

Distributed Event- and Self-Triggered Coverage Control with Speed Constrained Unicycle Robots

Yuni Zhou, Lingxuan Kong, Stefan Sosnowski, Qingchen Liu and Sandra Hirche

Abstract—Voronoi coverage control is a particular problem of importance in the area of multi-robot systems, which considers a network of multiple autonomous robots, tasked with optimally covering a large area. This is a common task for fleets of *fixed-wing* Unmanned Aerial Vehicles (UAVs), which are described in this work by a unicycle model with constant forward-speed constraints. We develop event-based control/communication algorithms to relax the resource requirements on wireless communication and control actuators, an important feature for battery-driven or otherwise energy-constrained systems. To overcome the drawback that the event-triggered algorithm requires continuous measurement of system states, we propose a self-triggered algorithm to estimate the next triggering time. Hardware experiments illustrate the theoretical results.

I. INTRODUCTION

Coverage control is one of the typical problems in the research field of multi-robot systems [1]. The coverage control problem considers the deployment of a network of mobile autonomous robots to optimally cover a specified area. Applications of coverage control involving multiple networked mobile robots (also referred to as a mobile sensor network) include search and rescue operations, surveillance, environmental monitoring, and exploration of hazardous/inaccessible regions. In many of these applications, an appropriate mobile robot of choice is a fixed-wing UAV. Fixed-wing UAVs can have great endurance in both flight time and range, able to stay in operation for many hours, and can be equipped with a range of different sensors.

The dynamic model of the robots must be considered when designing a distributed control algorithm to solve the coverage control problem. The pioneering work by Cortes et. al [1] considered single-integrator robot dynamics. In [2], multiple single-integrator robots covering the same partition of the region are considered. For robots with complex dynamics, e.g. fixed-wing UAVs, unicycle models are a more suitable choice to capture the robot dynamics [3], [4]. However, since unicycle models have complex nonholonomic dynamics, control algorithms developed for single-integrator

robot models cannot be readily applied. Our recent work [5] has studied the problem of optimal coverage control of unicycle robots and designed the corresponding controller.

Along with the idea in [1], follow-up research works [6]–[10] mainly focus on continuous-time controller design considering practical constraints from sensing devices or robot dynamics/kinematics. It is worth mentioning that the information required for control is the relative position between the robots, because of which wireless communication might be neglected. However, a multi-robot system benefits from information sharing within a wireless network in which each robot acts as a network node [11]. For example, 1) the external localization devices can provide more accurate position information; 2) the environment disturbance to the relative information sensing can be avoided. However, a wireless network is usually resource-limited in the sense of communication bandwidth. Thus how to design resource-aware control algorithms with a limited number of information exchanges is crucial for applying distributed control algorithms in practice. This motivated the development of distributed event-triggered control schemes, see, for example, [12]–[15]. In an event-triggered algorithm, the control and communication task is executed only when specific trigger conditions are met with respect to local system states. This approach significantly reduces the cost of communication resources since events are generated aperiodically and adaptively.

Event-based coverage control is still a relatively new research topic. In [16], a distributed event-triggered controller was designed for the Voronoi coverage problem. However, their considered robot dynamics are single-integrators and Zeno behavior cannot be excluded. In [17], a self-triggered mechanism was proposed to achieve both guaranteed and dual guaranteed Voronoi coverage control. The work in [18], [19] follows a similar streamline of [17]. However, their proposed algorithm can be regarded as an extension of discrete-time Lloyd descent, which means, continuous-time robot dynamics/kinematics cannot be involved.

In this work, we present distributed control algorithms for optimal coverage problems with unicycle-type robots, whose forward speed is constrained as a constant. To help achieve the static coverage objective, and for explicit use in the control algorithm to be designed, we define a virtual rotation center for each robot. The event-based scheduler computes an input mismatch according to the robots' states and triggers the control actuation and communication as long as a defined norm of the input mismatch exceeds a state-dependent threshold function. By using Lyapunov

* Corresponding author: qingchen.liu@tum.de.

Y. Zhou, S. Sosnowski, Q. Liu and S. Hirche are with the Chair of Information-Oriented Control, Technical University of Munich, Munich, Germany. {ge69fos, sosnowski, qingchen.liu, hirche}@tum.de.

L. Kong is with the Department of Engineering, University of Cambridge, Cambridgeshire, U.K. leo.kong6577@gmail.com.

The work of Q. Liu was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant 754462. The work of S. Hirche was supported by the German Research Foundation (DFG) within the Joint Sino-German research project Control and Optimization for Event-triggered Networked Autonomous Multi-agent Systems (COVEMAS).

analysis, we show that the virtual rotation center for each robot asymptotically converges to a centroidal Voronoi tessellation, thus achieving the optimal coverage objective. Our proposed algorithm can also guarantee Zeno-free triggering by showing a strictly positive lower bound of the inter-event time interval. Meanwhile, we also propose a self-triggered algorithm to overcome the drawback that the event-triggered algorithm requires continuous measurement of system states. The central idea is to estimate the next triggering time, by utilizing the certain bounds of the speed of the evolution states and the information of the threshold function. Hardware experiments have been conducted to illustrate the correctness of the proposed algorithms and demonstrate their performance.

The remainder of this paper is structured as follows: Section II introduces some preliminaries and provides a formal definition of the problem. Section III presents the event-triggered and self-triggered control algorithms with detailed theoretical analysis. The designed algorithms are verified by experiments in Section IV. Finally, we conclude this paper in Section V.

II. PRELIMINARIES AND PROBLEM STATEMENT

A. Notation

The set of complex numbers is denoted by \mathbb{C} . The imaginary unit is $i := \sqrt{-1}$. A complex number $z \in \mathbb{C}$ is denoted as $\mathbf{Re}(z) + i\mathbf{Im}(z)$, where $\mathbf{Re}(z)$ and $\mathbf{Im}(z)$ are its real and imaginary part, respectively. The complex conjugate of z is denoted by \bar{z} . For $z_1, z_2 \in \mathbb{C}$, the scalar product is defined by $\langle z_1, z_2 \rangle = \mathbf{Re}(\bar{z}_1 z_2)$. The norm of $z \in \mathbb{C}$ is defined as $\|z\| = \langle z, z \rangle^{\frac{1}{2}}$.

