IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

Time-inverted Kuramoto Model Meets Lissajous Curves: Multi-Robot Persistent Monitoring and Target Detection

Manuel Boldrer[1], Lorenzo Lyons[1], Luigi Palopoli[2], Daniele Fontanelli[2], Laura Ferranti[1]

***Abstract*—This work proposes a distributed strategy to achieve both persistent monitoring and target detection in a rectangular and obstacle-free environment. Each robot has to repeatedly follow a smooth trajectory and avoid collisions with other robots. To achieve this goal, we rely on the time-inverted Kuramoto dynamics and the use of Lissajous curves. We analyze the resiliency of the system to perturbations or temporary failures, and we validate our approach through both simulations and experiments on real robotic platforms. In the latter, we adopt Model Predictive Contouring Control as a low level controller to minimize the tracking error while accounting for the robots' dynamical constraints and the control inputs saturation. The results obtained in the experiments are in accordance with the simulations.**

***Index Terms*—Multi-Robots, Distributed Control, Kuramoto model, Persistent Monitoring, Target Detection**

I. Introduction

AUTONOMOUS robots, such as ground robots or unmanned aerial vehicles, are increasingly used to perform sensitive, but repetitive tasks, such as helping human operators monitor high-risk or contaminated areas. These tasks require multiple robots to collaborate to efficiently cover a target region. To achieve this goal, in the last decade distributed control strategies for robots' coordination have emerged as an effective and reliable solution. The use of multiple robots offers many advantages over the use of a single robot, especially when the task requires optimizing the spatial organization of the agents (like in coverage). Two recognized benefits of this solution are the performance improvement coming from the natural exploitation of parallelism and sensors distribution, and the increased robustness due to redundancy and/or diversity in the system. This paper focuses on the design of a resilient coordination strategy for a team of autonomous robots performing persistent monitoring of a known mission space and target detection.

Manuscript received: July, 25, 2022; Revised: September, 22, 2022; Accepted: November, 3, 2022.

This paper was recommended for publication by Editor M. Ani Hsieh upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the NWO-TTW Veni project HARMONIA (no. 18165). (*Corresponding author: Manuel Boldrer*).

[1]M. Boldrer, L. Lyons, L. Ferranti are with the Dept. of Cognitive Robotics, Delft University of Technology, Mekelweg 2, 2628 CD, Delft, Netherlands, {m.boldrer, l.lyons, l.ferranti}@tudelft.nl. [2]D. Fontanelli and L. Palopoli are with the Dept. of Industrial Engineering (DII) and with the Dept. of Information Engineering and Computer Science (DISI), respectively, University of Trento, Via Sommarive 5, Trento, Italy, {daniele.fontanelli,luigi.palopoli}@unitn.it.

Digital Object Identifier (DOI): see top of this page.

A. Related work

Coverage problems, like the one considered in this work, are widely addressed in the literature. They can be classified in two broad categories: static coverage and dynamic coverage. The aim of static coverage is to deploy the robots in a final configuration that minimize a coverage cost function [1], [2], [3]. When the number of robots or the sensing range is not sufficient to statically cover the mission space, dynamic coverage comes into play. Different papers in the literature address this problem, both considering unknown environments [4], [5], well-defined points of interest to visit or a desired path to monitor periodically [6], [7].

A problem that is strictly related to coverage is target detection. The aim is to coordinate the robots' motion in order to maximize the probability to detect an event or a target in the mission space. This is not an issue for static coverage since the environment is completely and constantly covered after that the equilibrium is reached. On the other hand, target detection is not guaranteed for dynamic coverage. In a recent survey [8], the authors classify different approaches on the basis of the guarantees that the algorithm can provide, that is, worst-case guarantees, probabilistic and none. In the capture problem, the agents have to sweep a given area providing worst-case guarantees. In other words, if the target is in the area of interest it must be detected in finite-time. The two main approaches in the literature rely on graph clearing [9] or geometry-based strategies [10]. A distributed approach for the capture problem is proposed in [11]. When resources are limited (e.g., in the number of agents or in the available time), it is reasonable to rely on probabilistic search, where probabilities of target detection can be estimated [12], [13].

Target detection and persistent monitoring become significantly more complex in presence of additional requirements on the trajectories followed by the robot. To the best of our knowledge, Borkar et al. [14] is the only method that guarantees target detection, persistent monitoring for each agent and collision avoidance, without relying on sharp turns and additional path planners to enable repetition. The main drawback of this solution is that, since they provide an open loop control strategy, it is highly sensitive to uncertainties, disturbances or failures. More recently, Borkar et al. [15] adopted a distributed and robust tracking system to follow reference points that are still computed in open loop and in a centralized fashion. Yao et al. [16] propose a coordination strategy based on guiding vector fields and consensus, however

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

they have to specify the desired distance between robots, hence they have to rely on the number of agents in the network, to achieve the desired robot deployment.

B. Paper contribution and organization

In our previous works [17], [18], we introduced the idea of using time-inverted Kuramoto dynamics for static and dynamic coverage applications. By taking inspiration from [14], we exploit the properties of the Lissajous curves to achieve dynamic coverage and guarantee, at the same time, target detection in a finite-time.

Our contribution is fourfold. First, we propose a novel distributed strategy for persistent monitoring and target detection in rectangular and obstacle-free spaces. In particular, we exploit the properties of the time-inverted Kuramoto dynamics [17], combined with the properties of the Lissajous curves [14] to achieve the goal. Second, we provide sufficient conditions for resiliency to perturbation in the case of tracking errors due to disturbances or agents' failures. Third, we provide two possible solutions to avoid collisions between agents during the transient. Finally, to test the effectiveness of our method on a real robotic platform, we introduce a tailored low-level controller based on Model Predictive Contouring Control (MPCC) to minimize the tracking errors, while accounting for the robots' dynamical constraints and the control inputs saturation.

The paper is organized as follows. Sec. II describes the problem setup and provides all the required background material. Sec. III shows the properties that emerge by combining the time-inverted Kuramoto dynamics with the Lissajous curve. Sec. IV provides the analysis of the resiliency to perturbation and collision avoidance between agents. Sec. V motivates the use of the time-inverted Kuramoto model in comparison with state-of-the-art methods. Sec. VI shows the details of the low-level MPCC designed for the experiments on car-like robots. Sec. VII describes the experimental setup and the results. Sec. VIII concludes the paper.

