2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Distributed reconfiguration of distance-based formations with virtual surface constraints

M. Guinaldo J. Sánchez-Moreno S. Zaragoza F. J. Mañas-Álvarez

***Abstract*— This paper proposes a method to recover from the failure or loss of a subset of agents in a distance-based formation problem, where the system is initially deployed forming a virtual shield embedded in the 3D space. First, a distributed algorithm is proposed to restore the topology, which is a Delaunay triangulation. After that, the nodes execute a distance-based distributed control law that considers adaptive target distances. These values are computed in parallel by the nodes, which try to reach an agreement with some constraints, given by the desired shield shape. The updating policy is based on events. The results are illustrated through simulation examples.**

I. Introduction

In recent years, the utilization of autonomous robot systems for a large number of tasks in the field of robotics has surged. One critical facet of this paradigm shift has been the deployment of cooperative multi-agent systems (MAS) in many applications such as sampling, monitoring, surveillance, and safeguarding critical infrastructures. In such MAS, each individual entity is often referred to as an *agent*, and complex behavior of the overall system can be achieved from relatively simple local commands.

Central to the effectiveness of these MAS is the maintenance of a formation among the agents [1]. Depending on the measurement capabilities of the agents, various strategies have been proposed [2]. One common approach assumes that each agent can measure distances to its neighboring agents, and then the concept of graph rigidity has allowed the design of distributed control laws for formation control [3], [4]. Related to the concept of rigidity, a Delaunay triangulation belongs to the class of proximity graphs [5], and it is the dual of the Voronoi diagram [6]. The graph of a Delaunay triangulation is rigid. Moreover, triangular formations have been shown to be robust against agent failures [7], and the concept of resilient formations has arisen. This line of research borrows from results in network science [8].

In this paper, we present a novel approach to address the issue of network reconfiguration for a team of agents with the control objective of maintaining a rigid formation with a desired given shape. A potential application of our research involves a team of Unmanned Aerial Vehicles (UAVs), deployed to form a protective "shield" around an area of interest, with the objective of safeguarding critical infrastructures, and in which one or more agents fail or are lost. We propose a two-stage method: first, it restores and maintains the topology, specifically a Delaunay triangulation; and secondly, it introduces a novel distance-based control law that considers dynamic values for the target distances between nodes, thereby ensuring the reconfigured formation remains rigid and closely approximates an almost uniform distribution of agents. It builds up on our previous work [9], where an algorithm to generate a target formation in 3D for a given number of agents and a given shape (a quadric surface), and a control law to achieve it were proposed. Then, in this paper, we assume that the system is initially as designed in [9], it loses a subset of agents, and we propose a distributed strategy to find a new configuration that preserves the almost uniform distribution of agents and a control law to reach it.

There are in the literature different approaches to deal with the problem of reconfiguration of networks or formations. In [10], an optimization formulation is used to solve the problem of constructing a network topology and generating a pairwise distance between communicating robots. However, this solution presents scalability issues, since requires the knowledge of the topology of the overall system. In [11] a heuristic rule of edge allocation is presented to approximately solve the original optimization design network problem. In [12] the authors propose a scaling strategy that could maneuver the system of mobile agents to achieve the desired formation of which the size may change (the target distances are scaled). Other works in the literature focus instead on mitigating the interference of a malicious agent on the control objective, such as [7]. The previous works differ from ours in that or the problem is different or a simplified version, or they need a centralized computation.

The rest of the paper is organized as follows: Section II introduces some preliminary concepts used throughout the paper. Section III describes the problem to be solved in this paper, whereas the proposed solution is given in Section IV. Section V illustrates with simulations the results of the paper. Finally, Section VI provides the conclusions and future work.

II. Preliminaries

A. Graph theory

Consider a set $\mathcal{N}$ of N agents. The topology of the MAS can be modeled as a static undirected graph $\mathcal{G}$, described by the set of agent-nodes $\mathcal{V}$ and the set of edges $\mathcal{E}$. For each agent i, $\mathcal{N}_i$ represents the neighborhood of i, i.e., $\mathcal{N}_i = \{j \in \mathcal{V} : \ (i,j) \in \mathcal{E}\}$. Note that $|\mathcal{N}_i| = \deg v_i$, where $|\cdot|$ represents the cardinality of the set $\mathcal{N}_i$ and deg is the degree of the vertex v_i associated to the node i.

This work was supported by Agencia Estatal de Investigación (AEI) under the Project PID2020-112658RB-I00/AEI/10.13039/501100011033. The first, second and forth authors are with the Dept. Computer Science and Automatic Control, UNED, Juan del Rosal 16, 28040 Madrid (Spain) (mguinaldo, jsanchez, fjmanas@dia.uned.es). The third author is with the Centro Universitario de la Defensa, Coronel López Peña s/n, 30729 San Javier, Murcia (Spain).