B. Locational Optimization

This section is based on the results of Cortes et. al [1]. Consider a group of n mobile robots, whose dynamics we define in the sequel, tasked with covering a convex polygon Q in \mathbb{R}^2 , and let q denote an arbitrary point in Q . A distribution density function is a map $\Phi : Q \rightarrow \mathbb{R}_+$ that represents a measure of information in Q . The location of the k^{th} robot moving in the space Q is denoted by $p_k, k \in \{1, \dots, n\}$, and $P = (p_1, \dots, p_n)$ captures the position of all n robots. Given a distribution density function $\Phi(q)$ that is known to all robots, we define a coverage performance function as

$$H(P) = \int_Q \min_{k \in \{1, \dots, n\}} \|q - p_k\|^2 \Phi(q) dq. \quad (1)$$

The quantity $\|q - p_k\|^2$ can be considered as a quantitative measure of how poorly a sensor positioned at p_k can sense the event of interest occurring at point q .

The minimizing operation inside the integral of the above performance function Eq. (1) induces a Voronoi partition $V(P) = (V_1, \dots, V_n)$ of the polygon Q . Formally, the k^{th} Voronoi partition, $k \in \{1, \dots, n\}$ is defined as follows:

$$V_k = \{q \in Q \mid \|q - p_k\| \leq \|q - p_j\|, \forall j \neq k\} \quad (2)$$

The set of regions V_1, \dots, V_n is called the Voronoi diagram for the generators p_1, \dots, p_n . Each Voronoi cell V_k is convex. When two Voronoi regions V_k and V_j are adjacent (i.e., they share an edge), robot k , with position p_k , is called a neighbor of robot j , with position p_j , (and vice versa). The set of indexes of the robots which are Voronoi neighbors of p_k is denoted by N_k . According to the definition of the Voronoi partition, we have $\min_{k \in \{1, \dots, n\}} \|q - p_k\|^2 = \|q - p_j\|^2$ for any q inside V_j . It follows that, $H(P)$ in (1) can be rewritten as

$$H(P) = \sum_{k=1}^n \int_{V_k} \|q - p_k\|^2 \Phi(q) dq \quad (3)$$

The following lemma establishes an important fact regarding the performance function (3).

Lemma 1 (Lemma 2.1, [1]): The gradient of $H(P)$ is given by

$$\frac{\partial H}{\partial p_k} = \int_{V_k} \frac{\partial}{\partial p_k} \|q - p_k\|^2 \Phi(q) dq \quad (4)$$

□

It follows immediately from Lemma 1 that

$$\frac{\partial H}{\partial p_k} = 2p_k \int_{V_k} \Phi(q) dq - 2 \int_{V_k} q \Phi(q) dq.$$

For all $k \in \{1, \dots, n\}$, define the (generalized) mass M_{V_k} , and centroid (or centre of mass) C_{V_k} of the Voronoi partition V_k as:

$$M_{V_k} = \int_{V_k} \Phi(q) dq, \quad C_{V_k} = \frac{1}{M_{V_k}} \int_{V_k} q \Phi(q) dq.$$

With these definitions, the gradient can now be expressed as

$$\frac{\partial H}{\partial p_k} = 2M_{V_k}(p_k - C_{V_k}) \quad (5)$$

Critical points of H are those in which every robot is at the centroid of its Voronoi cell. A Voronoi configuration corresponding to critical points of H is called a Centroidal Voronoi Tessellation. The objective of this paper is to design a distributed controller for each robot k . Then, collectively, the robots can move into positions that establish a Centroidal Voronoi Tessellation.

C. Unicycle Model with Constant Speed

We choose to model each robot as a unicycle. Unicycle dynamics are appropriate for capturing fixed-wing UAVs or ground-based vehicles. The unicycle model for the k^{th} robot, $k \in \{1, \dots, n\}$, is given as

$$\begin{aligned} \dot{x}_k(t) &= v_k \cos(\theta_k(t)) \\ \dot{y}_k(t) &= v_k \sin(\theta_k(t)) \\ \dot{\theta}_k(t) &= u_k(t) \end{aligned}$$

where $x_k(t) \in \mathbb{R}, y_k(t) \in \mathbb{R}$ are the coordinates of robot k in the real plane and $\theta_k(t)$ is the heading angle (i.e. the forward facing direction of the robot) at time t . The forward velocity $v_k \in \mathbb{R}_+$ is by definition strictly positive. In this paper, we assume that it is *fixed/constant/nonidentical*, and cannot be

designed as a control input. In practice, the cruising speed may be the most fuel-efficient speed for the robot (which is desirable for long periods of surveillance) or optimal against some other performance measure. The control input $u_k(t)$ is to be designed for steering the orientation of robot k . When there is no risk of confusion, we drop the argument t .

For analysis purposes, we express the position of robot k , (x_k, y_k) , in the complex plane using the complex notation $r_k = x_k + iy_k := \|r_k\|e^{i\theta_k} \in \mathbb{C}$. Then the above unicycle dynamics can be reformulated as

$$\begin{aligned}\dot{r}_k(t) &= v_k e^{i\theta_k(t)} \\ \dot{\theta}_k(t) &= u_k(t)\end{aligned}\quad (6)$$

In the complex plane, robot k 's instantaneous center of the circular orbit c_k can be given by

$$c_k(t) = r_k(t) + \frac{v_k}{\dot{\theta}_k(t)} i e^{i\theta_k(t)} \quad (7)$$

To help achieve the control objective, and for the explicit use in the control algorithm to be designed, we define for each robot k a virtual center z_k , as

$$z_k(t) = r_k(t) + \frac{v_k}{\omega_0} i e^{i\theta_k(t)}. \quad (8)$$

Here, ω_0 is a non-zero constant. Differentiating both sides of (8), and substituting in (6), the dynamics of the virtual center can be written as:

$$\dot{z}_k(t) = v_k e^{i\theta_k(t)} - \frac{v_k}{\omega_0} e^{i\theta_k(t)} u_k(t), \quad k = 1, \dots, n \quad (9)$$

