objects in their sensing range. The main contribution of this paper is cooperative identification and coverage of an

⋆ The authors thank the German Research Foundation (DFG) for partly funding this work as part of the Collaborative Research Centre 768 (SFB 768).

unknown environment with multiple autonomous agents. Our approach builds upon the anti-flocking algorithm (Ganganath et al., 2016) and extends it with an improved obstacle avoidance and recognition of inaccessible areas. Through local information exchange and judicious path planning for the recognition of restricted areas inside of obstacles in the exploration domain, agents are able to identify their environment fully autonomously in a safe and efficient way.

The outline of this paper is organized as follows: In Section 2, preliminaries for the proposed framework are briefly introduced. Section 3 describes the problem in sensor-based area coverage. In Section 4, our cooperative area exploration and identification scheme is presented. The proposed exploration scheme is evaluated through simulation scenarios in Section 5. Finally, the conclusion of this study is presented in Section 6.

2. PRELIMINARIES

In this section, we introduce the dynamics of the mobile agents for the exploration-oriented path planning. We consider the agents as point-masses and the dynamics are based on potential functions. The agents receive virtual repulsive forces by following the gradient of a potential field. The state of each agent is defined by $(\mathbf{p}_i, \mathbf{v}_i) \in \mathbb{R}^m \times \mathbb{R}^m$, which represent the position and velocity of the i-th agent in m dimensional space, respectively. The dynamics of a group of n agents are given by the following system of differential equations:

$$\begin{aligned} \dot{\mathbf{p}}_i &= \mathbf{v}_i, \\ \dot{\mathbf{v}}_i &= \mathbf{u}_i, \end{aligned} \tag{1}$$

for $i = 1, \ldots, n$, $n \in \mathbb{N}$. If the Euclidean distance between two agents is less than r, they are called neighbors. The neighborhood of agent i is defined as

$$\mathcal{N}_i = \{j \in \mathcal{V} : \|\mathbf{p}_j - \mathbf{p}_i\| < r\}, \tag{2}$$

where $\mathcal{V} = \{1, \ldots, n\}$ is a set of all agents. The dynamics of agent interactions are defined by distance-dependent potential functions. In order to ensure the differentiability of these functions at $z = 0$, a mapping, the so called σ-norm (Olfati-Saber, 2006), is introduced as follows:

$$\|z\|_\sigma = \frac{1}{\epsilon}[\sqrt{1 + \epsilon\|z\|^2} - 1] \tag{3}$$

with a fixed parameter $\epsilon > 0$. In addition, we define the following *bump function* as in (Olfati-Saber, 2006):

$$\rho_h(z) = \begin{cases} 1, & z \in [0, h) \\ \frac{1}{2}[1 + \cos(\pi \frac{z - h}{1 - h})], & z \in [h, 1) \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $h \in (0, 1)$. The function (4) is a smooth steadily dissipative function and we use it later to weight the adjacency matrix

$$a_{ij}(\mathbf{p}) = \rho_h\left(\frac{\|\mathbf{p}_j - \mathbf{p}_i\|_\sigma}{r_\alpha}\right) \in [0, 1], \qquad j \neq i \tag{5}$$

with $r_\alpha = \|r\|_\sigma$. If the corresponding distance between agent i and j greater than r_α, then $\|a_{ij}\| = 0$.

3. PROBLEM DESCRIPTION

In this study, we assume that every agent has only a range sensor with a limited measuring range. In addition, it can only measure the relative position of its neighbors and exchange information only with other agents within its limited communication bandwidth. The measurement noise and sensor uncertainties are neglected. For dealing with noisy measurements, the reader is referred to distributed Kalman Filter (Olfati-Saber, 2007). Moreover, each agent is able to localize itself in the map.

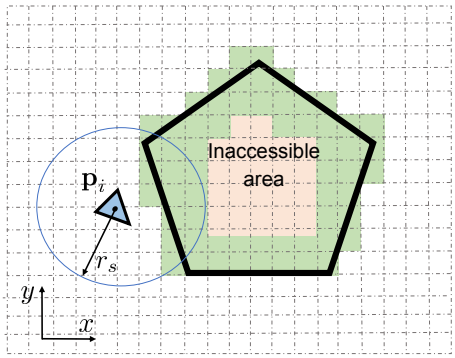


Fig. 1. Illustration of the identification issue in an exploration task. Green-colored cells represent identifiable areas of the obstacle.

Over the years, various sensor-based approaches have been proposed to cover an area with multiple agents. Most of them are based on information exchange to plan an optimal coverage motion without overlapping of the sensing regions. In *grid*-based methods that divide the exploration area into very fine cells such as in (Ganganath et al., 2018, 2016; Yuan et al., 2018b), the coverage task is accomplished completely, when the agent has sensed and identified each cell in the exploration area.

In real-life applications, area boundaries are usually given for the search task and obstacle locations are not known. The agents should autonomously identify each *cell* in the virtual map. However, some cells in the area are inaccessible due to obstacles or some other restrictions in the domain (Fig. 1). Hence, they cannot be sensed by the agents and remain *unidentified*.

4. COOPERATIVE EXPLORATION SCHEMA

In order to address the described issue in Section 3, we propose an exploration method which consists of four components: Motion planning, anti-flocking with local communication, obstacle recognition and circumnavigation.