Assume that the edges have been labeled as e_k and arbitrarily oriented, and its cardinality is labeled as N_e. Then the incidence matrix $H(\mathcal{G}) = [h_{ik}] \in \mathbb{R}^{N \times N_e}$ is defined as $h_{ik} = -1$ if v_i is the tail of the edge e_k, $h_{ik} = 1$ if v_i is the head of e_k, and $h_{ik} = 0$ otherwise. The Laplacian matrix $L(\mathcal{G}) \in \mathbb{R}^{N \times N}$ of a network of agents is defined as $L(\mathcal{G}) = H(\mathcal{G})H^\top(\mathcal{G})$. The Laplacian matrix $L(\mathcal{G})$ is positive semidefinite, and if $\mathcal{G}$ is connected and undirected, then $0 = \lambda_1(\mathcal{G}) < \lambda_2(\mathcal{G}) \leq \cdots \leq \lambda_N(\mathcal{G})$, where $\{\lambda_j(\mathcal{G})\}$ are the eigenvalues of $L(\mathcal{G})$. The adjacency matrix of $\mathcal{G}$ is $A(\mathcal{G}) = [a_{ij}]$, where $a_{ij} = 1$ if there is an edge between two vertices v_i and v_j, and 0 otherwise.

Given a graph $\mathcal{G}$, the *line graph* $\mathcal{G}_e \equiv (\mathcal{V}_e, \mathcal{E}_e)$ of $\mathcal{G}$ is the graph that results from taking the edges of $\mathcal{G}$ as its vertices of the new graph and joining two whenever the corresponding edges share a vertex of $\mathcal{G}$. Similarly to the vertex-adjacency matrix, the edge-adjacency matrix can be defined, i.e., $A_e(\mathcal{G}_e) = [a^e_{kl}]$ so that $a^e_{kl} = 1$ if e_k and e_l share a vertex, and 0 otherwise. Also, the neighborhood of an edge k can be defined as $\mathcal{N}^e_k = \{l \in \mathcal{V}_e : \ a^e_{kl} = 1\}$.

B. Distributed averaging algorithms

Let us consider a system whose topology is described by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Assume that each node v_i has some value V_i. The objective of the average consensus is that all nodes calculate the average of these values: $\mu_V = \frac{1}{N}\sum_{i=1}^N V_i$. In a typical consensus formulation, each node v_i updates its state at iteration ℓ using a linear update as

$$x_i(\ell + 1) = p_{ii}x_i(\ell) + \sum_{j \in \mathcal{N}_i} p_{ij}x_j(\ell), \tag{1}$$

where $x_i(0) = V_i$.

Definition 1. The nodes are said to reach asymptotic average consensus if $\lim_{\ell \to \infty} x_i(\ell) = \frac{1}{N}\sum_{j=1}^N x_j(0) = \mu_V, \ \forall i \in \mathcal{V}$.

Definition 2. A matrix $P \in \mathbb{R}^{N \times N}_{>0}$ is called doubly stochastic if it is both column and row stochastic, that is $\sum_{i=1}^N p_{ij} = 1, \ \sum_{j=1}^N p_{ij} = 1, \ \forall i = 1, \ldots, N$.

If the weights p_{ij} in (1) are chosen so that the matrix $P = [p_{ij}]$ is doubly stochastic, then the system reaches average consensus asymptotically and the convergence rate, at the iteration k, is dominated by $|\lambda_2(P)|^k$, $|\lambda_2(P)| < 1$ [13].

C. Rigidity and distance-based formation

Let us consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N vertices embedded in $\mathbb{R}^m$, and define the composite vector $p = (p_1^\top, \ ..., \ p_N^\top)^\top \in \mathbb{R}^{mN}$. A formation is defined as a set of a graph $\mathcal{G}$ and its realization p, i.e., $(\mathcal{V}, \mathcal{E}, p)$.

In distance-based formation, it is the constraints over the distances of the edges in $\mathcal{E}$ what characterize the formation. The distance rigidity theory includes several concepts, such as distance rigidity and infinitesimal distance rigidity [3].

Let us consider the distance function $f_\mathcal{G}(p)$ as $f_\mathcal{G}(p) = \frac{1}{2}(\ldots, \|p_j - p_i\|^2, \ldots)^\top$. We define the rigidity matrix $R(p)$ as follows:

$$R(p) = \frac{\partial f_\mathcal{G}(p)}{\partial p} \in \mathbb{R}^{N_e \times mN}, \tag{2}$$

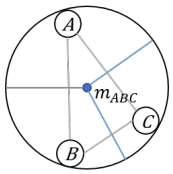

Fig. 1. The triangle formed by $\{A, B, C\}$, its circumcircle, and its circumcenter m_{ABC}.

where N_e is the number of edges of the graph $\mathcal{G}$. If $R(p)\delta p = 0$, then δp is an infinitesimal distance motion of $(\mathcal{V}, \mathcal{E}, p)$. Analyzing the properties of $R(p)$ allows us to infer further properties of the formation $(\mathcal{V}, \mathcal{E}, p)$ [3]:

Definition 3. A formation $(\mathcal{V}, \mathcal{E}, p)$ is infinitesimally rigid if $\text{rank}(R(p)) = 2N - 3$ ($\mathbb{R}^2$) or $\text{rank}(R(p)) = 3N - 6$ ($\mathbb{R}^3$).

