Distributed Predictive Connectivity Control for Double Integrator Agents based on a Receding Horizon Scheme

Touraj Soleymani, *Student Member*, *IEEE*, Emanuele Garone, *Member*, *IEEE*, and Marco Dorigo, *Fellow*, *IEEE*

***Abstract*— In this paper, we propose a distributed constrained connectivity control algorithm for a network of dynamically decoupled agents with constrained discrete-time linear dynamics. This control algorithm works based on a receding horizon control (RHC) scheme and acts as a middleware that modifies the set-points defined by the user or by high-level control units whenever their direct application would violate system constraints. To guarantee the connectivity of the communication graph, the algorithm enforces that a specific spanning tree exists at each time. The algorithm is allowed, under certain conditions, to switch between interaction graphs in order to enhance system performance. Among all possible spanning trees, we propose to use the Euclidean Minimum Spanning Tree (EMST), and we study its advantages. The overall algorithm is described, and some of its properties are pointed out. Some simulations conclude the paper and show the effectiveness of the proposed method.**

***Index Terms*—Connectivity Control, Proximity Networks, Distributed Command Governor, Constrained Control.**

I. Introduction

In recent years many distributed control and coordination schemes have been introduced in the literature for mobile agents in proximity networks [1]. Examples include formation flight [2], coverage [3], flocking [4], and partitioning [5]. A crucial constraint in proximity networks is connectivity maintenance, which is required to guarantee the availability of a connected communication network for the agents.

A measure often used for studying connectivity is the second smallest eigenvalue of the Laplacian matrix, known as the Fiedler value or algebraic connectivity. Kim and Mesbahi [6] proposed a control algorithm that maximizes the Fiedler value through centralized semi-definite programming. Later, De Gennario and Jadbabaie [7] proposed a distributed super-gradient algorithm based on the Fiedler vector to maximize the algebraic connectivity.

In a different approach, Zavlanous and Pappas [8] used the sum of powers of the adjacency matrix, called k-hop connectivity matrix, to maintain k-connectivity in the network. In [9], the same authors used the determinant of the reduced Laplacian matrix to form a potential field which preserves connectivity. To avoid infinite control inputs whenever communication links tend to be lost in the potential fields, Dimarogonas and Johansson [10] and Ajorlou et al. [11] proposed distributed algorithms which produce bounded control inputs.

The above-mentioned approaches do not allow the deletion of communication links after they have been established. This can be quite conservative whenever connectivity control is integrated with mobility control. In [12], Zavlanous et al. proposed a distributed hybrid control which allows both the addition and deletion of links. In their work, agents agree on sequential deletions by using an auction-based algorithm which guarantees that no violation of connectivity occurs. In [13], Aragues et. al employed the minimum spanning tree for coverage control with connectivity maintenance. For a detailed list of works on connectivity, please refer to [14] and the references therein.

Most of the works on connectivity control assume that the agent dynamics is a single integrator. Moreover, the few papers that consider double integrators [15], [16], [17] or nonholonomic dynamics [18] do not fully take into account constraints on the dynamics. However, in real-world applications, dynamics is subject to different limitations such as input saturations, and state constraints. For this reason, control algorithms based on reactive control and potential functions which are not responsible for the consequences of the current actions, can yield constraint violations, as highlighted in [19].

In this paper, we propose a novel distributed constrained connectivity control algorithm which is able to guarantee the connectivity of a network of agents with constrained dynamics, e.g., with local actuator, velocity, and position constraints. This control algorithm works based on a receding horizon control (RHC) scheme, i.e., the turn-based distributed Command Governor (CG) recently introduced in [20]. The proposed connectivity control algorithm will act as a middleware that modifies the set-points defined by the user or by high-level control units whenever their direct application would violate system constraints.

In particular, to guarantee the connectivity of the communication graph the algorithm enforces that a specific spanning tree exists at each time. To make the interaction graph adaptive, the algorithm is allowed, under certain conditions, to switch between interaction graphs. Among all spanning trees, we propose to use the Euclidean Minimum Spanning Tree (EMST), and we study its advantages.