4.1 Motion Planning of Mobile Agents

The considered system in this study has the following double-integrator dynamics:

$$\begin{aligned} \dot{\mathbf{p}}_i &= \mathbf{v}_i, \\ \dot{\mathbf{v}}_i &= \mathbf{u}_i^\alpha + \mathbf{u}_i^\beta + \mathbf{u}_i^\gamma, \end{aligned} \tag{6}$$

where $\mathbf{u}_i^\alpha$ is the control input for the inter-agent *collision avoidance*, $\mathbf{u}_i^\beta$ is for the *obstacle avoidance* and $\mathbf{u}_i^\gamma$ is for the *tracking* of a target point. Each agent can receive the position of other agents within the communication range r_c and also the relative position of the closest point on an

obstacle within its sensor range r_s. In this way, each agent can identify the set of neighboring agents

$$\mathcal{N}_i^\alpha = \{j \in \mathcal{V} : \|\mathbf{p}_j - \mathbf{p}_i\| < r_c\}, \tag{7}$$

and the set of detected obstacle points

$$\mathcal{N}_i^\beta = \{\hat{\mathbf{p}}_{i,k} \big| \ \|\hat{\mathbf{p}}_{i,k} - \mathbf{p}_i\| < r_s\}, \tag{8}$$

at a given time, where $\hat{\mathbf{p}}_{i,k}$ is the position of the closest point on the obstacle k detected by the agent. The control input to keep a safe distance d_a to other agents is given by

$$\mathbf{u}_i^\alpha = c_1^\alpha \sum_{j \in \mathcal{N}_i^\alpha} \phi_\alpha(\|\mathbf{p}_j - \mathbf{p}_i\|_\sigma)\mathbf{n}_{ij} + c_2^\alpha \sum_{j \in \mathcal{N}_i^\alpha} a_{ij}(\mathbf{p})(\mathbf{v}_j - \mathbf{v}_i), \tag{9}$$

where $\mathbf{n}_{ij} = \sigma_\epsilon(\mathbf{p}_j - \mathbf{p}_i) = \dfrac{\mathbf{p}_j - \mathbf{p}_i}{\sqrt{1 + \epsilon\|\mathbf{p}_j - \mathbf{p}_i\|^2}}$ a vector from $\mathbf{p}_j$ to $\mathbf{p}_i$ and $\epsilon \in (0, 1)$.

We formulate the following control input to avoid collision with obstacles and to keep a safe distance d_o:

$$\mathbf{u}_i^\beta = c_1^\beta \sum_{k \in \mathcal{N}_i^\beta} \phi_\beta(\|\hat{\mathbf{p}}_{i,k} - \mathbf{p}_i\|_\sigma)\hat{\mathbf{n}}_{i,k} + c_2^\beta \sum_{k \in \mathcal{N}_i^\beta} b_{i,k}(\mathbf{p})(\hat{\mathbf{v}}_{i,k} - \mathbf{v}_i), \tag{10}$$

with

$$\hat{\mathbf{n}}_{i,k} = \frac{\hat{\mathbf{p}}_{i,k} - \mathbf{p}_i}{\sqrt{1 + \epsilon\|\hat{\mathbf{p}}_{i,k} - \mathbf{p}_i\|^2}}, \ b_{i,k}(\mathbf{p}) = \rho_{h_\beta}\left(\frac{\|\hat{\mathbf{p}}_{i,k} - \mathbf{p}_i\|_\sigma}{d_\beta}\right),$$

where $\hat{\mathbf{v}}_{i,k}$ is the projection of the $\mathbf{v}_i$ onto the edge of the obstacle k. Moreover, $b_{i,k}(\mathbf{p})$ describes the weighted *adjacency* between agent i and obstacle k. The repulsive potential forces are defined with the bump function (4) as follows:

$$\phi_\alpha(z) = \rho_{h_\alpha}(z/d_\alpha)(\sigma_1(z - d_\alpha) - 1), \tag{11}$$

$$\phi_\beta(z) = \rho_{h_\beta}(z/d_\beta)(\sigma_1(z - d_\beta) - 1), \tag{12}$$

where $d_\beta < r_\beta$ with $d_\beta = \|d_o\|_\sigma$, $r_\beta = \|r_s\|_\sigma$ and $d_\alpha = \|d_a\|_\sigma$. Finally, the control term for the *navigation* is defined as

$$\mathbf{u}_i^\gamma = -c_1^\gamma(\mathbf{p}_i - \mathbf{p}_i^t) - c_2^\gamma \mathbf{v}_i, \tag{13}$$

where $\mathbf{p}_i^t \in \mathbb{R}^m$ represents the *target* position of agent i at time instant $t_k > 0$. Here, c_η^ν are positive constant parameters for all $\eta = 1, 2$ and $\nu = \alpha, \beta, \gamma$.

4.2 Anti-flocking and Collective Map Creation

In order to maximize the identified domains, we apply the anti-flocking algorithm presented in study (Ganganath et al., 2016). First, similar to (Yuan et al., 2018b) the exploration area (EA) is divided into equal-sized cells and each agent generates its own information map $\mathbf{M}_i$, which is considered as a matrix. The elements of the matrix $\mathbf{M}_i = [m_i(\mathbf{x}_{\mu,\nu})]$ represent the *identified* ($m_i(\mathbf{x}_{\mu,\nu}) = 1$), *unidentified* ($m_i(\mathbf{x}_{\mu,\nu}) = 0$), and *occupied* ($m_i(\mathbf{x}_{\mu,\nu}) = -1$) cells. $m_i(\mathbf{x}_{\mu,\nu}) \in \{0, 1, -1\}$ is also called the sensing information. $\mathbf{x}_{\mu,\nu}$ denotes the coordinates of a cell center. μ and ν represent the rows and the columns of the information map $\mathbf{M}_i$, respectively.