Definition 4. A graph is minimally rigid if it is rigid and the removal of a single edge causes it to lose rigidity. Mathematically, this condition can be checked by the number of edges N_e, so that if $N_e = 2N - 3$ in $\mathbb{R}^2$ or $N_e = 3N - 6$ in $\mathbb{R}^3$ the graph is minimally rigid.

We next introduce a control system derived from the gradient descent flow. In [4], a distributed control law is proposed for distance-based formation control. Let us denote $d_{ij} = \|p_i - p_j\|$ and d^*_{ij} is the prescribed distance for the edge $(i, j) \in \mathcal{E}$. Define the potential function of the form $W = \frac{1}{4}\sum_{(i,j)\in\mathcal{E}}(d_{ij} - {d^*_{ij}}^2)^2$. The gradient descent control law for each agent i derived from this potential function is:

$$u_i = -\nabla_{p_i} W = -\sum_{j \in \mathcal{N}_i} (d_{ij} - {d^*_{ij}}^2)(p_i - p_j). \tag{3}$$

The control system $\dot{p}_i = u_i$ with u_i (3) can be studied using the infinitesimal distance rigidity property as follows [14]:

Theorem 1. *All agents close to a target infinitesimally distance rigid formation with the controller* (3) *exponentially converge to a formation consistent with desired distances.*

D. Delaunay triangulation

The following definitions and concepts are the basics for 2D Delaunay triangulations.

Definition 5. A triangulation of a set $\mathcal{P}$ points is a planar graph with vertices at the coordinates $p_i \in \mathcal{P}$ and edges that subdivide the convex hull $H(\mathcal{P})$ into triangles, so that the union of all triangles equals the convex hull.

Any triangulation with N vertices consists of $2(N-1) - N_b$ triangles and has $N_e = 3(N-1) - N_b$ edges, where N_b denotes the number of agents on the boundary $\partial H(\mathcal{P})$ of the convex hull. The edges of a triangulation do not cross each other. Furthermore, the triangulation of $N > 3$ points is not unique. The Delaunay triangulation is a proximity graph that can be constructed in terms of the circumcircle of each of the triangles (the circumcircle is the unique circle passing through the three vertices of a triangle, and its circumcenter is the center of such circle, see Fig. 1).

Definition 6. [15]. A triangle of a given triangulation of a set $\mathcal{P}$ of points is said to be Delaunay if there is no point $p_i \in \mathcal{P}$ in the interior of its circumcircle.

Definition 7. [15]. A Delaunay triangulation is a triangulation in which all triangles satisfy the local Delaunay property.

III. Problem formulation

The state of each mobile agent is described by the vector $p_i(t) = \begin{pmatrix} p_{x,i}(t) & p_{y,i}(t) & p_{z,i}(t) \end{pmatrix}^\top$, which represents the Cartesian coordinates. Let the N agents obey the single-integrator dynamics:

$$\dot{p}_i(t) = u_i(t), \quad i = 1, \ldots, N, \tag{4}$$

where $u_i(t) \in \mathbb{R}^3$ are the control inputs of agent i, which will be described later.

The team of agents is initially deployed to protect a certain area of interest that, without loss of generality, is placed around the origin, i.e., $p_0^* = \mathbf{0}$. For the aforementioned purpose, the agents form a mesh with a certain shape that we call a *shield*. We assume that the shape of this "virtual" shield is given by a quadric surface $\mathcal{S} \in \mathbb{R}^3$ described in the following compact form

$$\mathcal{S} \equiv p^\top Q_1 p + Q_2 = 0, \tag{5}$$

where $p \in \mathbb{R}^3$, $Q_1 \in \mathbb{R}^{3\times 3}$ such that $Q_1 = Q_1^\top$, and $Q_2 \in \mathbb{R}$. Additionally, since the shield is deployed around the point $p_0^* = \mathbf{0}$, we consider quadric surfaces in their normal form [16], which imposes some constraints on the values of Q_i. Furthermore, the shield might require the definition of some additional constraints for the positioning of the agents, for example, having an upper and/or lower bound on some of the coordinates. In general, we constrain $z \geq 0$. For example, for an hemispherical shield, $Q_1 = \frac{1}{R^2} I_3$ and $Q_2 = -1$, with $z \geq 0$, where $I_3 \in \mathbb{R}^{3\times 3}$ is the identity matrix and R is the radious of the sphere.

For an agent i, we can define a function $f_{\mathcal{S}}(p_i)$ such that $f_{\mathcal{S}}(p_i) = 0$ iff $p_i \in \mathcal{S}$:

$$f_{\mathcal{S}}(p_i) = p_i^\top Q_1 p_i + Q_2. \tag{6}$$

In [9], an algorithm is proposed to the *almost* uniform distribution of the N agents in $\mathcal{V}$ over the virtual surface $\mathcal{S}$ (5), and the generation of a Delaunay triangulation for the topology that provides target distances d_{ij}^* for all edges in $\mathcal{E}$. In the current paper, we assume that this is the initial configuration for the agents. Moreover, a distributed gradient descent control law that adds an additional term to (3)

$$u_i = -\kappa_1 \sum_{j\in\mathcal{N}_i} (d_{ij}^2 - d_{ij}^{*2})(p_i - p_j) - \frac{\kappa_2}{2} f_{\mathcal{S}}(p_i) \frac{\partial f_{\mathcal{S}}(p_i)}{\partial p_i}, \tag{7}$$

where $f_{\mathcal{S}}(p_i)$ given in (6), is also proposed in [9], proving that the target formation is achieved locally and asymptotically while obtaining the desired shield shape.