D. Problem Statement

Given a convex polygon $Q \in \mathbb{C}$, suppose there are n unicycle robots with dynamics given by (6), for $k = 1, \dots, n$. Let v_k be the constant forward speed for robot k , with v_k not necessarily equal to v_j for $k \neq j$, $k, j \in \{1, \dots, n\}$, and z_k be the virtual center as defined in (8). Let Z be the stacked column vector of all virtual centers z_k , i.e. $Z = (z_1, \dots, z_n)$. The objective is to design the control input u_k to steer the virtual center z_k of each unicycle robot to a desired Voronoi centroid C_{V_k} so that the polygon Q can be covered optimally in the sense of minimizing the performance function

$$H_V(Z) = \sum_{k=1}^n \int_{V_k} \|q - z_k\|^2 \Phi(q) dq. \quad (10)$$

That is, we desire

$$\lim_{t \rightarrow \infty} z_k(t) = C_{V_k}(t) \quad \forall k = 1, \dots, n \quad (11)$$

Once the virtual center has arrived at the centroid, i.e. (11) has been achieved, each robot will orbit about the centroid with the fixed angular speed ω_0 . That is, $\lim_{t \rightarrow \infty} \dot{\theta}_k = \omega_0$ for all k , which when combined with the existing assumption that v_k is constant, implies robot k is executing a steady-state circular orbit. Since v_k and v_j are not necessarily equal for $k \neq j$ and $k, j \in \{1, \dots, n\}$, it follows that the robots' steady-state orbit radii are not necessarily equal.

Liu et. al [5] proposed a continuous controller:

$$u_k(t) = w_0 + \gamma w_0 \left\langle z_k(t) - C_{V_k}(t), v_k e^{i\theta_k(t)} \right\rangle \quad (12)$$

where $\gamma > 0$ is a positive control gain. Building on this previous approach, we aim to design event-triggered and self-triggered control algorithms to achieve optimal coverage, which should satisfy the following requirements:

- 1) The algorithm is distributed in the sense that only neighbor information is received for each robot to update the controller.
- 2) Zeno behavior should be completely excluded for each robot in implementing the control algorithm.
- 3) The self-triggered algorithm does not require any continuous measurement of the states.

The following lemma will later be used for stability analysis.

Lemma 2 (Lemma 3, [20]): Consider a continuously differentiable function $g : \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^{\geq 0}$. If there exist continuous functions $\gamma : \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^+$ and $\beta : \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^+$ satisfying $\dot{g}(t) \leq -\gamma(t)g(t) + \beta(t)$, then

$$g(t) \leq e^{-\int_0^t \gamma(s) ds} g(0) + \int_0^t e^{-\int_s^t \gamma(r) dr} \beta(s) ds \quad (13)$$

Furthermore, the following statements hold:

- If $\int_0^\infty \gamma(t) dt = \infty$ and $\lim_{t \rightarrow \infty} \frac{\beta(t)}{\gamma(t)} = 0$, then $\lim_{t \rightarrow \infty} g(t) = 0$.
- If $\int_0^\infty \gamma(t) dt = \infty$ and $\limsup_{t \rightarrow \infty} \frac{\beta(t)}{\gamma(t)} < \infty$, then $\{g(t)\}_{t \geq 0}$ is bounded.

Remark 1: Since fixed wing UAVs can be set to work in different heights, collision avoidance is not a strict requirement here and subject to future work.

III. PROBLEM ANALYSIS AND ALGORITHM FRAMEWORK

A. Event-Triggered Controller Design

Let the event time instants for robot k be denoted as $t_0^k = 0, t_1^k, \dots, t_l^k, \dots$, where $l \in \mathbb{Z}$ denoting the set of all nonnegative integers. In this paper, the execution rule is based on actuation errors rather than measurement errors. The control input mismatch for robot k is defined as

$$e_k(t) = u_k(t_l^k) - u_k(t), \quad t \in [t_l^k, t_{l+1}^k) \quad (14)$$

Every time an event is triggered, $e_k(t)$ is reset to be equal to zero. The event-based control input is described by

$$u_k(t_l^k) = w_0 + \gamma w_0 \left\langle z_k(t_l^k) - C_{V_k}(z_k(t_l^k), z_j(t_{l_j}^j)), v_k e^{i\theta_k(t_l^k)} \right\rangle \quad (15)$$

where $j \in \mathcal{N}_k, l^j = \operatorname{argmin}_{a \in \mathbb{N}} : t \geq t_a^j$. For $t \in [t_l^k, t_{l+1}^k)$, $t_{l_j}^j$ is the last event time of robot j . Each robot takes into account the last update value of each of its neighbors in its control law. According to the definition of the actuation error (14) we obtain

$$u_k(t_l^k) = u_k(t) + e_k(t) \quad (16)$$

We define an auxiliary variable

$$g_k(z_k(t)) = \left\langle z_k(t) - C_{V_k}(t), v_k e^{i\theta_k(t)} \right\rangle \quad (17)$$

By substituting (16) into the dynamic equation of each robot's virtual center (9), we obtain

$$\dot{z}_k(t) = -v_k e^{i\theta_k(t)} \left(\gamma g_k(z_k(t)) + \frac{e_k(t)}{\omega_0} \right) \quad (18)$$

We consider the following trigger function for each robot:

$$f_k = \underbrace{|e_k(t)|}_{\text{error}} - \underbrace{\sigma\gamma\omega_0 |g_k(z_k(t))|}_{\text{comparison threshold}} - \mu_k(t), \quad (19)$$

where $\gamma > 0$, $\omega_0 > 0$, $0 < \sigma < 1$, $\mu_k(t) = \gamma\omega_0 e^{-\alpha_k t}$ with $0 < \alpha_k < 1$. The error term is reset to zero whenever $f_k = 0$, i.e., the event is triggered. We now present the main result of the event-triggered control algorithm.

Theorem 1: Consider a group of n unicycle-type robots with constant, non-identical speeds modeled by (6) and driven by controller (15). The controller and the trigger function only require neighboring information to update. If each robot updates its input when the designed state-dependent trigger function $f_k = 0$, then their virtual centers converge asymptotically to the set of centroidal Voronoi tessellation (local minimum equilibrium for the coverage performance function) on Q and no robot exhibits Zeno behavior.