II. Problem Setup and Preliminary Results

This paper addresses the persistent monitoring problem joined with target detection. This problem can be formulated as follows:

Problem 1. Given a convex and rectangular space of dimensions $[-A, A] \times [-B, B]$, and N homogeneous robotic agents with an omnidirectional sensing range of radius $r_{s,i}$, synthesize a distributed control law that satisfies the following requirements:

1) all the rectangular search space must be swept up continuously and by all the robots' sensors;
2) if a rogue element is introduced into the mission space, its presence is detected in finite-time;
3) avoid collisions between robots;
4) the robots must follow smooth trajectories, i.e., without discontinuities or sharp turns.

A. Graph topology constraints

Given N mobile robots $\mathcal{V} = \{1, \dots, N\}$, we assume that the i–th robot can communicate with the j–th robot if $(i, j) \in \mathcal{E}$.

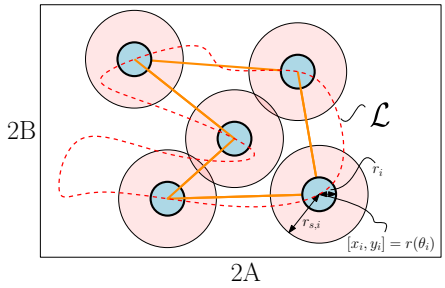

Figure 1: We show the problem setup. The space of interest is a rectangular space $[-A, A] \times [-B, B]$, the $N = 5$ robots are depicted as blue circles and their motion is constrained on the closed dashed curve $\mathcal{L}$. The i–th robot has a sensing radius $r_{s,i}$ and an encumbrance radius r_i. The edges of the ring topology are depicted as orange links.

Let us denote with $\mathcal{V}(i)$ the i–th entry of $\mathcal{V}$. In this paper we assume to have a constant and undirected ring-like topology, that is, $\mathcal{E} = \{(\mathcal{V}(i), j) : j = \mathcal{V}(i+1), \forall i = 1, \dots, N-1\} \cup \{(\mathcal{V}(i), j) : j = \mathcal{V}(i-1), \forall i = 2, \dots, N\}\} \cup \{(\mathcal{V}(1), \mathcal{V}(N))\} \cup \{(\mathcal{V}(N), \mathcal{V}(1))\}$, where we allow to have an arbitrary order of the $\mathcal{V}$ entries. We indicate this set of edges with $\mathcal{R}$, and we indicate with $\mathcal{R}_i = \{j : (i, j) \in \mathcal{R}\}$. In practice each agent exchanges information with exactly 2 robots, and the overall communication graph topology $\mathcal{G}(\mathcal{V}, \mathcal{R})$ results to be connected. This constraint is necessary to obtain the desired equilibrium configurations (for more details see [17], [18]).

B. Motion constraints

The robots are constrained to move along a closed path $\mathcal{L} \subset \mathbb{R}^2$, which can be represented in parametric form as $r(\gamma) : \mathbb{R} \to \mathbb{R}^2$, where $\gamma \in [0, 2\pi]$. The position of the i–th robot is $r(\theta_i)$, where $\theta_i \in \mathbb{R}$ identifies the i–th robot's state. Figure 1 depicts the problem setup. Given this setup we want to coordinate the robots along a closed path so that the previously mentioned requirements are satisfied. To solve this problem, we first impose the time-inverted Kuramoto dynamics to the robots' states $\theta = [\theta_1, \dots, \theta_N]^\top$. Second, on the basis of the mission space $[-A, A] \times [-B, B]$, the i-th agent's sensing range $r_{s,i}$, and the i-th agent encumbrance r_i, we properly select the number of agents N and the Lissajous curve $\mathcal{L}$ on which we constrain the motion of each agent.

Remark 1. *The assumption of having a robot that perfectly moves along a desired path is quite optimistic in an experimental setup, especially due to dynamic constraints and control inputs saturation. As discussed in our experimental section, the problem can be effectively solved using state–of–the–art path tracking algorithms.*

C. Time-inverted Kuramoto dynamics

The time-inverted Kuramoto model was introduced in [17] for *static* coverage and in [18] for persistent monitoring of predefined closed paths of interest. The time-inverted Kuramoto dynamics can be expressed as:

$$\dot{\theta}_i = \omega - \sum_{j \in \mathcal{R}_i} \sin(\theta_j - \theta_i), \quad \forall i = 1, \dots, N, \tag{1}$$

where ω identifies the natural frequency, that is, the desired velocity of the robots at the equilibrium. The main results, on which we will rely on, read as follows:

Theorem 1 (Convergence to the equilibrium [18])**.** *The dynamical system* (1) *converges to the equilibrium point*

$$\theta^{\star(p)} = [\theta_0 + 2z_1\pi, \theta_0 + 2z_2\pi + \frac{2\pi p}{N}, \dots \\ \dots, \theta_0 + 2z_N\pi + \frac{2\pi p(N-1)}{N}]^\top, \tag{2}$$

where $\theta_0 \in \mathbb{R}$ can be any real number, $p \in \mathcal{P}_k = \{y \in \mathbb{Z} \mid y \in (N/4 + kN, 3N/4 + kN), \forall k \in \mathbb{Z}\}$, and $z_i \in \mathbb{Z}$ for all $i = 1 \dots N$.

This important theorem tells us that a subset of the equilibrium configurations in (2) is stable. In particular, the configuration is stable when the value of p in (2) is an integer value and lives in the specified intervals.

Let us recall also the following:

Definition 1 (Cluster)**.** *A group $\mathcal{Y}$ of robots forms a cluster if for every $i, j \in \mathcal{Y}$ there exists a $k \in \mathbb{Z}$ such that $\theta_i - \theta_j = 2k\pi$.*

Lemma 1 (κ-clustered coverage [18])**.** *Given $N > 2$ and the dynamics* (1)*, the number of agents that cluster together at the stable equilibrium point* (2) *is given by the greatest common divisor between N and p, denoted by $\kappa = gcd(N, p)$.*

Lemma 2 (Equilibrium manipulation [18])**.** *Given an odd number of agents $N > 2$, being in the equilibrium configuration* (2) *with $p = \frac{N-1}{2}$, by considering the modified equilibrium configurations $\tilde{\theta}^{\star(\frac{N-1}{2})} = \alpha\theta^{\star(\frac{N-1}{2})}$, we can reconstruct all the equilibria $\theta^{\star(\bar{p})}$ with $\bar{p} \in \mathbb{N}$, by tuning the parameter α.*