The organization of the paper is as follows. In Section II, we provide the problem statement. In Section III, we detail the connectivity control algorithm. In Section IV, we present

T. Soleymani is with Technische Universität München, Barer Str. 21, 80333 Munich, Germany (email: touraj@tum.de).

E. Garone, and M. Dorigo are with Université Libre de Bruxelles, Ave. F. Roosevelt 50, 1050 Brussels, Belgium (email: {egarone,mdorigo}@ulb.ac.be).

the results of simulations, and finally in Section V, we draw some conclusions.

II. Problem Statement

Consider a network composed of N dynamically decoupled agents with discrete-time linear dynamics (e.g., double integrator). The dynamics of each agent i is described by

$$x_i(t+1) = Ax_i(t) + Bu_i(t) \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times \bar{m}}$, and $x_i(t) \in \mathbb{R}^n$ and $u_i(t) \in \mathbb{R}^{\bar{m}}$ denote the state and control input of agent i at time step t, respectively. The state of the agent can be decomposed as $x_i(t)^T = [p_i(t), v_i(t)]$ where $p_i(t)$ is the position of the agent, and $v_i(t)$ is its velocity. The state and control input are subject to local convex constraints

$$x_i(t) \in \mathcal{X}_i,\ u_i(t) \in \mathcal{U}_i,\ \forall t \geq 0 \tag{2}$$

where $\mathcal{X}_i$ and $\mathcal{U}_i$ are state and input constraint sets, respectively.

We assume that the agents can communicate with each other. The *communication graph* can be represented by a time-varying undirected graph $\mathcal{G}_C(t) = (\mathcal{V}, \mathcal{E}_C(t))$ with the vertex set $\mathcal{V} = \{1, \dots, N\}$ and the edge set $\mathcal{E}_C(t) \subseteq \mathcal{V}\times\mathcal{V}$ where $(i,j) \in \mathcal{E}_C(t)$ if and only if agents i and j can communicate. In the sequel, we model the communication graph by a proximity graph. Hence, two agents i and j can communicate when their relative Euclidean distance is smaller than certain radius R, i.e., $||p_i(t) - p_j(t)|| \leq R$.

Definition 1 (Graph connectivity): An undirected graph $\mathcal{G}$ is connected if there exists a path between every two vertices of the graph.

Connectivity can be defined from the algebraic graph theory perspective:

Lemma 1: [21] Let $\lambda_1(\mathcal{G}) \leq \lambda_2(\mathcal{G}) \leq \dots \leq \lambda_N(\mathcal{G})$ be the ordered eigenvalues of the Laplacian matrix $\mathcal{L}(\mathcal{G})$. Then, the graph $\mathcal{G}$ is connected if and only if $\lambda_2(\mathcal{G}) > 0$.

Assuming that each agent plays a non-cooperative game, the objective of this paper is to determine a distributed supervision scheme that, whenever necessary, modifies the behavior of each single agent to ensure that the communication graph is always connected and that the local constraints are satisfied.

III. Distributed Constrained Connectivity Control

A. Distributed Command Governor

We adopt the turn-based distributed CG, and extend it to the case of general convex constraints appropriate for our problem. Assume that each agent has been pre-compensated so as to be asymptotically stable. Then, the dynamics of agent i is described as

$$x_i(t+1) = \Phi x_i(t) + Gg_i(t) \tag{3}$$

where $\Phi \in \mathbb{R}^{n\times n}$ is a Schur matrix (all eigenvalues smaller than 1), $G \in \mathbb{R}^{n\times m}$, and $g_i(t) \in \mathbb{R}^m$ is the command (manipulable reference input) of the agent i. For the sake of simplicity, it is assumed that all the systems have the same dynamics. We also assume that the reference input $r_i(t) \in \mathbb{R}^m$ of each agent is expressed by the user or a high-level control unit.

The aggregate system is subject to convex constraints

$$c_j\big(x(t), g(t)\big) \leq 0, \quad j = 1, ..., n_c \tag{4}$$

to be satisfied at each time step, where $x(t)^T = [x_1(t), \dots, x_N(t)]$ and $g^T(t) = [g_1(t), \dots, g_N(t)]$ denote the aggregate system state and aggregate command, respectively.

The goal of the distributed CG is to determine, at each time step t and for each agent i, a suitable command $g_i(t)$ that is the best approximation of the reference input $r_i(t)$ such that it would fulfill the global constraints (4) if it were constantly applied from t onward.