Now assume that a subset of the N agents of the team is lost due to internal (loss of connectivity or other failures) or external (attack) reasons. Let us denote this subset as $\mathcal{N}_{loss}$ with $|\mathcal{N}_{loss}| = N_{loss} \geq 1$. Then, we are ready to formulate the problem statement:

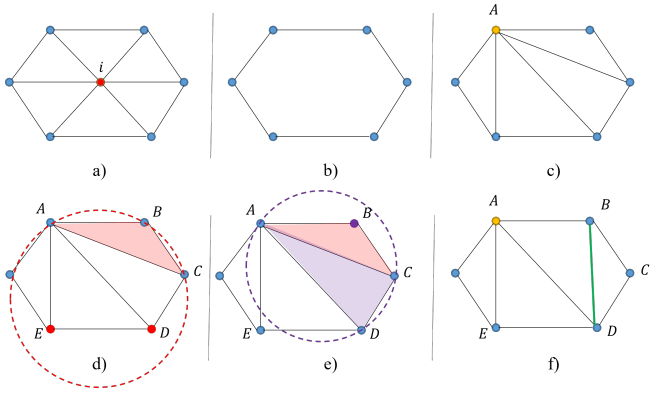

Fig. 2. Example of links reconfiguration before executing Algorithm 4.

Problem 1. *Given the team of N agents* (4) *deployed and forming the virtual shield described by* (5) *that loses a subset of them, $\mathcal{N}_{loss}$, find a procedure that reconfigures the system so that an almost uniform distribution for the agents is recovered and that can be executed distributedly.*

IV. Problem solution

Next, we present the procedure to reconfigure the shield after the loss of one or more agents. The idea is to redistribute the rest of the agents to fill the gap left by the lost nodes. The proposed solution has a twofold contribution: 1) The restoration and maintenance of the topology, and 2) a control law based on (7) that considers dynamic values for the target distances between nodes instead of fixed values d_{ij}^*.

We distinguish two types of nodes in the network: fixed (*anchor*) and moving. The anchors denoted by $\mathcal{N}_{\mathcal{F}}$ keep fixed positions and the distances between anchors also remain constant. Typically, the anchors are the nodes in the boundary of the triangulation $\partial\mathcal{T}$ (at $z = 0$). The moving agents are denoted by $\mathcal{N}_{\mathcal{M}}$ such that $\mathcal{V} = \mathcal{N}_{\mathcal{F}} \cup \mathcal{N}_{\mathcal{M}}$.

Assumption 1. The lost agents are all moving agents, i.e., $\mathcal{N}_{loss} \notin \mathcal{N}_{\mathcal{F}}$. Furthermore, the number of losses N_{loss} is upper bounded by $|\mathcal{N}_{\mathcal{M}}| - 1$.

A. Restoring and maintaining the topology

To illustrate the procedure by a graphical example first. For simplicity, we consider the case of 1 lost agent, but the designed algorithm is for a general set. Moreover, the drawing is presented in the plane, but we remark that the setting of this work is on a surface $\mathcal{S} \in \mathbb{R}^3$. The initial configuration is depicted in Fig. 2a. Assume that agent i (in red) is lost (Fig. 2b). One of the neighboring agents ($A \in \mathcal{N}_{\mathcal{M}}$) takes control and creates new links (Fig. 2c).

The agents need to maintain local information $\mathcal{I}_j$ that includes: the position p_j, the set of neighbors, $\mathcal{N}_j$, and the set of triangles that agent j forms part of, $\mathcal{T}_j$. All the nodes j in the boundary of the gap, $\partial\mathcal{H}$, execute these two actions: $\mathcal{N}_j \leftarrow \mathcal{N}_j \setminus \{i\}$, $\forall t \in \mathcal{T}_j : \; i \in t$, $\mathcal{T}_j \leftarrow \mathcal{T}_j \setminus t$, that is, all the triangles that included node i are removed. Then, Algorithm

Algorithm 1 Restoring triangulation algorithm for node A

Require: $\mathcal{I}_A = \{p_A, \mathcal{N}_A, \mathcal{T}_A\}$
for $j \in \partial\mathcal{H}$
  if $j \notin \mathcal{N}_A$
    $\mathcal{N}_A \leftarrow \mathcal{N}_A \cup \{j\}$
    Find $j' \in \mathcal{N}_A \cap \partial\mathcal{H} : \ j' \in \mathcal{N}_j$
    $\mathcal{T}_A \leftarrow \mathcal{T}_A \cup \{j, j'\}$
  end
end

Algorithm 2 Restoring triangulation algorithm for $j \in \partial\mathcal{H}$

Require: $\mathcal{I}_j = \{p_j, \mathcal{N}_j, \mathcal{T}_j\}$
if $A \notin \mathcal{N}_j$
  $\mathcal{N}_j \leftarrow \mathcal{N}_j \cup \{A\}$
end
find $j' \in \mathcal{N}_j \cap \partial\mathcal{H} : \ j' \in \mathcal{N}_A$
  $\mathcal{T}_j \leftarrow \mathcal{T}_j \cup \{A, j'\}$
end

1 shows the local information is updated for the agent A that takes control to restore the triangulation. The rest of the nodes $j \in \partial\mathcal{H}, j \neq A$, execute Algorithm 2.