III. Time-inverted Kuramoto meets Lissajous

In the preliminaries, we discussed the time-inverted Kuramoto dynamics; a strategy that can be used to achieve κ-clustered coverage on an arbitrary closed path $\mathcal{L}$. In this section we show that, by directly imposing the time-inverted Kuramoto dynamics on the Lissajous curves, interesting properties arise. The parametric form of a non-degenerative Lissajous curve [19] can be expressed as follows

$$x(\gamma) = A\cos(a\gamma),\ y(\gamma) = B\sin(b\gamma) \tag{3}$$

where γ is the parameter, $a, b \in \mathbb{Z}$ and a is an odd number. By exploiting the results in [14], it can be show that the coordinates at the equilibrium (2) i.e.,

$$(x_i(\gamma), y_i(\gamma)) = \left(A\cos(\theta_i^{\star(p)} - a\gamma), B\sin(\theta_i^{\star(p)} + b\gamma)\right), \tag{4}$$

assuming $a + b = N$, satisfy the equation

$$\frac{y_i(\gamma)^2}{B^2} + \frac{x_i(\gamma)^2}{A^2} - \frac{2x_i(\gamma)y_i(\gamma)\sin((a+b)\gamma)}{AB} = \cos^2((a+b)\gamma), \tag{5}$$

which represents a family of ellipses centred in the origin.

First of all let us derive conditions under which the 1-clustered coverage formation is a stable configuration.

Theorem 2 (1-clustered coverage)**.** *Let us consider $N > 2$ and $N \neq \{4, 6\}$, by imposing* (1) *there always exists a stable configuration that corresponds to the* 1*-clustered coverage configuration.*

Proof. By exploiting the result of Lemma 1 and the definition of $\mathcal{P}_k$ given in Theorem 1, it is sufficient to prove that there exists at least one value for $p \in \mathcal{P}_0$ that is co-prime with respect to N. If N is a prime number e.g., $N = \{3, 5\}$, since $\mathcal{P}_0$ is a nonempty set, the condition is satisfied. For $N > 6$ and $N \neq 8$, we have that $\text{card}(\mathcal{P}_0) \geq 4$, hence we have one of the two following cases: if N is odd, $\frac{N-1}{2} \in \mathcal{P}_0$, and $\gcd(N, \frac{N-1}{2}) = 1$; if N is even we have $\{\frac{N}{2} - 1, \frac{N}{2} - 2\} \in \mathcal{P}_0$ and, due to the Euclidean algorithm [20], N is co-prime with respect to at least one of the two values. To conclude the proof, we notice that for $N = 8$, $\frac{N}{2} - 1 \in \mathcal{P}_0$ and, again, $\gcd(N, \frac{N}{2} - 1) = 1$. □

By relying on the theory behind the time-inverted Kuramoto dynamics and the results given by [14], we are able to provide conditions for collision avoidance, complete coverage and target detection by assuming the robots to be in the equilibrium configuration (4).

1) Collision avoidance: Let us assume $N > 2$ and $N \neq \{4, 6\}$, by assuming the i–th agent to have a circular footprint of radius r_i, the condition $r_i < r_{i,\max}, \forall i = 1 \dots N$, ensures collision avoidance at every 1-clustered equilibrium configuration (2) (notice that a stable 1-clustered equilibrium configuration exists in view of Theorem 2), where

$$r_{i,\max} = \sin\left(\frac{\pi}{N}\right)\frac{AB}{\sqrt{A^2a^2 + B^2b^2}}. \tag{6}$$

2) Complete coverage: Ability to cover all the mission space with the sensor range $r_{s,i}$ in a finite-time. Let us refer to as nodes on the Lissajous curve the points at the intersections and at the boundaries of the rectangular space of interest. It is shown in [14] that the maximum x coordinate difference between two adjacent node positions is given by $x_{\max} = A\sin(\frac{\pi}{2b})$, while for the y coordinate $y_{\max} = B\sin(\frac{\pi}{2a})$. Then the complete coverage condition can be written the following condition on the sensing radii,

$$r_{s,i} > \max\left\{B\sin\left(\frac{\pi}{2a}\right), A\sin\left(\frac{\pi}{2b}\right)\right\},\ \ \forall i = 1 \dots N. \tag{7}$$

3) Target detection: The ability to detect a target element in the mission space in a finite-time. This condition has to imply complete coverage (the opposite does not apply). The sensing radius that ensures overlapping sensing range for each κ-clustered equilibrium configuration (2) and for any γ value reads as

$$r_{s,i} \geq \sin\left(\frac{\pi}{N/\kappa}\right)\sqrt{A^2 + B^2},\ \ \forall i = 1 \dots N. \tag{8}$$

Notice that the maximum time required to detect a rogue element in the space is given by $T_{\max} = \frac{2\pi/\omega}{N/\kappa}$.

Figure 2 shows the time evolution of the time-inverted Kuramoto dynamics (1) on the Lissajous curve (3). We select the parameters in order to satisfy collision avoidance and target detection (as a consequence also complete coverage). Figure 3 presents the results related to the target detection. We simulate the presence of $n = 10^4$ moving elements in the scenario depicted in Figure 2, where $N = 15$ and $\omega = 0.05$ (rad/s). The maximum detection time $T_{\max} = 8.38$ (s) remains satisfied; all the elements are detected in $T \leq 4.5$ (s).

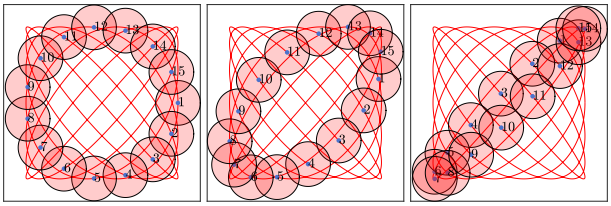

Figure 2: Evolution in time of the time-inverted Kuramoto dynamics (1) on the Lissajous curve (3), where $N = a + b$ with a odd. We select a sensing radius $r_{s,i}$ that satisfies the target detection condition (8), and r_i satisfying the collision avoidance condition (6).

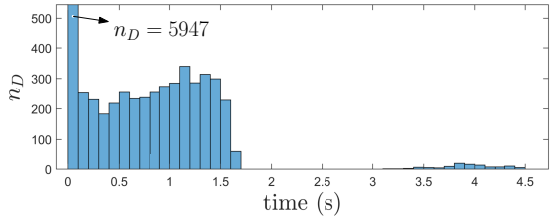

Figure 3: Number of new detections during the evolution of the time-inverted Kuramoto model in Figure 2. We added in the environment $n = 10^4$ moving elements. By setting $\omega = 0.05$ (rad/s), all the element are detected within 4.5 (s).