According to the anti-flocking algorithm, the *target* position $\mathbf{p}_i^t$ of an agent is calculated, evaluating the benefit function given by

$$\xi_i = (1 - |m_i(\mathbf{x}_{\mu,\nu})|)(\rho_\gamma + (1 - \rho_\gamma)\lambda_i(\mathbf{x}_{\mu,\nu})), \tag{14}$$

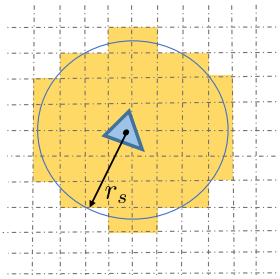


Fig. 2. Visual representation of area sensing. The yellow-colored cells are identified by the agent.

where ρ_γ is a constant. The function λ_i is defined as

$$\lambda_i(\mathbf{x}_{\mu,\nu}) = exp(-\kappa_1\|\mathbf{p}_i - \mathbf{x}_{\mu,\nu}\| - \kappa_2\|\mathbf{p}_i^t - \mathbf{x}_{\mu,\nu}\|) \tag{15}$$

with positive constants κ_1 and κ_2. In this way, each cell has a benefit value based on (14). The target position is selected for the time instant $t_k + 1$ as follows:

$$\mathbf{p}_i^t(t_k + 1) = \underset{\mathbf{x}_{\mu,\nu} \in \tilde{\mathcal{X}}_i}{\operatorname{argmax}} \xi_i(\mathbf{x}_{\mu,\nu}, t_k), \tag{16}$$

where $\tilde{\mathcal{X}}_i = \{\mathbf{x}_{\mu,\nu} | \mathbf{x}_{\mu,\nu} \in \mathcal{X}, \|\mathbf{x}_{\mu,\nu} - \mathbf{p}_j\| \geq \|\mathbf{x}_{\mu,\nu} - \mathbf{p}_i\| > r_s, j \in \mathcal{N}_i^\alpha\}$. Here, $\mathcal{X}$ is the set of all cell center points.

For each agent i, whose communication range is r_c, the information map is updated in the following way:

- In the beginning of the exploration ($t_k = 0$), all cells in the information map $\mathbf{M}_i$ are unidentified.
- Once an area is sensed, the cells $m_i(\mathbf{x}_{\mu,\nu})$ with $\|\mathbf{x}_{\mu,\nu} - \mathbf{p}_i\| \leq r_s$ are defined as *identified* (Fig. 2). In addition, if an obstacle is detected, the cell which includes the sensed obstacle point $\hat{\mathbf{p}}_i$ is saved as *occupied*.
- If an agent j is in the communication range of the agent i,
$$\|\mathbf{p}_j - \mathbf{p}_i\| < r_c,$$
the agents can exchange missing information and complete each others maps,
$$\mathbf{M}_i(t_k) = \mathbf{M}_j(t_k).$$

In addition, $\mathbf{p}_i^t$ is updated based on the *recalculation criteria* (Ganganath et al., 2016) to reduce overlapping sensing ranges and to minimize the traveling effort.

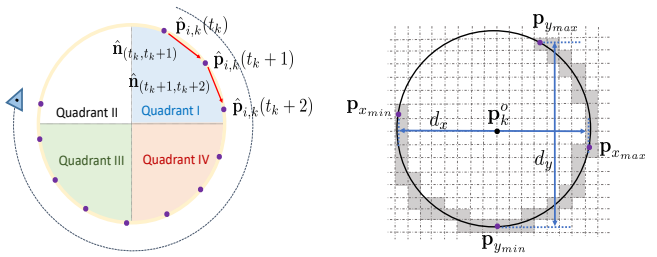
4.3 Obstacle Recognition

Obstacle recognition can be easily included into the *anti-flocking* framework with a novel algorithm, which allows agents to identify inaccessible areas on their own. The proposed algorithm can categorize the shape of obstacles and restricted areas. Each agent is capable of memorizing the sensed obstacle points $\hat{\mathbf{p}}_{i,k}(t_k)$ over the discrete time.

The obstacle classification is based on the rate of change of tangential vectors. Hence, the agents calculate *difference vectors* $\hat{\mathbf{n}}_{(t_k,t_k+1)}$ between consecutive sensed points as

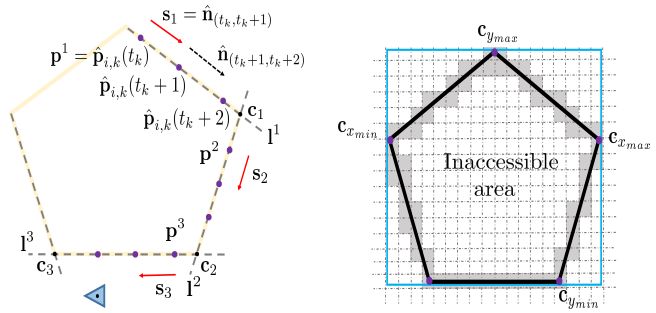
$$\hat{\mathbf{n}}_{(t_k,t_k+1)} = \frac{\hat{\mathbf{p}}_{i,k}(t_k + 1) - \hat{\mathbf{p}}_{i,k}(t_k)}{\|\hat{\mathbf{p}}_{i,k}(t_k + 1) - \hat{\mathbf{p}}_{i,k}(t_k)\|}. \tag{17}$$

The cells, which include the sensed points, are marked in the information map as occupied at the same time. Since the distances between these sensed points are very short



(a) Recognition scheme for circles (b) Obstacle map

Fig. 3. Illustration of the circle recognition.