To this end, we define the *interaction graph* $\mathcal{G}_I(t) = (\mathcal{V}, \mathcal{E}_I(t))$ with $\mathcal{E}_I(t) \subseteq \mathcal{V}\times\mathcal{V}$ where $(i,j) \in \mathcal{E}_I(t)$ if and only if agents i and j interact with each other. Since agents are dynamically decoupled, the only interaction between agents is due to the inter-agent constraints (i.e., connectivity constraints). To characterize constraints sparsity, let $\mathcal{I}_i$ denote the set of indices of constraints depending on agent i, i.e., $j \in \mathcal{I}_i$ if and only if $c_j(x(t), g(t)) < 0$ explicitly depends on $x_i(t)$ or $g_i(t)$.

The set of neighbors $\mathcal{N}_i$ of agent i in the interaction graph is the set of agents that share at least one constraint with agent i, i.e.,

$$\mathcal{N}_i = \{j |\ \mathcal{I}_i \cap \mathcal{I}_j \neq \varnothing, j \neq i\}.$$

At this point, if we introduce $x_{\mathcal{N}_i}(t)$, $g_{\mathcal{N}_i}(t)$ as the aggregate state and aggregate command of the agents in $\mathcal{N}_i$, the constraints (4) associated with agent i can be rewritten as

$$c_j\big(x_i(t), g_i(t), x_{\mathcal{N}_i}(t), g_{\mathcal{N}_i}(t)\big) \leq 0, \quad \forall j \in \mathcal{I}_i. \tag{5}$$

To satisfy the constraints (5), it is enough to select g_i such that for any $i \in \mathcal{V}$ [1]

$$c_j\big(\hat{x}_i(k, x_i, g_i), g_i, \hat{x}_{\mathcal{N}_i}(k, x_{\mathcal{N}_i}, g_{\mathcal{N}_i}), g_{\mathcal{N}_i}\big) \leq 0, \quad k = 0, 1, ..., \infty,\ \forall j \in \mathcal{I}_i \tag{6}$$

where

$$\hat{x}_i(k, x_i, g_i) = \Phi^k x_i + \sum_{\tau=0}^{k-1} \Phi^{k-\tau-1} G g_i$$
$$\hat{x}_{\mathcal{N}_i}(k, x_{\mathcal{N}_i}, g_{\mathcal{N}_i}) = (I \otimes \Phi^k) x_{\mathcal{N}_i} + (I \otimes \sum_{\tau=0}^{k-1} \Phi^{k-\tau-1} G) g_{\mathcal{N}_i}$$

are the state prediction of agent i and neighbors of agent i along the virtual time k, respectively, and $\otimes$ denotes the Kronecker product.

Using standard arguments in CG [22], the necessity to check an infinite number of constraints in (6) can be relaxed at the cost of a slight increase in conservativity by using the following two sets of constraints

$$\begin{aligned} &c_j\big(\hat{x}_i(k, x_i, g_i), g_i, \hat{x}_{\mathcal{N}_i}(k, x_{\mathcal{N}_i}, g_{\mathcal{N}_i}), g_{\mathcal{N}_i}\big) \leq 0 \\ &c_j\big(\bar{x}_i(g_i), g_i, \bar{x}_{\mathcal{N}_i}(g_{\mathcal{N}_i}), g_{\mathcal{N}_i}\big) + \delta \leq 0, \\ &k = 0, 1, ..., k_0,\ \forall j \in \mathcal{I}_i \end{aligned} \tag{7}$$

[1] In the following, when it is obvious, for simplicity we drop the argument time t.

where

$$\bar{x}_i(g_i) = (I - \Phi)^{-1} G g_i$$
$$\bar{x}_{\mathcal{N}_i}(g_{\mathcal{N}_i}) = (I \otimes (I - \Phi)^{-1} G) g_{\mathcal{N}_i} \quad (8)$$

are the steady state of agent i and neighbors of agent i, respectively. Note that the constants $\delta > 0$ and $k_0 < \infty$ can be determined as in [22].

The main idea of the turn-based scheme is to partition the set of agents in subsets $\mathcal{T}_1, ..., \mathcal{T}_\chi$ such that two agents in the same subset are not neighbors in the interaction graph, i.e., they do not share any constraints

$$\forall i, j \in \mathcal{T}_k \Rightarrow j \notin \mathcal{N}_i. \quad (9)$$

Each subset $\mathcal{T}_k$ is hereafter referred to as a *turn*. The determination of a set of turns is closely related to the well known problem of graph coloring [23].