The authors in [14] provide these conditions by assuming to place the agents in the 1-clustered equilibrium configuration (2) and by imposing a pre-defined velocity profile $V_i(\gamma)$. However, this approach is effective in an ideal scenario, where there are not disturbances or unexpected events. In the following section we show how the proposed algorithm allows us to deal with these important practical issues.

IV. Resiliency of the System

The resiliency of a networked system measures its capability to recover promptly from unexpected events [21]. In particular, we analyze how the system reacts to perturbations on the agents' positions, and to agents' failures.

Theorem 3 (Resiliency to perturbation)**.** *Given the robots in the equilibrium point $\theta^{\star(p)}$, with $p \in \mathcal{P}_k$, by perturbing the i–th agent by $\delta\theta_i$, if*

$$\cos(\theta_i^{\star(p)} - \theta_j^{\star(p)} + \delta\theta_i) < 0, \forall j \in \mathcal{R}_i, \tag{9}$$

then the equilibrium point $\theta^{\star(p)}$ remains unaltered.

Proof. Let us derive the Jacobian matrix $J^{(p)}$ of (1) associated with the equilibrium $\theta^{\star(p)}$. The entries of the matrix $J^{(p)}$ are as follows

$$J^{(p)} = \begin{cases} a_{ii} = \sum_{j\in\mathcal{R}_i} \cos\left(\theta_j^{\star(p)} - \theta_i^{\star(p)}\right) \\ a_{hi} = \begin{cases} -\cos\left(\theta_h^{\star(p)} - \theta_i^{\star(p)}\right), & h \in \mathcal{R}_i \\ 0, & h \notin \mathcal{R}_i \end{cases} \end{cases}$$

By applying the Gershgorin theorem [22], it can be noticed that the condition $\cos(\theta_j^{\star(p)} - \theta_i^{\star(p)} + \delta\theta_i) < 0, \forall j \in \mathcal{R}_i$, ensures that all the eigenvalues associated with the linearized system are nonpositive, hence the system does not leave the equilibrium configuration $\theta^{\star(p)}$. □

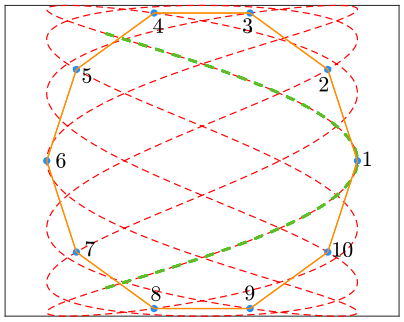

Figure 4: Final equilibrium configuration for the system (1) with $N = 10$. The Lissajous curve is represented by a red dotted line and its parameters are $a = 7, b = 3, A = B = 5$. The agents' positions on the curve are depicted as blue dots and the admissible perturbation that maintain the same equilibrium configuration (it satisfies (9)) is depicted as a green dashed line.

This is an important sufficient condition that ensures the system to recover the initial equilibrium configuration, despite an unexpected perturbation in the agents' positions along the curve. By exploiting the results of Theorem 3 we can extend the findings to resiliency to agents' failures:

Theorem 4 (Resiliency to agents' failures)**.** *Suppose that at the stable equilibrium $\theta^{\star(p)}$ one or more robots stop their motion, If at every instant of time, $\cos(\theta_j - \theta_i) < 0, \forall j \in \mathcal{R}_i, \forall i = 1, .., N$, then, if the stopped robots resume the dynamics* (1)*, the system still converges to the initial configuration $\theta^{\star(p)}$.*

Proof. In accordance with the proof of Theorem 3, if the condition $\cos(\theta_j - \theta_i) < 0$ is always satisfied then the system cannot jump in another equilibrium configuration. □

Remark 2. *Notice that proof of Theorem 4 relies on the linear approximation of the Kuramoto model. Nevertheless, simulations and experiments support the condition found.*

In Figure 4 we report the equilibrium configuration for the case of $N = 10$, $a = 7$, $b = 3$, $A = 5$, $B = 5$. The agents are represented by blue dots, the Lissajous curve is the red dashed line, the edges of the graph topology are specified by the solid orange lines, while the green dashed line denotes the set of admissible positions of, e.g., robot 1 such that the condition (9) is satisfied. As a consequence, by perturbing the robot position inside that range, the final equilibrium configuration is recovered, as formally stated in Theorem 3. In section V we provide a simulation to validate the results on resiliency to agents' failures.

A. Preserving collision avoidance

The condition in Equation (6) ensures collision avoidance between the agents when they are in a 1-clustered equilibrium configuration. When the system is not at the equilibrium, i.e., has been perturbed, even if (9) is satisfied, we cannot ensure collision avoidance. In order to address this problem, we observe two solutions: i) implement a lower level controller for collision avoidance or ii) design non-intersecting Lissajous curves (it applies only for three dimensional spaces). The former solution can be done by implementing a lower level

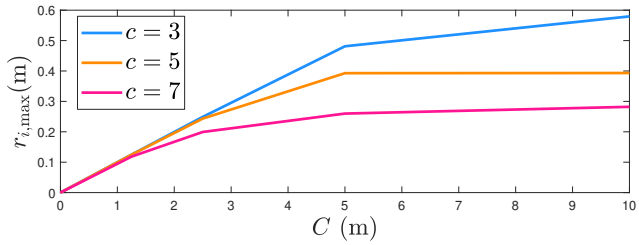

Figure 5: Variation of the maximum agents' occupancy radius allowed to avoid collisions, with the parameters c and C in (10).

collision avoidance strategy e.g., [23], [24], [25]. This solution is able to guarantee safety (also in the case of generic κ-clustered configurations) at the price of deviating from the Lissajous path or from the desired velocity. However, it may lead to loss of target detection guarantees and/or complete coverage, and it asks for updated convergence proofs.