(a) Recognition scheme for polygons (b) Obstacle map

Fig. 4. Illustration of the polygon recognition. The gray colored cells represent occupied areas.

due to a small time step Δt, difference vectors can be considered as approximated tangential vectors. An obstacle is classified as a polygon if the consecutive difference vectors are piecewise identical. However, if the rate of change of difference vectors is not zero, the following scheme for circle recognition is applied, which is illustrated in Fig. 3.

1) Primarily, for the algorithm, the circle is divided virtually into quadrants.
2) Based on the sign change of components of the difference vectors $\{\hat{\mathbf{n}}_{(t_k,t_k+1)},\dots,\hat{\mathbf{n}}_{(t_k+n,t_k+n+1)}\}$, the transitions between consecutive quadrants are detected (see Table A.1).
3) After three transitions to different consecutive quadrants, the algorithm assumes that all four quadrants were recognized by the agent and the next step is performed. Note that the sign change of both components of the difference vectors is alternating (see Appendix A).
4) During circumnavigation, all sensed points $\mathcal{P}_i = \{\hat{\mathbf{p}}_{i,k}(t_k), \hat{\mathbf{p}}_{i,k}(t_k+1),\dots,\hat{\mathbf{p}}_{i,k}(t_k+n)\}$ are stored. In order to define the diameter of the circle, we utilize extrema of sensed points as follows:

$$\mathbf{p}_{x_{max}} = \underset{\hat{\mathbf{p}}_{i,z}\in\mathcal{P}_i}{\operatorname{argmax}} \left[\hat{\mathbf{p}}_{i,z}^T \cdot \begin{pmatrix}1\\0\end{pmatrix}\right], \quad \mathbf{p}_{x_{min}} = \underset{\hat{\mathbf{p}}_{i,z}\in\mathcal{P}_i}{\operatorname{argmin}} \left[\hat{\mathbf{p}}_{i,z}^T \cdot \begin{pmatrix}1\\0\end{pmatrix}\right]$$

$$\mathbf{p}_{y_{max}} = \underset{\hat{\mathbf{p}}_{i,z}\in\mathcal{P}_i}{\operatorname{argmax}} \left[\hat{\mathbf{p}}_{i,z}^T \cdot \begin{pmatrix}0\\1\end{pmatrix}\right], \quad \mathbf{p}_{y_{min}} = \underset{\hat{\mathbf{p}}_{i,z}\in\mathcal{P}_i}{\operatorname{argmin}} \left[\hat{\mathbf{p}}_{i,z}^T \cdot \begin{pmatrix}0\\1\end{pmatrix}\right]$$

The diameter of the circle is defined as $d = max(d_x, d_y)$, where

$$d_x = (1,0)\cdot(\mathbf{p}_{x_{max}} - \mathbf{p}_{x_{min}}),$$
$$d_y = (0,1)\cdot(\mathbf{p}_{y_{max}} - \mathbf{p}_{y_{min}}).$$

Furthermore, the circle origin is estimated as

$$\mathbf{p}_k^o = \begin{cases} \frac{1}{2}\left(\mathbf{p}_{x_{min}} + \mathbf{p}_{x_{max}}\right) & \text{if } d_x > d_y, \\ \frac{1}{2}\left(\mathbf{p}_{y_{min}} + \mathbf{p}_{y_{max}}\right) & \text{if } d_x < d_y. \end{cases}$$

5) All cells whose distance from center point to the origin of the circle is less than the radius $d/2$ are defined as *occupied* in the map.

Recognition of a polygon is described in the following and Fig. 4 illustrates the recognition scheme.

1) The identified sides of a polygon are defined by the tuples $(\mathbf{s}_n, \mathbf{p}^n)$ with $n \in \{1,\dots,N\}$. $N \in \mathbb{N}$ is the number of the detected polygon sides. $\mathbf{s}_n$ and $\mathbf{p}^n$ denote the tangential vector of a side and first sensed point on the side, respectively. The tangential vector is equal to one of the difference vectors described in (17). In this way, a list of tuples is created to describe a polygon.
2) Polygons are two-dimensional objects that consist of at least three sides. Hence, the list of tuples created in the first step should include at least three different difference vectors to interpret an obstacle as a closed area.
3) Once the first clustered direction $\mathbf{s}_1$ is determined for the second time, the next step of the algorithm is performed.
4) With the help of the tuples $\{(\mathbf{s}_1,\mathbf{p}^1),(\mathbf{s}_2,\mathbf{p}^2),\dots,(\mathbf{s}_N,\mathbf{p}^N)\}$, we define N straight lines as follows:
$$\mathbf{l}^n: \quad \mathbf{x} = \mathbf{p}^n + q\cdot\mathbf{s}_n, \quad q\in\mathbb{R}$$
where $\mathbf{x}$ is an arbitrary point on the line. Then, we determine the points of intersection of consecutive lines in order to define the vertex points of the polygon $\mathcal{C}_i = \{\mathbf{c}_1,\dots\mathbf{c}_N\}$.
5) Furthermore, the section in the map (Fig. 4(b), blue-colored frame) with the obstacle is considered closer using the extreme coordinate values of vertex points

$$\mathbf{c}_{x_{max}} = \underset{\mathbf{c}_z\in\mathcal{C}_i}{\operatorname{argmax}} \left[\mathbf{c}_z^T \cdot \begin{pmatrix}1\\0\end{pmatrix}\right], \quad \mathbf{c}_{x_{min}} = \underset{\mathbf{c}_z\in\mathcal{C}_i}{\operatorname{argmin}} \left[\mathbf{c}_z^T \cdot \begin{pmatrix}1\\0\end{pmatrix}\right],$$

$$\mathbf{c}_{y_{max}} = \underset{\mathbf{c}_z\in\mathcal{C}_i}{\operatorname{argmax}} \left[\mathbf{c}_z^T \cdot \begin{pmatrix}0\\1\end{pmatrix}\right], \quad \mathbf{c}_{y_{min}} = \underset{\mathbf{c}_z\in\mathcal{C}_i}{\operatorname{argmin}} \left[\mathbf{c}_z^T \cdot \begin{pmatrix}0\\1\end{pmatrix}\right].$$