Let us assume that at each time step t, each agent i knows $x_{\mathcal{N}_i}(t)$ and $g_{\mathcal{N}_i}(t-1)$, and assume that a certain turn $\mathcal{T}_k(t)$ is in charge. Then, if all the agents $i \notin \mathcal{T}_k(t)$ are instructed to keep their previous command, i.e., $g_i(t) = g_i(t-1)$, then the only variable in (7) for each agent $i \in \mathcal{T}_k(t)$ will be $g_i(t)$. This allows to formulate the following strategy:

Algorithm 1 Turn-Based Distributed Command Governor

REPEAT AT EACH TIME t
1. IF $i \in \mathcal{T}_k(t)$
   1.1 RECEIVE $x_{\mathcal{N}_i}(t)$ AND $g_{\mathcal{N}_i}(t-1)$, $\forall j \in \mathcal{I}_i$
   1.2 SOLVE

$$g_i(t) = \arg\min_{g_i} ||g_i - r_i(t)||^2_\Psi \quad (10)$$
$$\text{subject to (7)}$$

2 ELSE
   2.1 SET $g_i(t) = g_i(t-1)$
3 APPLY $g_i(t)$
4 SEND $g_i(t)$ AND $x_i(t)$ TO THE NEIGHBORS $\mathcal{N}_i$

where $\|z\|^2_\Psi$ denotes the quadratic form $z^T \Psi z$ ($\Psi = \Psi^T > 0$). It is then possible to prove that:

Theorem 1: Consider the stable system (3), the constraints (4), and the turn mechanism (9) that is assumed to be periodic, i.e., $\forall t > 0, \exists t_p$ such that $\cup_{i=0}^{t_p} \mathcal{T}(t+i) = \mathcal{V}$. If at time $t = 0$, there exists a $g(0)$ such that (7) is satisfied $\forall i \in \mathcal{V}$, then the following statements hold:

i) For each agent $i \in \mathcal{V}$, at each decision time t, the minimizer in (10) uniquely exists and can be obtained by locally solving a convex constrained optimization problem [20],
ii) The overall system acted by the agents implementing Algorithm 1 never violates the constraints (4) [20].
iii) As a non-cooperative distributed RHC scheme, the algorithm always converges to a Nash equilibrium [24].

B. Connectivity Constraints and Local Constraints

In this subsection, we characterize the constraints (4) of the aggregate system that are due to local constraints or connectivity constraints.

Typical local constraints (2), in addition to the equality constraint (3), for each agent include:

- Input saturation constraints: $\|u_i(x_i(t), g_i(t))\|_\infty \leq u_{max}$;
- Velocity constraints: $||v_i(t)||_2 \leq v_{max}$;
- Constraints on the admissible position: $p_i(t) \in \mathcal{P}_i$, with $\mathcal{P}_i$ a convex region.

A possible way to guarantee connectivity of the communication graph $\mathcal{G}_C(t)$ is to enforce that a specific spanning tree (or a connected spanning subgraph) exists from t onward. Let the interaction graph $\mathcal{G}_I(t)$ be a spanning tree of the communication graph $\mathcal{G}_C(t)$ at time step t. To enforce this spanning tree to exist from t onward, the following $N-1$ convex constraints must be satisfied for all future time

$$||p_i(t) - p_j(t)|| \leq R, \ \forall (i,j) \in \mathcal{E}_I(t). \quad (11)$$

Note that in this case

$$\mathcal{N}_i = \{j | \ (i,j) \in \mathcal{E}_I(t), j \neq i\} \quad (12)$$

where $\mathcal{E}_I(t)$ is the edge set of the spanning tree. Interestingly enough, under the topology induced by (11), a simple two-turns partition of the agents can always be built.

Lemma 2: For any spanning tree $\mathcal{G}$, if $\mathcal{N}_i$ is as in (12), then the vertex set $\mathcal{V}$ can be partitioned in two turns $\mathcal{T}_1$ and $\mathcal{T}_2$ ensuring (9).