On the other hand, the design of non-intersecting Lissajous curves is not feasible in a two dimensional space (e.g., for Unmanned Ground Vehicles). Nevertheless, it can be applied in a three dimensional space (e.g., for Unmanned Aerial Vehicles). In this case we can consider the three dimensional Lissajous curve

$$\begin{cases} x(\gamma) &= A\cos(a\gamma) \\ y(\gamma) &= B\sin(b\gamma) \\ z(\gamma) &= C\sin(c\gamma). \end{cases} \tag{10}$$

Notice that, if a, b, $c \in \mathbb{N}$ are co-prime numbers, then we obtain a Lissajous knot [26], which by definition avoids self-intersections. This way, by adding a space dimension, we can tune the parameters C and c in (10) to avoid collisions even in the transient generated after a perturbation. In Figure 5, we show the effects of the parameters c and C, on the maximum radius $r_{i,\max}$ that still guarantees collision avoidance.

Notice that we do not ensure collision avoidance for random initial configurations (overtaking maneuvers can take place between agents on the Lissajous curve). Nevertheless, at the equilibrium, if a perturbation (which does not induce to overtaking maneuvers) occurs and condition (3) is satisfied, we have both collision avoidance guarantees and Lissajous path following.

V. Comparison with state of the art

The choice behind the time-inverted Kuramoto model, rather than other strategies, is motivated by multiple factors. First, by relying on Lemma 2 we can manipulate the final equilibrium configuration according to specific requirements (e.g., we can rely on κ-clustered configurations to increase the redundancy in the system in order to account for sensing failures). To the best of our knowledge this is a unique feature that other existing approaches do not provide. Second, our approach is resilient to disturbances or tracking errors, according to Theorem 3. Third, our method can be easily adapted to any parametric curve, such as non-degenerate Lissajous curves.

Beside these motivations, the main advantage of our approach over the strategies proposed in [15], [14] is that we do not

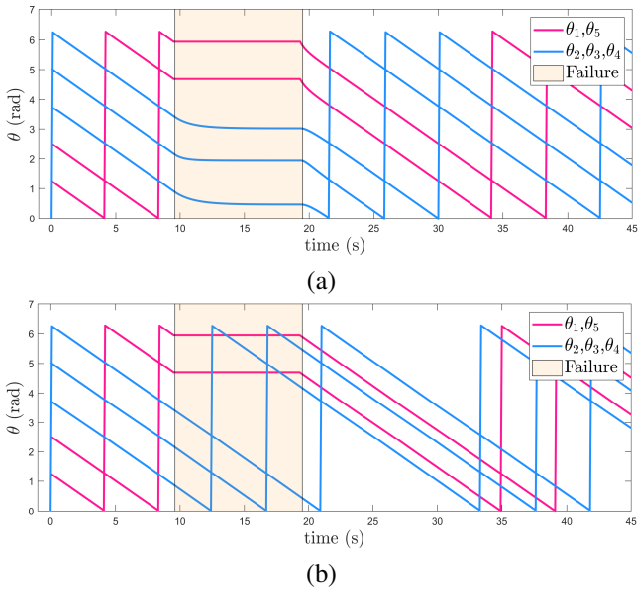

Figure 6: Evolution in time of the states θ. Two robots fail and stop working at time $t_1 = 9.6$ (s). The robots resume their functions at time $t_2 = 19.5$ (s). In (a) the time-inverted Kuramoto model recovers quickly to the equilibrium configuration, in (b) the solution proposed in [15] does not.

directly assume a reference point computed in open loop and in a centralized manner. Our method allows each agent to locally compute a reference point on the basis of the agent's distance with respect to its neighbours. It enhances flexibility, robustness and the coordination between agents. Moreover, the number of agents N is only needed to properly compute the Lissajous curve. Then, the coordination strategy remains agnostic to the number of agents, that is, the knowledge of N is not needed for coordination purposes since the information is encoded in the Lissajous curve.

In the following, we provide a comparison between our approach and the one proposed in [15], where we enforce a finite-time failure of two agents. Figure 6 shows the results of the comparison. With our approach, the robotic network reacts to the agents' failure and once the agents recover, since conditions in Theorem 4 are met, the system goes back to the initial equilibrium configuration. With the approach proposed in [15], the robotic network does not react to the failures and, after the agents recover their operation, the desynchronization in the reference point generation leads to the loss of target detection and collision avoidance guarantees. Figure 6 depicts the evolution in time of the states θ. The system starts in an equilibrium configuration. At time t_1 two agents remain stuck, while at t_2 they resume functioning. Figure 6-(a) shows the results obtained by relying on the time-inverted Kuramoto model. It can be noticed that after t_2 the system returns to the initial equilibrium configuration. In contrast, with the baseline that relies on an open loop reference in Figure 6-(b), the system does not re-adapt autonomously and the desired configuration (2) is not recovered.

Yao et al. [16], which is based on linear consensus [27] and guiding vector fields, is another noteworthy approach to coordinate the robots motion. The main advantage of our

strategy with respect to the one proposed in this recent work is due to the fact that the time-inverted Kuramoto dynamics, as we already mentioned, can rely on multiple equilibrium configurations, and it does not have to know a priori the number of robots in the system. On the other hand, a coordination algorithm based on linear consensus has to specify the desired distance between neighbours; in this way it is not possible to change the configuration without changing the control scheme. Moreover, to define the desired distance between agents, it has to rely on the number of robots in the scene. Finally, to build a proper guiding vector field, assumptions on the curve to follow have to be made.

VI. Low level MPCC design

The time-inverted Kuramoto model considers robots as points moving along the Lissajous curve, that is, it considers a one dimensional problem. A lower control routine is needed to enforce path following and velocity tracking. We rely on Model Predictive Control (MPC). MPC allows our method to consider the full dynamic model of the robots. Furthermore, thanks to its constraint-handling abilities, the MPC controller is able to account for actuator saturation. This is particularly valuable since the turns featured in the Lissajous path can exceed the steering angle limits of the robots. For this work, we chose a particular formulation of MPC referred to as Model Predictive *Contouring* Control (MPCC) [28]. Compared to traditional MPC for tracking, MPCC does not require an explicitly time-dependent reference trajectory. These features make MPCC particularly well-suited to act as a low level controller for our purposes. In [28] however, the controller aims at tracking a given longitudinal speed, while in our case we require to track a certain $\dot{\theta}$. We have therefore modified the relative term in the cost function. The optimization problem solved by the MPCC, for the i-th agent, is the following:

$$\min_{\delta_i,\tau_i} \sum_{k=0}^{M} J_{\dot{\theta}_i}^k + q_1 {\varepsilon_{i,\text{lag}}^k}^2 + q_2 {\varepsilon_{i,\text{lat}}^k}^2 + q_3 {\delta_i^k}^2 + q_4 {\tau_i^k}^2 \quad (11a)$$