6) If a cell center point in the considered section is surrounded by occupied cells in all x- and y-directions, it is defined as *occupied*.

Remark 1. If an agent leaves the obstacle or changes its direction of motion for a short time during the circumnavigation, but then returns to the obstacle, one side of the polygon might be considered more than once. In order to eliminate this, the following case differentiation is implemented:

$$\boldsymbol{s}_n = \begin{cases} -\hat{\boldsymbol{n}}_{(t_k,t_{k+1})}, & \text{if } \left[(1\,,\,0)\cdot\hat{\boldsymbol{n}}_{(t_k,t_{k+1})}\right] < 0 \\ \hat{\boldsymbol{n}}_{i,(t_k,t_{k+1})}. & \text{otherwise} \end{cases} \tag{18}$$

Remark 2. The presented scheme does not allow recognition of irregular-shaped and hybrid obstacles, where a subset of difference vectors indicate a circle and another one a polygon.

4.4 Circumnavigation

The proposed algorithm requires *circumnavigation*, which means a complete navigation around an entire obstacle. Our navigation concept with the continuous motion along an obstacle surface is mainly inspired by the tangential navigation scheme presented in (Brandão et al., 2013).

Once an obstacle is in the range of d_{tan} ($r_s > d_{tan} > d_o$), the circumnavigation is activated. In the first step, the agent determines its rotation angle γ as follows:

$$\gamma = \begin{cases} \beta - \alpha - 90°, & \text{for } \beta \geq 0° \\ \beta - \alpha + 90°, & \text{for } \beta < 0° \end{cases} \tag{19}$$

where the angle α and β are the orientation of the agent relative to the target position and the orientation to the closest obstacle point, respectively (see Fig. 5). The angle γ is used as a rotation angle for the projection of the *temporary target* position $\mathbf{p}_i^{tt}$ calculated as

$$\mathbf{p}_i^{tt} = \mathbf{p}_i + \begin{bmatrix} cos(\gamma) & -sin(\gamma) \\ sin(\gamma) & cos(\gamma) \end{bmatrix} (\mathbf{p}_i^t - \mathbf{p}_i). \tag{20}$$

The navigation of the agent is performed similar to (13) as follows:

$$\mathbf{u}_i^\gamma = -c_1^\gamma \left(c_n \cdot \frac{\mathbf{p}_i - \mathbf{p}_i^{tt}}{\|\mathbf{p}_i - \mathbf{p}_i^{tt}\|} \right) - c_2^\gamma \mathbf{v}_i, \tag{21}$$

where c_1^γ and c_2^γ are positive constant weighting factors to have a controlled acceleration during the tangential navigation.

Once the distance between the agent and an obstacle holds $\|\hat{\mathbf{p}}_{i,k} - \mathbf{p}_i\| > d_{tan}$ during surface following, the agent switches to a new strategy utilizing the last sensed point $\hat{\mathbf{p}}_{i,e} = \hat{\mathbf{p}}_{i,k}(t_k - 1)$ on the obstacle with $\|\hat{\mathbf{p}}_{i,k}(t_k - 1) - \mathbf{p}_i\| \leq d_{tan}$. We define a new rotation angle $\delta > 0$, through which we set the temporary target point as follows:

$$\mathbf{p}_i^{tt} = \begin{cases} \hat{\mathbf{p}}_{i,e} + d_{tan} \begin{bmatrix} cos(\delta) & -sin(\delta) \\ sin(\delta) & cos(\delta) \end{bmatrix} \mathbf{n}_{i,e} & \text{if } \beta \geq 0° \\ \hat{\mathbf{p}}_{i,e} + d_{tan} \begin{bmatrix} cos(\delta) & sin(\delta) \\ -sin(\delta) & cos(\delta) \end{bmatrix} \mathbf{n}_{i,e} & \text{if } \beta < 0° \end{cases} \tag{22}$$

with

$$\mathbf{n}_{i,e} = \frac{\mathbf{p}_i - \hat{\mathbf{p}}_{i,e}}{\|\mathbf{p}_i - \hat{\mathbf{p}}_{i,e}\|},$$

where the $\hat{\mathbf{p}}_{i,e}$ is the reference point for the calculation (see Fig. 5). Once the first $\mathbf{p}_i^{tt}$ is reached ($\|\mathbf{p}_i^{tt} - \mathbf{p}_i\| < 0.7$), the agent sets a new temporary target position based on (22). If a new point $\hat{\mathbf{p}}_{i,k}$ with $\|\hat{\mathbf{p}}_{i,k} - \mathbf{p}_i\| < d_{tan}$ is sensed, the navigation is performed based on (20). In the meantime, the following condition should be satisfied to leave the obstacle extremity:

$$|\alpha| \leq \omega, \tag{23}$$

where ω represents a tolerance angle range. In the absence of a sensed point, the condition (23) helps the agent stop setting new temporary positions if the deviation between its motion $\mathbf{v}_i$ and the vector from its current position $\mathbf{p}_i$ to the target position $\mathbf{p}_i^t$ is in a certain range.