Once the triangulation is restored, the new triangles might not meet Delaunay's condition. The following result provides a condition that can be checked locally by the three nodes that form a triangle. Let us denote them as A, B, and C and their positions as $p_A, p_B, p_C \in \mathbb{R}^3$, respectively. The idea is to check if any node D at $p_D \in \mathbb{R}^3$ is at a distance shorter than the radius of the circumcircle of the triangle formed by A, B, C (see Fig. 1). This is an extension of the results in $\mathbb{R}^2$ of [17] to an embedded surface $\mathcal{S} \in \mathbb{R}^3$.

Theorem 2. *Let $O_{ABC} = (p_A \ p_B \ p_C)$, and the plane in which the triangle formed by A, B, C is embedded as f_π with normal vector v_π. If there not exist any other node D with position p_D that satisfies that $|M_{ABCD}| > 0$, then the triangle formed by A, B, C is Delaunay's, where*

$$M_{ABCD} = \begin{pmatrix} \Lambda & v_P \\ \begin{pmatrix} p_D^\top & 1 \end{pmatrix} & \|p_D\|^2 \end{pmatrix}, \Lambda = \begin{pmatrix} O_{ABC} & \mathbf{1} \\ v_\pi^\top & 0 \end{pmatrix}, \tag{8}$$

with $v_P^\top = (\|p_A\|^2 \ \|p_B\|^2 \ \|p_C\|^2 \ 2|O_{ABC}|)$ and $\mathbf{1} = (1 \ 1 \ 1)^\top$.

Proof: The proof is omitted here due to space constraints but can be deduced following Lemmas 1-2 and Theorem 1 in [9]. □

Remark 1. The plane f_π and its normal vector v_π are obtained easily from relative coordinates between the points, for example, $p_{BA} = p_B - p_A, p_{CA} = p_C - p_A$.

If any of the new created triangles in the gap is not Delaunay, then the algorithm known as Lawson Flip is applied [17]. Fig. 2d-f illustrates it with an example. For the triangle formed by A, B, C, nodes D and E violates the condition of Theorem 2, and this also holds for the triangle ACD and node B, respectively. Then link AC is replaced by BD. Note that when a triangle is not Delaunay's, so is one of the adjacent, and the problematic edge is the one

Algorithm 3 Lawson Flip Algorithm

1: **Require:** $\mathcal{I}_A = \{p_A, \mathcal{N}_A, \mathcal{T}_A\}$
2: Identify the two neighbors B and D that share the common edge AC using $\mathcal{T}_A$
3: Gather current pos. p_B, p_C, p_D to construct M_{ABCD}
4: **if** $|M_{ABCD} > 0|$
5: $\mathcal{N}_A \leftarrow \mathcal{N}_A \setminus \{C\}$, $\mathcal{T}_A \leftarrow \mathcal{T}_A \setminus \{B, C\}, \{C, D\}$
6: $\mathcal{T}_A \leftarrow \mathcal{T}_A \cup \{B, D\}$
7: **end**

that that the two triangles share in common, in the case of Fig. 2, AC. Algorithm 3 lists the set of actions taken by agent A. Node C executes similar actions but replacing lines 6-9 by $\mathcal{N}_C \leftarrow \mathcal{N}_C \setminus \{A\}$, $\mathcal{T}_C \leftarrow \mathcal{T}_C \setminus \{A, B\}, \{A, D\}$, and $\mathcal{T}_C \leftarrow \mathcal{T}_C \cup \{B, D\}$. Finally, agents B and D updates its local information as follows: $\mathcal{N}_B \leftarrow \mathcal{N}_B \cup \{D\}$, $\mathcal{T}_B \leftarrow \mathcal{T}_B \setminus \{A, C\}, \mathcal{T}_B \leftarrow \mathcal{T}_B \cup \{A, D\}, \{C, D\}$; $\mathcal{N}_D \leftarrow \mathcal{N}_D \cup \{B\}$, $\mathcal{T}_D \leftarrow \mathcal{T}_D \setminus \{A, C\}$, $\mathcal{T}_D \leftarrow \mathcal{T}_D \cup \{A, B\}, \{B, C\}$.