$$\text{s.t. } X_i^k = f(X_i^{k-1}, \delta_i^{k-1}, \tau_i^{k-1}), \text{ for } k = 1, ..., M \quad (11b)$$

$$\delta_{\min} \leq \delta_i \leq \delta_{\max}, \;\; \tau_{\min} \leq \tau_i \leq \tau_{\max}, \quad (11c)$$

where k is the time index, M is the number of stages in the optimization problem, $J_{\dot{\theta}_i}$ is the $\dot{\theta}$ tracking term (15) that substitutes the velocity tracking term in [28], q_1, q_2, q_3, q_4 are weights normalized with respect to the weight of $J_{\dot{\theta}_i}$, $\varepsilon_{i,\text{lag}}$ is the so called lag error and is a specific term of the MPCC formulation (refer to [28] for more details), $\varepsilon_{i,\text{lat}}$ is the lateral distance to the path, δ_i and τ_i are respectively the steering angle and the throttle, $\delta_{\min}$, $\delta_{\max}$, $\tau_{\min}$, $\tau_{\max}$ are the minimum and maximum values of the respective inputs. We decided not to include explicit collision avoidance constraints, since we want to focus on the results provided by the time-inverted Kuramoto model. X_i is the predicted state of the i–th robot and f represents the robot dynamics. The state X_i is defined as:

$$X_i = \begin{bmatrix} x_i & y_i & \eta_i & v_i & s_i \end{bmatrix}^\top,$$

where x_i, y_i, η_i, v_i, s_i are the position of the rear axle, orientation, longitudinal velocity, and the natural parameter of the closest point to the i–th robot on the reference path, respectively. The dynamics of each robot are considered as a kinematic bicycle model [29]. The longitudinal acceleration is $\dot{v}_i = -cv_i + \alpha\tau_i - \beta$, where c is the damping coefficient, and α and β are the motor force coefficients (we evaluated them experimentally through step response tests for the target application). The dynamic constraint (11b) also includes the dynamics of the natural parameter s_i, necessary to evaluate $J_{\dot{\theta}_i}(k)$. In summary, the dynamic constraint (11b) (in continuous time) is given by:

$$\dot{X}_i = \begin{bmatrix} v_i \cos\eta_i \\ v_i \sin\eta_i \\ \frac{v_i \tan(\delta_i)}{l} \\ -cv_i + \alpha\tau_i - \beta \\ v_i \end{bmatrix}, \quad (12)$$

where l is the length between the front and rear axle of each robot.

Concerning the term $J_{\dot{\theta}_i}$, the time-inverted Kuramoto model (1) provides a desired $\dot{\theta}_i^D$, that can be tracked by the MPCC. This, however, is expressed as a feedback law depending on the neighboring agents' positions. Thus, to implement this policy directly, the MPCC should be designed in a fully distributed way. This would imply a large amount of network traffic and would ultimately hinder the real-time applicability of the overall control strategy. For this reason we opted to keep the desired $\dot{\theta}_i^D$ constant during the whole prediction horizon. In this way the θ_j values need to be exchanged only once per sampling time. Our choice is further legitimized by noticing that at equilibrium the $\dot{\theta}_i^D$ remains constant. The cost term $J_{\dot{\theta}_i}$ is defined as

$$J_{\dot{\theta}_i} = (\dot{\theta}_i^D - \dot{\theta}_i)^2. \quad (13)$$

The state of the vehicle provides s and $\dot{s}$ (since $\dot{s} = v$ as described in (12)), thus we must rewrite (13) as a function of these quantities. The expression of the arc-length s for a certain value of γ is as follows:

$$s(\gamma(t)) = \int_{\gamma_0}^{\gamma(t)} \|r'(\tilde{\gamma})\|_2 d\tilde{\gamma}, \quad (14)$$

where we set $\gamma_0 = 0$ as defined in II-B, $r'(\tilde{\gamma})$ is the partial derivative of the Lissajous curve with respect to $\tilde{\gamma}$, and the $\|\cdot\|_2$ operator is the Euclidean norm. By computing the time derivative of (14) we obtain $\dot{s} = \dot{\gamma}\|r'(\gamma)\|_2$. Note that the MPCC formulation can only predict future values of s, thus the term $\|r'(\gamma)\|_2$ needs to be expressed as a function of the latter, that is, an expression of $\gamma(s)$ is needed. This however is not trivial since it requires the inversion of (14) that is not possible to formalize analytically. For this reason we have evaluated $\gamma(s)$ numerically, and used a suitable polynomial approximation of it during run-time. The velocity tracking term in the MPCC cost function (13) can be re-written as:

$$J_{\dot{\theta}_i}(s, \dot{s}) = (\dot{\theta}_i^D - \dot{s}/\|r'(\gamma(s))\|_2)^2. \quad (15)$$

VII. Experimental results

Due to hardware availability reasons, we are going to consider 3 scaled-down car-like robots and 2 simulated agents. To

control the robots a dedicated ROS network has been set up and we rely on an OptiTrack motion capture system to receive the positions of the robots in real-time. The ring topology $\mathcal{R}$ in (1) adopted in the experiments reads as follows, $\mathcal{R}_1 = \{2,4\}, \mathcal{R}_2 = \{1,5\}, \mathcal{R}_3 = \{4,5\}, \mathcal{R}_4 = \{1,3\}, \mathcal{R}_5 = \{2,3\}$. The radius of encumbrance of each vehicle is $r_i = 0.125$ (m). We consider the following Lissajous curve for the experiments:

$$\mathcal{L} : \begin{cases} x(\gamma) & = 1.8\cos(3\gamma) \\ y(\gamma) & = 0.8\sin(2\gamma). \end{cases} \tag{16}$$

We satisfy collision avoidance at the equilibrium, that is, $r_i < r_{i,\max}$ (6). At run-time each agent computes its desired $\dot{\theta}_i^D$ according to (1). Note that to perform this step the i–th agent requires the θ_j values, with $j \in \mathcal{R}_i$. Once $\dot{\theta}_i^D$ is computed the value is passed to the MPCC, which solves the optimization problem (11), (12) and computes the control inputs τ_i, δ_i. The video of the experiments can be found at [30].