Remark 3. The presented circumnavigation approach can be extended to overcome local minima due to concave obstacles by integrating corner avoidance scheme presented in (Brandão et al., 2013).

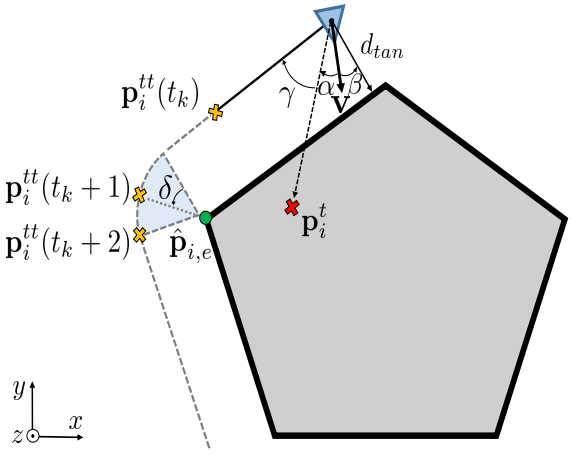


Fig. 5. Visual representation of the circumnavigation.

5. SIMULATION RESULTS

In this section, we evaluate the proposed online sensor-based scheme. With the introduced approach, agents successfully discover the unknown area in the simulated scenario.

In the simulation, we consider a system of $n = 3$ agents in the $m = 2$ dimensional plane. The agents are randomly placed in the EA and their initial velocities $\mathbf{v}_i(0) \in \mathbb{R}^2$ are $(0, 0)^T$.

The EA is defined in a domain $[-50, 50] \times [-50, 50]$ with a cell size of $[1 \times 1]$. The values of the anti-flocking parameters, the weighting factors of the control terms and the parameters for the circumnavigation are given in Table 1. In the simulation, the input $|\mathbf{u}_i^\gamma|$ is bounded by $|\mathbf{u}_{max}| = 8$ to avoid too high accelerations. The priorities of the control objectives are defined: *Collision avoidance*, *obstacle avoidance* and *tracking* with $c_1^\beta = c_1^\alpha > c_1^\gamma$.

Fig. 6 depicts the positions and trajectories of agents during an exploration mission without using the obstacle recognition scheme at the time instant $t = 23\,s$. The EA and the area coverage through all agents are also illustrated in this figure. The EA contains one polygon and one circular obstacle that are not known to agents beforehand. The white triangles represent the positions and the directions of motion of the agents. The yellow- and blue-colored areas represent the identified and unidentified regions, respectively. The gray-colored areas are occupied or inaccessible domains identified by agents during the exploration. It is observed that the agents are in a continuous circumnavigation mode and try to cover the inner areas of obstacles.

Fig. 7 shows the positions and trajectories of the agents for important time instants with obstacle recognition and tangential navigation scheme. The EA includes one polygon and one circular obstacle. In addition, the exchange of information map enables the group to make optimal decisions about the next target position for efficient exploration.

At $t = 9\,s$, an agent detects the polygon-shaped obstacle and another agent senses the circular one. After the

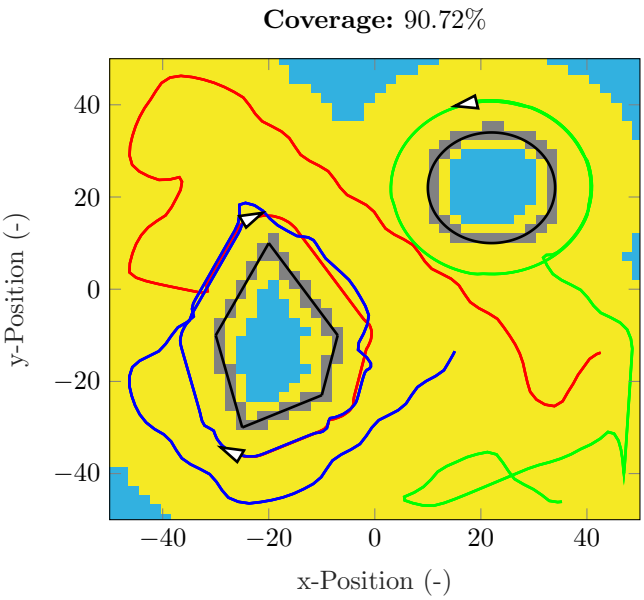


Fig. 6. Exploration task without obstacle recognition with $n = 3$ agents at $t = 23\,s$.

Table 1. Parameter Setting

Anti-flocking	r_c	r_s	d_{tan}	d_o
	40	10	7	6
	h_α	h_β	ϵ	d_a
	0.2	0.9	0.08	12
	κ_1	κ_2	ρ_γ	
	0.04	0.01	0.2	
Weighting of the control terms	c_1^α	c_1^β	c_1^γ	
	60	60	30	
	c_2^α	c_2^β	c_2^γ	
	$2\sqrt{c_1^\alpha}$	$2\sqrt{c_1^\beta}$	$2\sqrt{c_1^\gamma}$	
Circumnavigation	δ	ω	c_n	
	20°	10°	4	

circumnavigation, the agents are able to identify the inner area of the obstacles and define them as occupied domains. It can be seen that the agents can avoid obstacles and identify inaccessible areas on their own.