B. Adaptive distance-based control law

After the loss of the agents, the condition of almost uniform distribution does not hold anymore. Hence, the method presented next has the objective of recovering this property on the virtual surface. For that purpose, the control law (7) is adapted to consider target values d_{ij}^* that are updated at discrete instances of time ℓ and denoted by μ_{ij}:

$$u_i = -\kappa_1 \sum_{j \in \mathcal{N}_i} (d_{ij}^2 - \mu_{ij}^2(\ell))(p_i - p_j) - \kappa_2 f_{\mathcal{S}}(p_i) \frac{\partial f_{\mathcal{S}}(p_i)}{\partial p_i}. \tag{9}$$

The proposed updating law for μ_{ij}^2 uses the concepts presented in Section II-A and Section II-B. In an ideal case, when no boundaries and other constraints exist, an equal value for all the target distances d_{ij}^* is compatible with a uniform distribution of all nodes in a plane. If updating laws of the form (1) are used with adequate choices of weights p_{ij} for d_{ij}^* (see Definition 2), they result in a common value that is the average of the initial values. One choice of p_{ij} compatible with distributed implementations are the Metropolis weights [13]. Let us denote $\mu_{ij}^2 \equiv \mu_k^2$, i.e., and let us formulate the problem in terms of the line graph of $\mathcal{G}$, $\mathcal{G}_e$. Furthermore, Assumption 1 allows us to introduce anologous categories for the edges, i.e, fixed and moving edges, respectively, $\mathcal{N}_\mathcal{F}^e$ and $\mathcal{N}_\mathcal{M}^e$.

Definition 8. An edge k that connects two nodes i and j is in the set $\mathcal{N}_\mathcal{F}^e$ if $i, j \in \mathcal{N}_\mathcal{F}$, i.e, both agents are anchors in the system. However, $k \in \mathcal{N}_\mathcal{M}^e$ if $i \in \mathcal{N}_\mathcal{M}$ or $j \in \mathcal{N}_\mathcal{M}$, i.e., if at least one the agents is not an anchor.

Then, we propose the following updating law with Metropolis weights:

$$\mu_k^2(\ell + 1) = p_{kk}\mu_k^2(\ell) + \sum_{l \in \mathcal{N}_k^e \cap \mathcal{N}_\mathcal{M}^e} p_{kl}\mu_l^2(\ell), \tag{10}$$

$$p_{kl} = \begin{cases} \frac{1}{1 + \max(deg\ e_k, deg\ e_l)}, & \forall l \in \mathcal{N}_k^e \cap \mathcal{N}_\mathcal{M}^e \\ 1 - \sum_{j \in \mathcal{N}_k^e \cap \mathcal{N}_\mathcal{M}^e} p_{kj}, & k = l \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

Algorithm 4 Algorithm to update target distances

Require: p_i, $p_{ij}: \ j \in \mathcal{N}_i$, $f_{\mathcal{S}}$
if $i \in \mathcal{N}_{\mathcal{M}}$
Compute $e_{ij} = d_{ij}^2 - \mu_{ij}^2$
If $|e_{ij}| \leq \delta \mu_{ij}^2$
  Compute $\theta_{i\to j}$ according to (14) for all $j \in \mathcal{N}_i$
  Send each $\theta_{i\to j}$ and $|\mathcal{N}_i|$ to the neighbor j
  Receive $\theta_{j\to i}$ and $|\mathcal{N}_j|$ from each neighbor $j \in \mathcal{N}_i$
  Compute for each edge ij μ_{ij}^2 according to (12)
  Update μ_{ij}^2 in control law in (9)
end
end

that is, the target value for the distance of each edge k is set in terms of the neighboring edges (those that share a vertex with k), excluding those that are connecting two fixed nodes. However, edges are not physical entities with computing capacity. Then, we need to transform updating law (10)-(11) to other that can be computed at the nodes.

Proposition 1. *The updating control laws* (10)-(11) *can be computed distributedly at the nodes $i \in \mathcal{N}_{\mathcal{M}}$ for the target square distances μ_{ij}^2 in* (9)*, $\forall j \in \mathcal{N}_i$, as follows*

$$\mu_{ij}^2(\ell+1) = \mu_{ij}^2(\ell) + \theta_{i\to j}(\ell) + \theta_{j\to i}(\ell), \tag{12}$$

if $i, j \in \mathcal{N}_{\mathcal{M}}$, and

$$\mu_{ij}^2(\ell+1) = \mu_{ij}^2(\ell) + \begin{cases} \theta_{i\to j}(\ell) \text{ if } j \in \mathcal{N}_{\mathcal{F}} \\ \theta_{j\to i}(\ell) \text{ if } i \in \mathcal{N}_{\mathcal{F}}, \end{cases} \tag{13}$$

$$\theta_{i\to j} = \sum_{l\in(\mathcal{N}_i\setminus j)} \frac{1}{|\mathcal{N}_i|+max(|\mathcal{N}_j|,|\mathcal{N}_l|)-1}\left(\mu_{il}^2 - \mu_{ij}^2\right) \tag{14}$$

$$\theta_{j\to i} = \sum_{l\in(\mathcal{N}_j\setminus i)} \frac{1}{|\mathcal{N}_j|+max(|\mathcal{N}_i|,|\mathcal{N}_l|)-1}\left(\mu_{jl}^2 - \mu_{ij}^2\right). \tag{15}$$

Proof: The proof is omitted due to space constraints.