The parameters used for the experiments are the following: $q_1 = 0.01$, $q_2 = 0.025$, $q_3 = 0.01$, $q_4 = 0.01$, $\delta_{\min} = -\frac{\pi}{9}$ (rad), $\delta_{\max} = \frac{\pi}{9}$ (rad), $\tau_{\min} = 0.1$, $\tau_{\max} = 0.25$, $M = 15$ in (11), $c = 0.94$, $\alpha = 36.8$, $\beta = 0.94$, $l = 0.175$ (m) in (12), $\omega = 0.3$ (rad/s) in (1).

The first experiment shows the importance of the feedback term in (1). We run the experiment neglecting it and imposing the dynamics $\dot{\theta}_i = \omega$, $\forall i = 1, \dots, N$. The obtained results can be found at [30]. The initial configuration of the system is picked around an equilibrium point (2). After few seconds the system diverges from the equilibrium due to modelling errors and disturbances, until a physical collision occurs.

The second experiment shows the results obtained by imposing the time-inverted Kuramoto dynamics (1). The initial configuration of the system (Figure 7-(a)) is set around an equilibrium point (2). Thanks to the feedback action, the system is able to remain in that equilibrium configuration, see Figures 7-(b,c,d). At the time instant of (d), to experimentally show the resiliency of our algorithm, we simulate a temporary failure. In particular, we modify the dynamics of the simulated agents in $\dot{\theta}_a = 0$, where $a = \{1,5\}$. The networked system converges to another equilibrium configuration (Figure 7-(f)), where all the agents are stationary. After few seconds we reset $\dot{\theta}_a$ as in (1) and the system, after a brief transient, comes back to its previous equilibrium configuration (Figures 7-(f,g,h)). Figure 8 reports the quantitative data for the same experiment. The obtained results are in strong agreement with the results obtained in the simulations, see Figure 6-(a).

In the last experiment we impose the time-inverted Kuramoto dynamics (1). We run the experiment for a sufficient amount of time in order to investigate the effects of disturbances and model mismatch on target detection, collision avoidance and complete coverage. In Figure 9 we focus on the agent 3; we report the evolution in time of the Euclidean distance between agents 3 and 4 (d_{34}), and between agents 3 and 5 (d_{35}), and the lateral error from the Lissajous path ($d_{3l} \equiv \varepsilon_{3,\text{lat}}$). The maximum value for the tracking error d_{3l} is the increase in the sensing range (7), sufficient to guarantee complete coverage. The maximum distance between two neighboring agents ($d_{34}(t^\star)$) is twice the sensing radius sufficient to ensure target detection. In this experiment the minimum sensing radius to ensure target detection (8) in the ideal case has to be multiplied by a factor $\eta_1 = 1.2429$, in order to compensate the effects of the lateral error d_{3l}. At the same time instant $t^\star = 20.8$ (s) the large amount of lateral error d_{3l} leads to a collision between agent 3 and the simulated agent 5, since $d_{35}(t^\star) = 0.192$ (m).

Regarding the risk of collision an effective solution is to include explicit constraints in the MPCC's optimization problem (11) (see section IV-A). Considering the tracking error we have observed how it has negative effects on the properties of the system, yet at the price of an enlarged sensing radius and a smaller encumbrance radius the system is able to recover the desired features. Notice that a proper choice of the Lissajous path can also be beneficial. The tracking error d_{3l} may be induced by several factors: (i) the limited values for the steering angle $\delta_{\min}, \delta_{\max}$, with respect to the minimum radius of curvature in the Lissajous curve; (ii) the discrepancies between the model and the real dynamics of the robots; and (iii) the assumption in the MPCC formulation of keeping constant the $\dot{\theta}_i^D$ values for the whole prediction horizon. To reduce the tracking error, we should focus on these aspects.

VIII. Conclusions

We proposed a distributed control strategy to coordinate a team of robots in order to achieve persistent monitoring and target detection in a known environment. We combined the time-inverted Kuramoto dynamics and the Lissajous curves to achieve our aim, in combination with a low-level model predictive controller tailored to the proposed application. We characterize the resiliency of the algorithm subject to perturbations and we provide conditions to preserve safety. We tested our approach on real robotic platforms. As a future work direction, we plan to investigate its scalability with an increasing number of robots and apply the method to different robotics platforms, such as UAVs, to test the use of three dimensional Lissajous paths.

References

[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] 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)*. IEEE, 2019, pp. 7838–7843.

[3] M. Zhong and C. G. Cassandras, "Distributed coverage control and data collection with mobile sensor networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2445–2455, 2011.

[4] C. Franco, D. M. Stipanović, G. López-Nicolás, C. Sagüés, and S. Llorente, "Persistent coverage control for a team of agents with collision avoidance," *Euro. Journal of Control*, vol. 22, pp. 30–45, 2015.

[5] S. C. Pinto, S. B. Andersson, J. M. Hendrickx, and C. G. Cassandras, "Optimal periodic multi-agent persistent monitoring of a finite set of targets with uncertain states," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 5207–5212.

[6] F. Pasqualetti, J. W. Durham, and F. Bullo, "Cooperative patrolling via weighted tours: Performance analysis and distributed algorithms," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1181–1188, 2012.

[7] S. L. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 410–426, 2011.

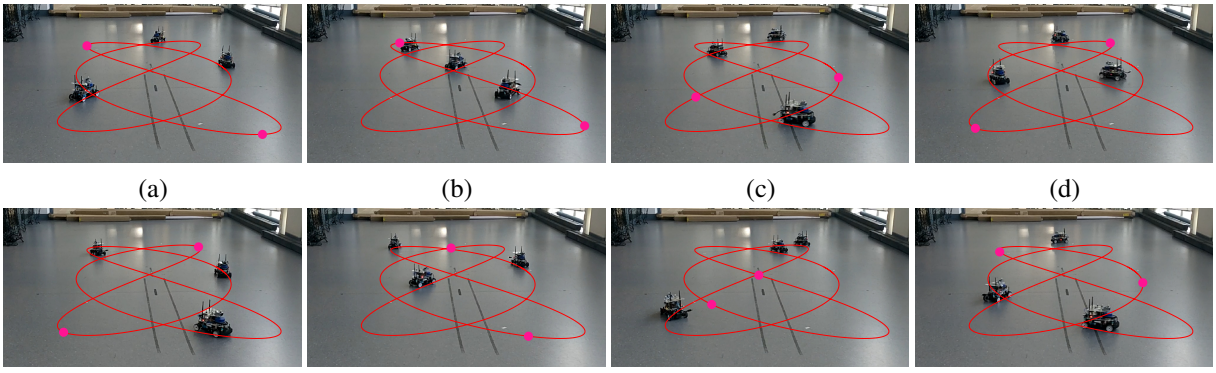

Figure 7: Experimental results with $N = 5$ robots (3 physical and 2 simulated). The robots are constrained to move on the Lissajous path described in (16). Each robot follows the time-inverted Kuramoto dynamics (1). The system starts at the equilibrium configuration and preserves it (a)-(d). At the time instant associated with (d) the agents 1 and 5 suddenly change their dynamics in $\dot{\theta}_{1,5} = 0$. The system reaches an alternative equilibrium configuration (e). After few seconds agents 1 and 5 resume their original dynamic (1) and the system comes back to the previous equilibrium configuration.