6. CONCLUSION

In this paper, we introduced a novel sensor-based exploration framework that allows agents to identify each cell in the exploration area by recognizing inaccessible domains autonomously. With the proposed scheme, multiple agents can efficiently explore an unknown area with predefined boundaries as a group through local information exchange. However, the main contribution is that the agents are capable of identifying edged and circular obstacles or restricted areas on their own without having prior knowledge. Combination of the proposed obstacle recognition with the circumnavigation approach makes it possible for the agents to identify inaccessible areas by evaluating only sensor data. Moreover, exploration can also be easily performed in more complex environments with concave obstacles by extending the circumnavigation scheme. Future work will focus on the identification of irregular-shaped obstacles.

REFERENCES

Acar, E.U. and Choset, H. (2002). Sensor-based coverage of unknown environments: Incremental construction of morse decompositions. *The International Journal of Robotics Research*, 21(4), 345–366.

Adepegba, A.A., Miah, S., and Spinello, D. (2016). Multi-agent area coverage control using reinforcement learning. *The Twenty-Ninth International Flairs Conference*.

Ayvali, E., Salman, H., and Choset, H. (2017). Ergodic coverage in constrained environments using stochastic trajectory optimization. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Brandão, A.S., Sarcinelli-Filho, M., and Carelli, R. (2013). An analytical approach to avoid obstacles in mobile robot navigation. *International Journal of Advanced Robotic Systems*, 10(6), 278.

Ganganath, N., Cheng, C., and Tse, C.K. (2016). Distributed antiflocking algorithms for dynamic coverage of mobile sensor networks. *IEEE Transactions on Industrial Informatics*, 12(15), 1795–1805.

Ganganath, N., Yuan, W., Cheng, C.T., Fernando, T., and Iu, H.H. (2018). Territorial marking for improved area coverage in anti-flocking-controlled mobile sensor networks. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*.

Ivić, S., Crnković, B., and Mezić, I. (2016). Ergodicity-based cooperative multiagent area coverage via a potential field. *IEEE Transactions on Cybernetics*, 47(8).

Kennedy, J., Chapman, A., and Dower, P.M. (2019). Generalized coverage control for time-varying density functions. In *2019 18th European Control Conference (ECC)*, 71–76.

Koval, A., Mansouri, S.S., and Nikolakopoulos, G. (2019). Online multi-agent based cooperative exploration and coverage in complex environment. In *2019 18th European Control Conference (ECC)*, 3964–3969. IEEE.

Ma, L., He, F., Wang, L., and Yao, Y. (2018). Multi-agent coverage control design with dynamic sensing regions. *Control Theory and Technology*, 16(3), 161–172.

Miah, S., Nguyen, B., Bourque, A., and Spinello, D. (2018). Non-autonomous area coverage and coordination of a multi-agent system for harbor protection applications. *Proceedings of 8th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, 1(15), 486–492.

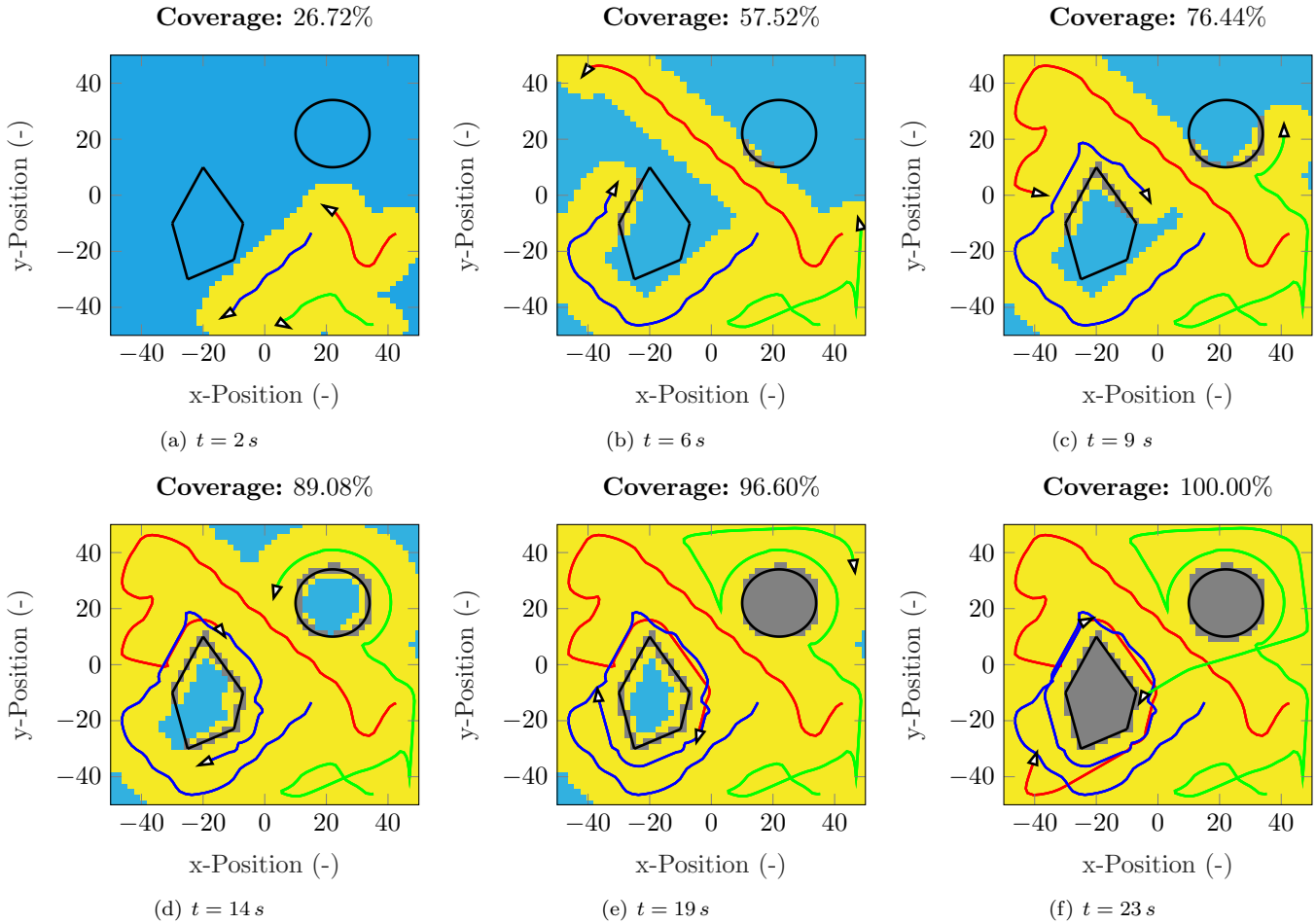
Miao, Y.Q., Khamis, A., and Kamel, M.S. (2010). Applying anti-flocking model in mobile surveillance systems. In *2010 International Conference on Autonomous and Intelligent Systems, AIS 2010*, 1–6. IEEE.