C. Event-based implementation

The system (4) with control law (7) is locally asymptotically stable [9], so that if the initial configuration is in a given formation over the surface, then small variations in d_{ij}^* are also asymptotically stable if the formation is feasible. However, the feasibility of new target distances is difficult to be known a priori. Thus, we propose an event-based implementation of (12)-(13). Then, μ_{ij}^2 is only updated if

$$|e_{ij}| = |d_{ij}^2 - \mu_{ij}^2(\ell)| \leq \delta \mu_{ij}^2(\ell), \tag{16}$$

where $\delta \in (0,1)$. That way, μ_{ij}^2 is only updated when the relative error goes beyond δ. The event-based updating mechanism (16) will make that the target square distances μ_{ij}^2 are not updated synchronously. Therefore, the convergence to the average of the initial values is no longer guaranteed, and the final value of target distances for each edge will depend on the sequence of events, which, in the end, depends on the feasibility of the updated target formation. The analytical study is left for future work. Algorithm 4 summarizes the proposed solution.

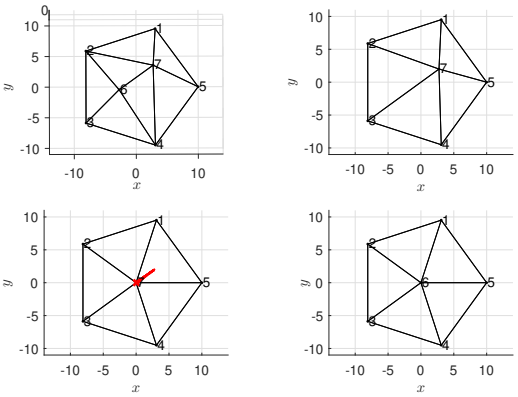

Fig. 3. 2D view of situations 1)-4) in Example 1.

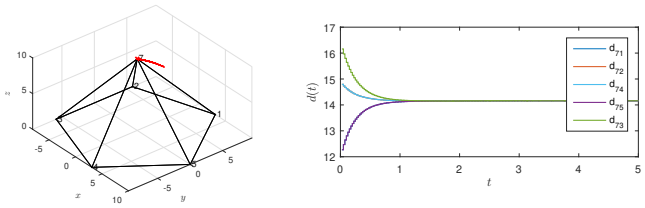

Fig. 4. 3D view of the shield and trajectory of agent 7 (left), and time evolution of the distances between agents 1-5 to 7 (right) in Example 1.

V. Simulation Example

We illustrate the proposed strategy through two examples with different N, shapes, and N_{loss}. We analyze four situations: 1) The initial configuration with N agents; 2) after the loss of one or more agents (N_{loss}) once the Delaunay condition of all triangles is checked and before the execution of Algorithm 4; 3) after the execution of Algorithm 4; and 4) the configuration generated by the Algorithm proposed in [9] with $N - N_{loss}$. The loss of the agent(s) is generated randomly. The trigger function (16) is defined with $\delta = 0.05$.

A. Example 1

Let us consider a hemispherical shield of $N = 7$ agents and $R = 10$ that loses one of them ($i = 6$). Fig. 3 shows the projection over the XY plane of the four situations 1)-4) described above. Agent 7 executes the procedure presented in Section IV-A to restore the triangulation (see Fig. 3 top-right). In this case, the Delaunay's condition is not violated. Observe that agent 7 moves and reaches the zenith of the semi-sphere (Fig. 4 left). In that case, the distances of all moving edges, d_{i7} with $i = 1, \ldots, 5$, converge to a common value (14.16), very close to the mean of the initial values of d_{i7} (14.06) (see Fig. 4 right). Note that Algorithm 4 reaches a common value because this configuration is feasible over the surface. Moreover, the error to the surface is $f_{\mathcal{S}}(p_7) = 0.005$.

B. Example 2

Let us consider a shield of $N = 25$ agents and with a given semi-ellipsoid shield of the form $\frac{x^2}{10^2} + \frac{y^2}{8^2} + \frac{z^2}{7^2} = 1, z \geq 0$,

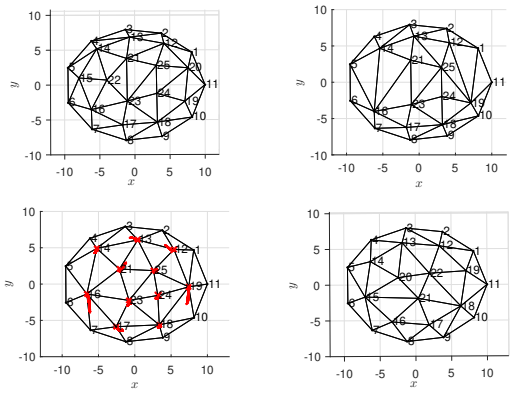

Fig. 5. 2D view of situations 1)-4) in Example 2.