Proof: See Appendix. ■

B. Self-Triggered Controller Design

The intuitive idea for a self-triggered algorithm is to estimate the next updating time instant, by using the bounds of the actuation error $|e_k(t)|$ and the bounds of the comparison threshold. We first illustrate how to compute the bound of $|e_k(t)|$. Since the actuation error $|e_k(t)| = |u_k(t_l^k) - u_k(t)|$, the triggering rule can be rewritten as

$$|u_k(t_l^k) - u_k(t)| \leq \sigma\gamma\omega_0 |g_k(z_k(t))| + \mu_k(t) \quad (20)$$

Note that between the event-triggered control updates, the control input is held constant via the Zero-order hold technique. This observation motivates us to provide an estimation for $u_k(t)$, if $t \in [t_l^k, \min\{t_{l+1}^k, \min_{j \in N_k} t_{l''}^j\})$, using:

$$u_k(t) = \dot{u}_k(t_l^k)(t - t_l^k) + u_k(t_l^k) \quad (21)$$

and the time derivative of $u_k(t)$ is expressed as follows:

$$\begin{aligned} \dot{u}_k(t_l^k) &= \gamma\omega_0 \left\langle \dot{z}_k(t_l^k) - \dot{C}_{V_k}(t_l^k), v_k e^{i\theta_k(t_l^k)} \right\rangle \\ &\quad + \gamma\omega_0 \left\langle z_k(t_l^k) - C_{V_k}(t_l^k), i v_k u_k e^{i\theta_k(t_l^k)} \right\rangle \end{aligned}$$

where $l'' \triangleq \arg \min_{m \in \mathbb{N}: t_l^k \leq t_m^j} \{t_m^j - t_l^k\}$; t_m^j represents the update time instants of all the robot k 's neighbors after time instant t_l^k . $t_{l''}^j$ represents the latest update time instant for any robot k 's neighbors after time instant t_l^k ; $\min_{j \in N_k} t_{l''}^j$ represent the latest update time instant among the robot k 's neighbors, which is the nearest to the time instant t_l^k . Hence $\min\{t_{l+1}^k, \min_{j \in N_k} t_{l''}^j\}$ is the next time when the control input is updated. Thus, the triggering rule is equivalent to

$$\begin{aligned} |u_k(t_l^k) - u_k(t)| &= |\dot{u}_k(t_l^k)(t - t_l^k)| \\ &\leq \sigma\gamma\omega_0 |g_k(z_k(t))| + \mu_k(t) \end{aligned}$$

With the auxiliary variable defined in (17), we have

$$|\dot{u}_k(t_l^k)(t - t_l^k)| \leq \sigma |\dot{u}_k(t_l^k)(t - t_l^k) + u_k(t_l^k) - \omega_0| + \mu_k(t_l^k)$$

Let $\xi_l^k = \xi(t) = t - t_l^k$, $t \geq t_l^k$. Since $\xi_l^k \geq 0$, we have the triggering condition

$$\xi_l^k (1 - \sigma) |\dot{u}_k(t_l^k)| \leq \sigma |u_k(t_l^k) - \omega_0| + \mu_k(t_l^k) \quad (22)$$

For $|\dot{u}_k(t_l^k)| \neq 0$, we have the positive inter-execution time interval

$$\xi_l^k = \frac{\sigma |u_k(t_l^k) - \omega_0| + \mu_k(t_l^k)}{(1 - \sigma) |\dot{u}_k(t_l^k)|} \quad (23)$$

where $0 < \sigma < 1$. In the determination of the inter-execution time, no continuous measurement of the neighbor's states, and no continuous computation of the control input is needed. The explanation of the self-triggered rule for each robot can be summarized as: if no state of any neighbors is received in the time interval $(t_l^k, t_l^k + \xi_l^k)$ and there is a positive inter-execution time interval $\xi_l^k \geq 0$ such that the triggering condition $\xi_l^k |\dot{u}_k(t_l^k)| = |\sigma \dot{u}_k(t_l^k) \xi_l^k + \sigma u_k(t_l^k) - \sigma \omega_0| + \mu_k(t)$ is satisfied, then the next update time t_{l+1}^k takes place at most ξ_l^k time units after t_l^k . In other words, $t_{l+1}^k \leq t_l^k + \xi_l^k$. Otherwise, if the robot k 's control input is updated due to an update of one of its neighbors for $(t_l^k, t_l^k + \xi_l^k)$, then the triggering condition needs to be re-checked and the next triggering time is new determined.

We also provide a pseudo-algorithm for illustrative purpose, see Algorithm 1. Here, we let τ_p represent the temporary selection for the next triggering time instant t_{l+1}^k . At time instant t_{l+1}^k , robot k will measure all the neighbors' states, send its own states to all the neighbors and its control input will be updated. Applying the designed self-triggering

Algorithm 1: Determination of the next triggering time t_{l+1}^k

Data: the latest triggering time t_l^k , the latest triggering time among the neighbors after t_l^k : $\min_{j \in N_k} t_{l''}^j$

Result: determination of the next triggering time t_{l+1}^k

Initialization: $t_0 \leftarrow t_l^k$; $u_k(t_0)$; $\xi_0 \leftarrow \xi_l^k$; $\dot{u}_k(t_0)$;

while $\dot{u}_k(t_0) \neq 0$ **do**

$\tau_p \leftarrow t_p + \xi_p$;

if any neighbor of robot k updates its control

law in $t \in (t_p, \tau_p)$ **then**

$p \leftarrow p + 1$, $t_p \leftarrow \min_{j \in N_k} t_{l''}^j$;

Update $u_k(t_p)$, $\dot{u}_k(t_p)$, ξ_p ;

Continue;

else

$t_{l+1}^k \leftarrow \tau_p$;

Break;

end

end

return t_{l+1}^k

rule, we have the following theorem:

Theorem 2: Consider a group of n unicycle-type robots with constant, non-identical speeds modeled by (6) and driven by the controller (15). If each robot decides when

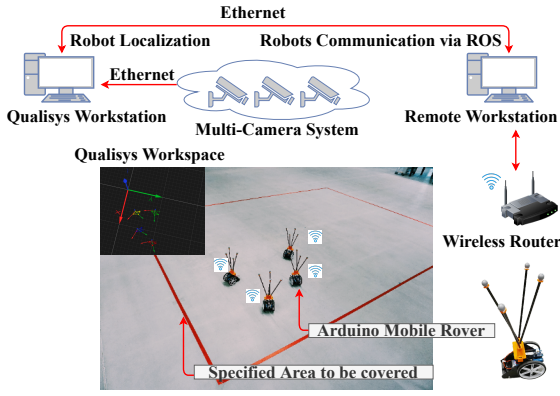


Fig. 1: Experimental setup

to update its control input according to Algorithm 1, then their virtual centers converge asymptotically to the set of centroidal Voronoi tessellation on Q and no robot exhibits Zeno behavior.