Nguyen, M.T., Rodrigues, L., Maniu, C.S., and Olaru, S. (2016). Discretized optimal control approach for dynamic multi-agent decentralized coverage. *IEEE International Symposium Intelligent Control (ISIC)*.

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on automatic control*, 51(3), 401–420.

Olfati-Saber, R. (2007). Distributed kalman filtering for sensor networks. In *46th IEEE Conference on Decision and Control*, 5492–5498.

Papatheodorou, S., Tzes, A., Giannousakis, K., and Stergiopoulos, Y. (2018). Distributed area coverage control with imprecise robot localization: Simulation and experimental studies. *International Journal of Advanced*



(a) $t = 2\,s$ (b) $t = 6\,s$ (c) $t = 9\ s$

(d) $t = 14\,s$ (e) $t = 19\,s$ (f) $t = 23\,s$

Fig. 7. Consecutive snapshots of an exploration task with $n = 3$ agents with obstacle recognition.

Robotic Systems, 15(5).

Ramaswamy, V. and Marden, J.R. (2016). A sensor coverage game with improved efficiency guarantees. *American Control Control Coference (ACC)*.

Salman, H., Ayvali, E., and Choest, H. (2017). Multi-agent ergodic coverage with obstacle avoidance. *Proceedings of the 27th International Conference on Automated Planning and Scheduling (ICAPS)*.

Semnani, S.H. and Basir, O.A. (2014). Semi-flocking algorithm for motion control of mobile sensors in large-scale surveillance systems. *IEEE Transactions on Cybernetics*, 45(1), 129–137.

Yuan, W., Ganganath, N., Cheng, C., Qing, G., and Lau, F.C.M. (2018a). Path planning for semi-flocking-controlled mobile sensor networks on mobility maps. *IEEE International Symposium on Circuits and Systems*.

Yuan, W., Ganganath, N., Cheng, C.T., Qing, G., and Lau, F.C. (2018b). Semi-flocking-controlled mobile sensor networks for dynamic area coverage and multiple target tracking. *IEEE Sensors Journal*, 18(21), 8883–8892.

Yuan, W., Ganganath, N., Cheng, C.T., Valaee, S., Guo, Q., and Lau, F.C. (2018c). Energy-efficient semi-flocking control of mobile sensor networks on rough terrains. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66(4), 622–626.

Quadrant	Curve parameter	x	y
I	$s \in (0, \frac{\pi}{2})$	< 0	> 0
II	$s \in (\frac{\pi}{2}, \pi)$	< 0	< 0
III	$s \in (\pi, \frac{3}{2}\pi)$	> 0	< 0
IV	$s \in (\frac{3}{2}\pi, 2\pi)$	> 0	> 0

Table A.1. Components of the tangential vectors in the quadrants of a circle.

Appendix A. CIRCLE IN THE PLANE

A circle with the radius r is a continuously differentiable, closed curve θ, which can be described with the following parameterization in cartesian coordinates:

$$\theta : [0, 2\pi] \to \mathbb{R}^2, \quad \theta(s) = \begin{pmatrix} r\cos(s) \\ r\sin(s) \end{pmatrix}. \tag{A.1}$$

The tangential vector $\boldsymbol{\tau}(s) = (x, y)^T$, which lies at the point $\boldsymbol{p}(s) = \theta(s)$ on the circle, is given by

$$\boldsymbol{\tau}(s) = \dot{\theta}(s) = \begin{pmatrix} -r\sin(s) \\ r\cos(s) \end{pmatrix}. \tag{A.2}$$

The corresponding tangent is a straight line defined by $\boldsymbol{p}(s)$ and $\boldsymbol{\tau}(s)$. From (A.2) follows $\dot{\boldsymbol{\tau}}(s) = \ddot{\theta}(s) \neq \mathbf{0} \quad \forall s \in [0, 2\pi]$, the tangential vector defined as a function of the curve parameter s is never constant. A transition from one quadrant to the next yields either the change of the sign of the x or the y component (Fig. 3(a)). This is summarized in Table A.1.