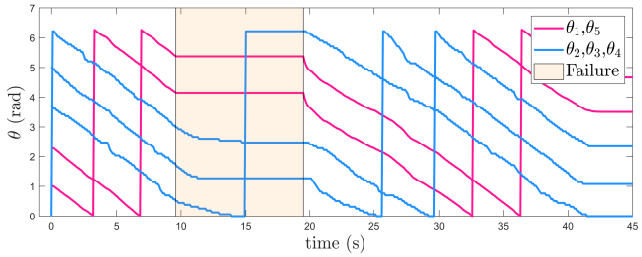

Figure 8: Evolution in time of the states of the agents in the experiment of Figure 7. In pink the states of the simulated agents, while in blue the states of the physical agents. The area in orange denotes the failure interval, where the simulated agents change their dynamics.

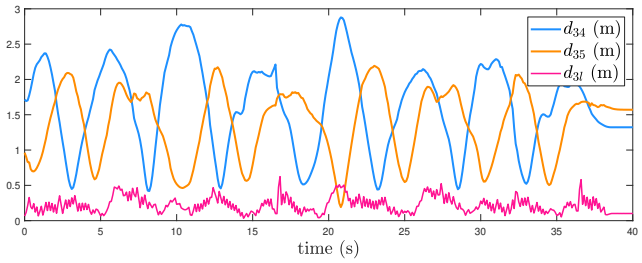

Figure 9: Evolution in time of the distance between robots 3 and 4 (d_{34}), the distance between robots 3 and 5 (d_{35}), and the tracking error for robot 3, i.e., the distance between the robot's position and the reference point on the Lissajous curve.

[8] C. Robin and S. Lacroix, "Multi-robot target detection and tracking: taxonomy and survey," *Aut. Robots*, vol. 40, no. 4, pp. 729–760, 2016.

[9] T. Chung, G. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Autonom. Robots*, vol. 31, no. 4, pp. 299–316, 2011.

[10] J. Urrutia, "Art gallery and illumination problems," in *Handbook of computational geometry*. Elsevier, 2000, pp. 973–1027.

[11] J. W. Durham, A. Franchi, and F. Bullo, "Distributed pursuit-evasion without mapping or global localization via local frontiers," *Autonomous Robots*, vol. 32, no. 1, pp. 81–95, 2012.

[12] J. P. Hespanha, H. J. Kim, and S. Sastry, "Multiple-agent probabilistic pursuit-evasion games," in *Proceedings of the 38th IEEE Conference on Decision and Control*, vol. 3. IEEE, 1999, pp. 2432–2437.

[13] G. Hollinger, S. Singh, J. Djugash, and A. Kehagias, "Efficient multi-robot search for a moving target," *The International Journal of Robotics Research*, vol. 28, no. 2, pp. 201–219, 2009.

[14] A. Borkar, A. Sinha, L. Vachhani, and H. Arya, "Collision-free trajectory planning on lissajous curves for repeated multi-agent coverage and target detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1417–1422.

[15] A. V. Borkar *et al.*, "Reconfigurable formations of quadrotors on lissajous curves for surveillance applications," *European Journal of Control*, vol. 56, pp. 274–288, 2020.

[16] W. Yao, H. G. de Marina, Z. Sun, and M. Cao, "Distributed coordinated path following using guiding vector fields," in *Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 10 030–10 037.

[17] M. Boldrer, F. Riz, F. Pasqualetti, L. Palopoli, and D. Fontanelli, "Time-inverted kuramoto dynamics for κ-clustered circle coverage," in *60th Conf. on Dec. and Control (CDC)*. IEEE, 2021, pp. 1205–1211.

[18] M. Boldrer, F. Pasqualetti, L. Palopoli, and D. Fontanelli, "Multi-agent persistent monitoring via time-inverted kuramoto dynamics," *IEEE Control Systems Letters*, 2022.

[19] W. Erb, C. Kaethner, M. Ahlborg, and T. M. Buzug, "Bivariate lagrange interpolation at the node points of non-degenerate lissajous curves," *Numerische Mathematik*, vol. 133, no. 4, pp. 685–705, 2016.

[20] H. M. Stark and H. M. Stark, *An introduction to number theory*. Mit Press Cambridge, MA, 1978.

[21] A. Prorok, M. Malencia, L. Carlone, G. S. Sukhatme, B. M. Sadler, and V. Kumar, "Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems," *arXiv preprint arXiv:2109.12343*, 2021.

[22] S. A. Gershgorin, "Uber die abgrenzung der eigenwerte einer matrix," *Izv. Akad. Nauk. USSR Otd. Fiz.-Mat. Nauk*, no. 6, pp. 749–754, 1931.

[23] L. Ferranti, L. Lyons, R. Negenborn, T. Keviczy, and J. Alonso-Mora, "Distributed nonlinear trajectory optimization for multi-robot motion planning," *IEEE Transactions on Control Systems Technology*, 2022.

[24] M. Boldrer, A. Antonucci, P. Bevilacqua, L. Palopoli, and D. Fontanelli, "Multi-agent navigation in human-shared environments: A safe and socially-aware approach," *Robotics and Autonomous Systems*, vol. 149, p. 103979, 2022.

[25] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[26] C. Lamm, "There are infinitely many lissajous knots," *manuscripta mathematica*, vol. 93, no. 1, pp. 29–37, 1997.

[27] W. Ren and R. W. Beard, *Distributed consensus in multi-vehicle cooperative control*. Springer, 2008, vol. 27, no. 2.

[28] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4459–4466, 2019.

[29] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 1094–1099.

[30] M. Boldrer, L. Lyons, L. Palopoli, D. Fontanelli, and L. Ferranti. Video associated with the submission. Youtube. [Online]. Available: https://youtu.be/pMJR5q4jNSs