Proof: 1) *Stability analysis:* This part is similar to the proof of the event-based triggering theorem. The virtual centers of all unicycles will asymptotically converge to the set of centroidal Voronoi tessellations on Q .

2) *Absence of Zeno behavior:* When $\dot{u}_k(t_l^k) \neq 0$, there always exists a positive inter-execution time ξ_k , then we can conclude that there is no Zeno behavior in the system. When $\dot{u}_k(t_l^k) = 0$, it indicates that the control input has reached the desired value ω_0 and the centroidal Voronoi tessellation is achieved. ■

IV. EXPERIMENTS

In this section, the results of hardware experiments are provided to illustrate the performance of the proposed event-triggered and self-triggered algorithms. A video is available in our supplementary materials.

A. Experimental setup

The structure of our experimental setup is illustrated in Fig. 1. The system comprises a Qualisys multi-camera system, a remote workstation and four Arduino mobile rovers. The Qualisys multi-camera system provides high precision position and orientation information for the ground robots with up to 300Hz refresh rate. The remote workstation is a Lenovo ThinkPad laptop (Intel i5-6200U CPU and 8GB RAM), running Ubuntu 16.04 and robot operating system (ROS) kinetic. The mobile robots used in our experiment are Arduino mobile rovers with Wi-Fi access, which is a differentially driven robot¹. The remote workstation and the Arduino mobile rovers are communicating in a Wi-Fi network, with an ASUS RT-AC1750 router.

Our control framework is two-layer. The lower layer runs on the Arduino mobile rovers, in which the program interprets the input commands (forward speed and angular speed) to the respective rotational speeds of the two wheels and uses

¹The complete description of the Arduino mobile rovers is referred to Arduino Engineering Kit.

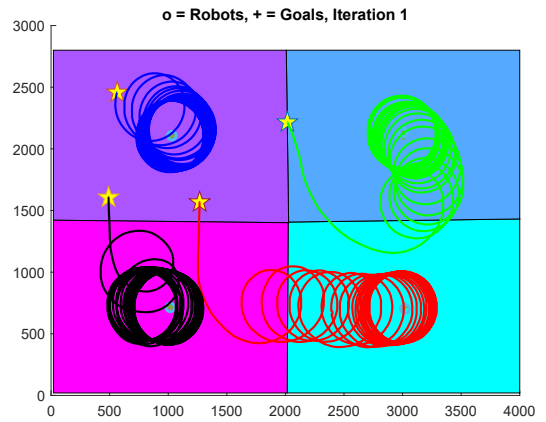


Fig. 2: Coverage performance using event-triggered controller. The coverage area is in the size of $4.0m \times 2.8m$. The yellow stars denote the initial positions of the robots. The solid lines with different colors represent the trajectories of each robot. All robots move in a circular orbit concerning their virtual center.

PID controllers to track them. The upper layer runs on the remote workstation, which consists of four ROS nodes. Each ROS node communicates with its neighbors to locally compute the Voronoi partitions (according to [21]) and the event-triggered and self-triggered control input (15). Then the control commands are transmitted to the Arduino mobile rovers via UDP protocol.

We provide two separate hardware experiments to demonstrate the performances of both event-triggered and self-triggered algorithms applied in a four-unicycle group. In both experiments, the coverage area Q is rectangular in the size of $4.0m \times 2.8m$. The density function $\Phi(q)$ is assumed to be 1 in the experiments. The angular frequency ω_0 defined in (8) is selected as 0.536rad/s for all robots. The robots' forward speeds are set as $v_k = 0.16\text{m/s}$ for all k . The selected values of the forward speed and angular speed meet the hardware constraints of the Arduino mobile rovers. The initial positions and initial orientations of the robots are manually chosen such that the initial virtual centers locate inside the coverage region.

B. Experimental results

The experimental results of the event-triggered controller is shown in Fig. 2, from which one can see that the optimal coverage objective has been successfully achieved. All virtual centers move asymptotically to the set of Centroidal Voronoi Tessellation. All robots move in a circular orbit concerning their virtual center. The trajectories of different robots are represented by solid lines with different colors. We note that the depicted trajectories are the raw data from Qualisys. The evolution of the event-triggered error and the comparison threshold in (19) is shown in Fig. 3. When the value of the error reaches the threshold, the error is reset to zero and an event is triggered. The control input is updated according to Theorem 1. The red line represents the evolution

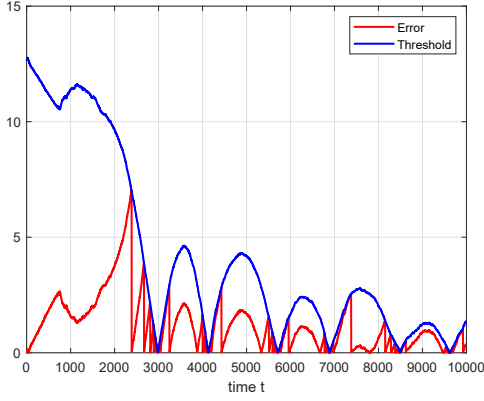


Fig. 3: Triggering behavior during the control process (time: ms). Error (red) vs. comparison threshold (blue).

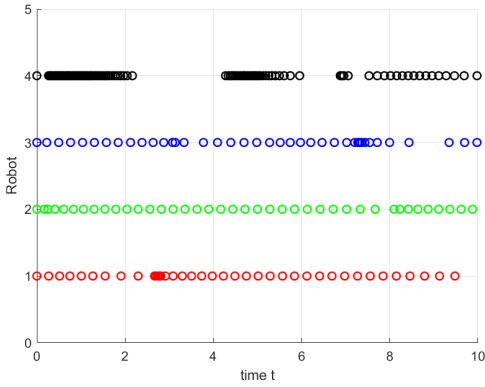


Fig. 4: Triggering time instants for the self-triggered algorithm. The red, green, blue and black dots denote the triggering times for robot 1, robot 2, robot 3 and robot 4, respectively. For illustration purpose, we select the time period between 0s to 10s with the triggering time instants for the four robots.

of the error. It stays below the designed threshold which is shown by the blue line in Fig. 3.