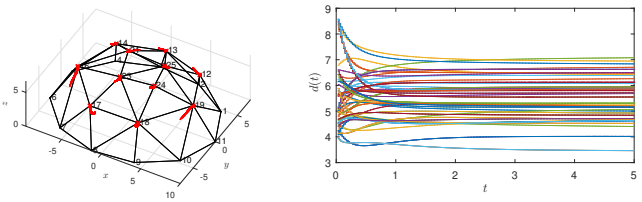

Fig. 6. 3D view of the shield and trajectories of the agents (left), and time evolution of the distances of moving edges in Example 2.

that loses three agents (agent 15, 20, and 22). Fig. 5 shows the projection over the XY in cases 1)-4), and the 3D view of the moving agents (12-25) is shown on the left of Fig. 6. The number of edges in this case is 52, 41 of them are moving. The convergence analysis of the distances between agents is not straightforward, but the time evolution of $d(t)$ is shown on the right of Fig. 6. A useful tool when the number of edges is large is the histogram, shown in Fig. 7. On the top-left, the initial distribution of distances is shown (case 1)). When the agents 15, 20, and 22 are removed, this distribution changes according to the top-right plot. After the execution of Algorithm 4, the distribution of agents is shown in the bottom-left. Note that this distribution has a symmetric shape that fits with a Gaussian. Actually, if we compute the average distances for cases 3) and 4), we obtain similar values, 5.43 and 5.49, respectively, but the distribution of distances is more symmetric in case 3) than in case 4). This illustrates the good performance of the proposed method (Algorithm 4). Finally, if the distance to the surface is computed for all moving agents, all are bounded by 0.025.

VI. Conclusions and Future Work

In this paper, we have proposed a distributed strategy to reconfigure a distance-based formation for multi-agent systems that suffer a failure or loss of a subset of agents, consisting of two steps. First, the topology (triangulation) is restored creating new links, which might be flipped if the Delaunay's condition, which is evaluated distributedly, does not hold. Secondly, new values for target distances are computed following a distributed event-based updating

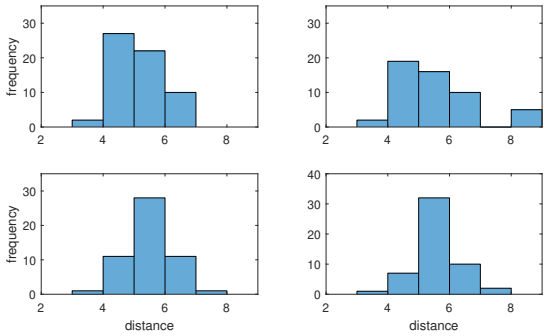

Fig. 7. Histogram of distances for the situations 1)-4) in Example 2.

mechanism, and used in the control law. The results show that convergence properties depend on the number of constraints, but in any case, the resulting formation approaches a uniform distribution of nodes over the virtual surface.

Future work will include the study of the impact of malicious agents over the resulting formation and the validation over an experimental platform.

References

[1] J. R. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE trans. Robot. Autom.*, vol. 19, no. 6, pp. 933–941, 2003.

[2] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[3] B. D. Anderson *et al.*, "Rigid graph control architectures for autonomous formations," *IEEE Control. Syst. Mag.*, vol. 28, no. 6, pp. 48–63, 2008.

[4] L. Krick, M. E. Broucke, and B. A. Francis, "Stabilisation of infinitesimally rigid formations of multi-robot networks," *Int. J. control*, vol. 82, no. 3, pp. 423–439, 2009.

[5] L. Mathieson and P. Moscato, "An introduction to proximity graphs," *Bus. Consumer Anal. New Ideas*, pp. 213–233, 2019.

[6] Ø. Hjelle and M. Dæhlen, *Triangulations and applications*. Springer Science & Business Media, 2006.

[7] D. Saldana, A. Prorok, M. F. Campos, and V. Kumar, "Triangular networks for resilient formations," in *Distributed Autonomous Robotic Systems: The 13th International Symposium*, 2018, pp. 147–159.

[8] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE J. on Sel. Areas Commun.*, vol. 31, no. 4, pp. 766–781, 2013.

[9] M. Guinaldo, J. Sánchez-Moreno, S. Zaragoza, and F. J. Mañas-Álvarez, "Distributed multi-UAV shield formation based on virtual surface constraints," *Robot. Auton. Syst. (accepted)*, 2024.

[10] R. K. Ramachandran, J. A. Preiss, and G. S. Sukhatme, "Resilience by reconfiguration: Exploiting heterogeneity in robot teams," in *2019 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 6518–6525.

[11] D. Xue, A. Gusrialdi, and S. Hirche, "A distributed strategy for near-optimal network topology design," in *21st International Symposium on Mathematical Theory of Networked and Systems (MTNS 2014)*, 2014.

[12] M.-C. Park, K. Jeong, and H.-S. Ahn, "Formation stabilization and resizing based on the control of inter-agent distances," *Int. J. Robust Nonlinear Control.*, vol. 25, no. 14, pp. 2532–2546, 2015.

[13] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. & Control. Lett.*, vol. 53, no. 1, pp. 65–78, 2004.

[14] Z. Sun, U. Helmke, and B. D. O. Anderson, "Rigid formation shape control in general dimensions: an invariance principle and open problems," in *54th IEEE Conf. Decis. Control*, 2015, pp. 6095–6100.

[15] B. Delaunay, "Sur la sphere vide," *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, vol. 7, no. 793-800, 1934.

[16] S. Venit and W. Bishop, *Elementary Linear Algebra, Brooks*. International Thompson Publishing, 1996.

[17] A. Schwab and J. Lunze, "A distributed algorithm to maintain a proximity communication network among mobile agents using the delaunay triangulation," *Eur. J. Control.*, vol. 60, pp. 125–134, 2021.