The second experiment illustrates the performance of the self-triggered algorithm. Due to the page limit, we omit the figure about the trajectories of the robots, since it is quite similar to the event-triggered case. The self-triggered control law can guarantee that the inter-event time intervals are lower bounded by a strictly positive constant. Thus, the Zeno behavior is excluded, which can be observed from Fig. 4.

From both experiments, it is observed that the optimal coverage objective can be achieved. The asynchronous feature of both the event-triggered and self-triggered algorithms are verified.

V. CONCLUSION

In this work, we develop event-/self-triggered algorithms for coverage control problems with a group of unicycle-type robots to facilitate the efficient usage of the network resource.

We firstly propose an event-based control algorithm, which generates events and control updates when the actuation error diverges to a designed threshold. Zeno behavior can be completely excluded by showing that strictly positive inter-event time intervals exist. Then we propose a self-triggered control algorithm to relax the strong assumption that continuous measurement is required for each robot, in which the core idea is to provide an estimation for the next update time instant based on the state and control value at the latest update time. Hardware experiments have been conducted to verify the correctness and performance of our proposed algorithms.

APPENDIX

Proof: The proof is divided into two parts. In the first part, the stability of the system (18) will be analyzed. In the second part, we will show Zeno behavior is excluded. 1) *Stability analysis:* By using Lem.1 and Eq.(5), the time derivative of the performance function (10) along the trajectory of system (18), is

$$\begin{aligned} \dot{H}_V(Z) &= \sum_{k=1}^n 2M_{V_k} \langle z_k(t) - C_{V_k}(t), \dot{z}_k(t) \rangle \\ &= - \sum_{k=1}^n 2M_{V_k} \left(\gamma g_k^2(z_k(t)) + \frac{e_k(t)}{\omega_0} g_k(z_k(t)) \right) \end{aligned}$$

Because γ , ω_0 and M_{V_k} are strictly positive, if $|e_k(t)| \leq \sigma \gamma \omega_0 |g_k(z_k(t))|$, $0 < \sigma < 1$, then $\dot{H}_V(Z) \leq 0$ for all $t \geq 0$. Because $\mu_k(t) > 0$, it is evident that $|e_k(t)| \leq \sigma \gamma \omega_0 |g_k(z_k(t))| + \mu_k(t)$. Each robot updates its input when $f_k = 0$, then we have

$$\begin{aligned} \dot{H}_V(Z) &\leq - \sum_{k=1}^n 2\gamma M_{V_k} g_k^2(z_k(t)) \\ &\quad + \sum_{k=1}^n \frac{2}{\omega_0} M_{V_k} |e_k(t)| |g_k(z_k(t))| \\ &\leq \sum_{k=1}^n (\sigma - 1) 2\gamma M_{V_k} g_k^2(z_k(t)) \\ &\quad + \sum_{k=1}^n 2\gamma M_{V_k} e^{-\alpha_k t} O |\cos \psi| \end{aligned}$$

where ψ is the angle between the vectors $z_k - C_{V_k}$ and $v_k e^{i\theta_k}$, and $O = \|z_k - C_{V_k}\| \|v_k e^{i\theta_k}\|$.

Note that we can rewrite the performance function (10) as follows (see [1] for more details):

$$H_V(Z) = \sum_{k=1}^n J_{V_k, C_{V_k}} + \sum_{k=1}^n M_{V_k} \|z_k - C_{V_k}\|^2 \quad (24)$$

where $J_{V_k, C_{V_k}} \in \mathbb{R}^+$ is the polar moment of inertia of the Voronoi cell V_k about its centroid C_{V_k} . Define

$$H_{V,1}(Z) = \sum_{k=1}^n J_{V_k, C_{V_k}}, H_{V,2}(Z) = \sum_{k=1}^n M_{V_k} \|z_k - C_{V_k}\|^2 \quad (25)$$

Then we have

$$\begin{aligned}
\dot{H}_V(Z) &= \dot{H}_{V,2}(Z) \\
&\leq -\sum_{k=1}^n (1-\sigma)2\gamma M_{V_k} \|z_k - C_{V_k}\|^2 \|v_k e^{i\theta_k}\|^2 \cos^2 \psi \\
&\quad + \sum_{k=1}^n 2\gamma e^{-\alpha_k t} M_{V_k} \|z_k - C_{V_k}\| \|v_k e^{i\theta_k}\| |\cos \psi|
\end{aligned} \tag{26}$$

Define ϱ as the set of Z, θ for which $\dot{H}_V(Z) = 0$. We consider the following two cases: 1) $\cos \psi = 0, z_k \neq C_{V_k}$; 2) $\cos \psi \neq 0$.

Case 1: $\cos \psi = 0, z_k \neq C_{V_k}$. Then we have

$$\dot{H}_V(Z) \leq 0 \tag{27}$$

$\cos \psi = 0$ means $z_k - C_{V_k} \perp v_k e^{i\theta_k}$. Then from Eq.(12) it is concluded that $u_k = \omega_0$, which implies $\dot{z}_k = 0 \forall k \in \{1, \dots, n\}$. The vector $v_k e^{i\theta_k}$ changes its direction with angular frequency ω_0 . Therefore $v_k e^{i\theta_k}$ is non-constant but $z_k - C_{V_k}$ is constant. It is to verify that the set $\{(Z, \theta) : z_k - C_{V_k} \perp v_k e^{i\theta_k}, z_k - C_{V_k} \neq 0, \forall k \in \{1, \dots, n\}\}$ is not the equilibrium of the system.

Case 2: $\cos \psi \neq 0$.

Let

$$\begin{aligned}
h_k(t) &= M_{V_k} \|z_k - C_{V_k}\|^2 \\
\kappa_k(t) &= (1-\sigma)2\gamma \|v_k e^{i\theta_k}\|^2 \cos^2 \psi \\
\beta_k(t) &= 2\gamma e^{-\alpha_k t} M_{V_k} \|z_k - C_{V_k}\| \|v_k e^{i\theta_k}\| |\cos \psi|
\end{aligned}$$

Then we can rewrite the Eq.(26) as follows:

$$\sum_{k=1}^n \dot{h}_k(x) \leq \sum_{k=1}^n (-\kappa_k(t)h_k(x) + \beta_k(t)) \tag{28}$$

$$\int_0^\infty \kappa_k(t) dt = \int_0^\infty (1-\sigma)2\gamma \|v_k e^{i\theta_k}\|^2 \cos^2 \psi dt = \infty \tag{29}$$

$$\lim_{t \rightarrow \infty} \frac{\beta_k(t)}{\kappa_k(t)} = 0 \tag{30}$$

By applying Lem.2, we can conclude that

$$\lim_{t \rightarrow \infty} h_k(t) = \lim_{t \rightarrow \infty} M_{V_k} \|z_k - C_{V_k}\|^2 = 0$$

Since M_{V_k} is the mass of the Voronoi cell k , which is positive, then we have

$$\lim_{t \rightarrow \infty} z_k(t) = C_{V_k}(t)$$

2) *Absence of Zeno behavior:*

Definition 1: A solution is said to have a nonvanishing dwell time if there exists $\tau > 0$ such that

$$\inf_l (t_{l+1}^k - t_l^k) \geq \tau \tag{31}$$

In other words, the existence of a lower bound τ for inter-execution time intervals is to be proved. Thus it can guarantee

the system does not exhibit Zeno behavior. For any $l \geq 0$, and any $k \in \{1, \dots, n\}$, consider the time interval $t \in [t_l^k, t_{l+1}^k)$. From the definition of $e_k(t)$ in Eq.(14) and the fact that in the time interval $u_k(t)$ is a constant, we observe the time derivative of $|e_k(t)|$ satisfies

$$\frac{d}{dt} |e_k(t)| \leq |\dot{u}_k| \tag{32}$$

From Eq.(12), we have

$$|\dot{u}_k| = \gamma w_0 \left| \left\langle \dot{z}_k - \dot{C}_{V_k}, v_k e^{i\theta_k} \right\rangle + \left\langle z_k - C_{V_k}, i v_k u_k e^{i\theta_k} \right\rangle \right|$$

then we have

$$\begin{aligned}
|\dot{u}_k| &\leq \gamma w_0 \left\| \dot{z}_k - \dot{C}_{V_k} \right\| \|v_k e^{i\theta_k}\| \\
&\quad + \gamma w_0 \|z_k - C_{V_k}\| \|v_k e^{i\theta_k}\| |u_k|
\end{aligned}$$

Define the following auxiliary variables:

$$A(t) = \left\| \dot{z}_k - \dot{C}_{V_k} \right\| \|v_k e^{i\theta_k}\|$$

$$B(t) = \|z_k - C_{V_k}\| \|v_k e^{i\theta_k}\| |u_k|$$

The system dynamics are assumed to satisfy the following properties:

- P(1) There exists a scalar constant $d > 0$ such that $\|z_k - C_{V_k}\| < d$.
- P(2) There exists positive constants $v_{k,nom}$ for each robot such that $v_k \leq v_{k,nom}$.

From Eq.(12), we obtain

$$|u_k| \leq w_0 + \gamma w_0 \|z_k - C_{V_k}\| \|v_k e^{i\theta_k}\| \tag{33}$$

Then from the assumed properties P(1) and P(2), we have that $|u_k|$ is upper bounded by a positive value. It is straightforward to see that the term $B(t)$ is bounded.

The time derivative of the centroid of the Voronoi cell is given by

$$\dot{C}_{V_k} = \frac{\partial C_{V_k}}{\partial z_k} \dot{z}_k + \sum_{j \in N_k} \frac{\partial C_{V_k}}{\partial z_j} \dot{z}_j \tag{34}$$

$$\dot{z}_k - \dot{C}_{V_k} = \dot{z}_k - \frac{\partial C_{V_k}}{\partial z_k} \dot{z}_k - \sum_{j \in N_k} \frac{\partial C_{V_k}}{\partial z_j} \dot{z}_j \tag{35}$$

From Eq.(18) we have

$$\|\dot{z}_k(t)\| \leq \left\| \gamma v_k e^{i\theta_k(t)} \right\| O + \left\| \frac{v_k}{\omega_0} e^{i\theta_k(t)} \right\| |e_k(t)| \tag{36}$$

According to the trigger condition we have $|e_k(t)| \leq |\gamma \omega_0 g_k(z_k(t))| + \mu_k(t)$. Together with the assumed properties we have that $\|\dot{z}_k(t)\|$ is upper bounded. Analysis similar to $\|\dot{z}_j(t)\|$.

We note that the work [22] provided the analytic solutions of the partial derivative $\frac{\partial C_{V_k}}{\partial z_k}$ and $\frac{\partial C_{V_k}}{\partial z_j}$. We refer the readers to [22] for more details.

In this paper, we consider the convex polygon in the planar case and that the area the robots need to cover is bounded. So the area of each Voronoi cell is bounded. It is evident that the integration area, in other words, the boundary of the Voronoi cell is bounded. Thus, $\left\| \frac{\partial C_{V_k}}{\partial z_k} \right\|$ and $\left\| \frac{\partial C_{V_k}}{\partial z_j} \right\|$ are bounded. Then we have

$$\left\| \dot{z}_k - \dot{C}_{V_k} \right\| \leq \left\| 1 - \frac{\partial C_{V_k}}{\partial z_k} \right\| \left\| \dot{z}_k \right\| + \sum_{j \in N_k} \left\| \frac{\partial C_{V_k}}{\partial z_j} \right\| \left\| \dot{z}_j \right\| \quad (37)$$

The term $\left\| \dot{z}_k - \dot{C}_{V_k} \right\|$ is bounded. Then we obtain that $B(t)$ is bounded. From the above conclusions, it is straightforward to conclude that $|\dot{u}_k|$ is bounded. Define a positive constant B_e , which represents the upper bound of $|\dot{u}_k|$. Then, we obtain

$$\frac{d}{dt} |e_k(t)| \leq B_e$$

It follows that

$$|e_k(t)| \leq \int_{t_l^k}^t B_e dt = (t - t_l^k) B_e \quad (38)$$

for $t \in [t_l^k, t_{l+1}^k)$ and for any l . Recall the trigger function Eq.(19), and the control input mismatch $e_k(t)$ is reset to zero at t_l^k . It follows that the next event time t_{l+1}^k is determined by the changing rates of $e_k(t)$ and the threshold $\sigma\gamma\omega_0 |g_k(z_k(t))| + \mu_k(t)$. It is evident that

$$|e_k(t)| = \sigma\gamma\omega_0 |g_k(z_k(t))| + \mu_k(t) \quad (39)$$

holds at the next trigger time t_{l+1}^k . In the stability analysis part we conclude that $\lim_{t \rightarrow \infty} z_k(t) = C_{V_k}(t)$. However, it is important to point out that in the evolution of the system, $z_k(t) - C_{V_k}(t) = 0$ may also hold at t_{l+1}^k while the optimal coverage is not achieved as z_k can be nonzero at t_{l+1}^k . We consider the triggering at t_{l+1}^k in the following two cases:

- Case 1: If $\left\| z_k(t_{l+1}^k) - C_{V_k}(t_{l+1}^k) \right\| \neq 0$, the equality $|e_k(t_{l+1}^k)| = \sigma\gamma\omega_0 |g_k(z_k(t_{l+1}^k))| + \mu_k(t_{l+1}^k)$ is satisfied.
- Case 2: If $\left\| z_k(t_{l+1}^k) - C_{V_k}(t_{l+1}^k) \right\| = 0$, the equality $|e_k(t_{l+1}^k)| = \mu_k(t_{l+1}^k)$ is satisfied.

It is evident to see that $\left\| z_k(t_{l+1}^k) - C_{V_k}(t_{l+1}^k) \right\| > 0$ for any $\left\| z_k(t_{l+1}^k) - C_{V_k}(t_{l+1}^k) \right\| \neq 0$. Compare the above two cases, we can conclude that it takes longer for the quantity $|e_k(t_{l+1}^k)|$ to increase to be equal to the quantity $|\gamma\omega_0 g_k(z_k(t_{l+1}^k))| + \mu_k(t_{l+1}^k)$ than to increase to be equal to the quantity $\mu_k(t_{l+1}^k)$. In other words, $\tau_{Case1} > \tau_{Case2}$.

According to Eq.(38),

$$B_e \tau_{Case2} \geq |e_k(t)| = \mu_k(t) = \gamma\omega_0 e^{-\alpha_k(t_l^k + \tau_{Case2})} \quad (40)$$

where $\gamma\omega_0 e^{-\alpha_k(t_l^k + \tau_{Case2})} > 0$, B_e is a positive constant. We can conclude that the inter-event time interval τ_{Case2} is strictly positive. Since there is a positive lower bound on the inter-event time intervals, there are no accumulation points in the event sequences, so Zeno behavior is excluded for all robots. ■

- [1] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
- [2] B. Jiang, Z. Sun, and B. D. O. Anderson, "Higher order voronoi based mobile coverage control," in *2015 American Control Conference (ACC)*, pp. 1457–1462, IEEE, 2015.
- [3] G. S. Seyboth, J. Wu, J. Qin, C. Yu, and F. Allgöwer, "Collective circular motion of unicycle type vehicles with nonidentical constant velocities," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 2, pp. 167–176, 2014.
- [4] Z. Sun, G. S. Seyboth, and B. D. O. Anderson, "Collective control of multiple unicycle agents with non-identical constant speeds: Tracking control and performance limitation," in *2015 IEEE Conference on Control Applications (CCA)*, pp. 1361–1366, IEEE, 2015.
- [5] Q. Liu, M. Ye, Z. Sun, J. Qin, and C. Yu, "Coverage control of unicycle agents under constant speed constraints," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 2471–2476, 2017.
- [6] M. Schwager, J. McLurkin, and D. Rus, "Distributed coverage control with sensory feedback for networked robots," in *Robotics: Science and Systems*, pp. 49–56, 2006.
- [7] A. Gusrialdi, S. Hirche, T. Hatanaka, and M. Fujita, "Voronoi based coverage control with anisotropic sensors," in *2008 American Control Conference*, pp. 736–741, IEEE, 2008.
- [8] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, and D. Rus, "Voronoi coverage of non-convex environments with a group of networked robots," in *2010 IEEE International Conference on Robotics and Automation*, pp. 4982–4989, IEEE, 2010.
- [9] A. Gusrialdi and C. Yu, "Exploiting the use of information to improve coverage performance of robotic sensor networks," *IET Control Theory & Applications*, vol. 8, no. 13, pp. 1270–1283, 2014.
- [10] M. Boldrer, D. Fontanelli, and L. Palopoli, "Coverage control and distributed consensus-based estimation for mobile sensing networks in complex environments," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 7838–7843, IEEE, 2019.
- [11] A. Mavrommati, E. Tzorakoleftherakis, I. Abraham, and T. D. Murphy, "Real-time area coverage and target localization using receding-horizon ergodic exploration," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 62–80, 2017.
- [12] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1291–1297, 2011.
- [13] B. Wei, F. Xiao, and M.-Z. Dai, "Edge event-triggered control for multi-agent systems under directed communication topologies," *International Journal of Control*, vol. 91, no. 4, pp. 887–896, 2018.
- [14] Z. Sun, N. Huang, B. D. O. Anderson, and Z. Duan, "A new distributed zero-free event-triggered algorithm for multi-agent consensus," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 3444–3449, IEEE, 2016.
- [15] Q. Liu, M. Ye, J. Qin, and C. Yu, "Event-triggered algorithms for leader-follower consensus of networked euler-lagrange agents," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 7, pp. 1435–1447, 2017.
- [16] N. Hayashi, Y. Muranishi, and S. Takai, "Distributed event-triggered control for voronoi coverage," in *2015 International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, pp. 1–4, IEEE, 2015.
- [17] C. Nowzari and J. Cortés, "Self-triggered coordination of robotic networks for optimal deployment," *Automatica*, vol. 48, no. 6, pp. 1077–1087, 2012.
- [18] D. Tabatabai, M. Ajina, and C. Nowzari, "Self-triggered distributed k-order coverage control," *arXiv preprint arXiv:1903.04726*, 2019.
- [19] M. Ajina, D. Tabatabai, and C. Nowzari, "Asynchronous distributed event-triggered coordination for multiagent coverage control," *IEEE Transactions on Cybernetics*, 2020.
- [20] Y. Liu, Y. Lou, B. D. O. Anderson, and G. Shi, "Network flows that solve least squares for linear equations," *Automatica*, vol. 120, p. 109108, 2020.
- [21] C. N. Hadjicostis and M. Cao, "Distributed algorithms for voronoi diagrams and applications in ad-hoc networks," tech. rep., Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 2003.
- [22] S. G. Lee, Y. Diaz-Mercado, and M. Egerstedt, "Multirobot control using time-varying density functions," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 489–493, 2015.