Proof: The proof is constructive. Let us arbitrarily assume that vertex i belongs to $\mathcal{T}_1$. In any tree, there exists only a unique path between any two vertices i and j. For any vertex j, if the length of the path between i and j is even, we assign j to the set $\mathcal{T}_1$, otherwise to the set $\mathcal{T}_2$. Since no other paths exists, the proof is concluded. ■

All the local constraints and connectivity constraints can be formulated in the form (4), and then are tractable by the distributed CG. In particular, from Theorem 1, it directly follows that, 1) the local constraints are always satisfied, 2) the spanning tree always exists, and then the communication graph is always connected.

Depending on the choice of the spanning tree at time $t = 0$, the use of a fixed interaction graph can be quite conservative as it may prevent the network to get certain configurations. For instance, if at time $t = 0$ a star graph is chosen (see Fig. 1.b), the path graph (see Fig. 1.a) is prevented.

We can resolve this limitation by allowing the algorithm to switch between different interaction graphs (spanning trees). However, switching between two interaction graphs cannot be done abruptly, because the transition could violate the constraints in the subsequent time steps. Therefore, we need a switching policy.

A switch from the interaction graph $\mathcal{G}_I$ with associated constraints $c_j, j = 1, \ldots, n_c$ to the interaction graph $\mathcal{G}'_I$ with associated constraints $c'_j, j = 1, \ldots, n_{c'}$ is *admissible* if for all future time steps the optimization problem (10) subject to the new constraints will admit a solution. The following lemma gives us a sufficient condition for an admissible switch:

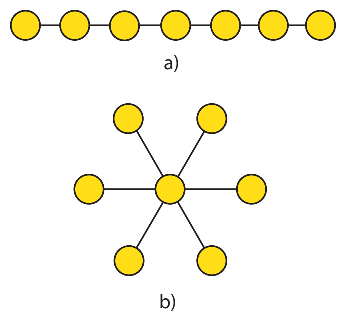

Fig. 1: Some spanning trees chosen for connectivity maintenance might be restrictive in terms of network configuration. a) a path graph, b) a star graph.

Lemma 3: Consider Algorithm 1 and assume $g(t)$ is the current applied command. A sufficient condition to guarantee that the switch to a new set of constraints $c'_j(t), j = 1, \ldots, n_{c'}$ is admissible from $t+1$ onward is that $g(t)$ simultaneously satisfies the inequalities (7) defined with respect to $c'_j(t)$ for all agents $i \in \mathcal{V}$.

Proof: To prove the statement it is enough to note that for all $\tau > t+1$, $g(\tau) = g(t)$ is an admissible solution as still satisfies (7) with the new constraints for all agents. ■

Clearly, when a switch is admissible, in order to apply the switch all agents must agree to change the constraint set to the new one at time $t+1$.

C. Dynamic Interaction Graph based on EMST

In this subsection, we propose to use a dynamic interaction graph defined by the Euclidian Minimum Spanning Tree (EMST) of the network.

Assume the weights in the communication graph $\mathcal{G}_C(t)$ are relative Euclidean distances between agents. The EMST $\mathcal{G}_{EMST} = (\mathcal{V}, \mathcal{E}_{EMST})$ is a spanning tree with minimum total weight, i.e.,

$$\sum_{(i,j)\in\mathcal{E}_{EMST}} w_{ij} \leq \sum_{(i,j)\in\mathcal{E}_{ST}} w_{ij} \tag{13}$$

where w_{ij} is the weight of the edge (i, j), and $\mathcal{E}_{ST}$ is the edge set of an arbitrary spanning tree $\mathcal{G}_{ST}$.

Definition 2 (Connectivity set of an agent): Associated with a connected spanning subgraph used for ensuring connectivity of a proximity network $\mathcal{G}$, the connectivity set of an agent is a set of points that the agent can stay on given the positions of its neighbors without removing the edges of the subgraph.

In the case of spanning trees, for each agent i this set is determined by the intersection of all the constraint sets given by (11) for $j \in \mathcal{N}_i$. Fig. 2 illustrates the EMST of a given proximity graph, and it represents the connectivity sets of a vertex imposed by the EMST and another spanning tree.

Lemma 4: Among all the connected spanning subgraphs used for ensuring connectivity of a proximity network $\mathcal{G}$, $\mathcal{G}_{EMST}$ provides the largest connectivity set on average for every agents.

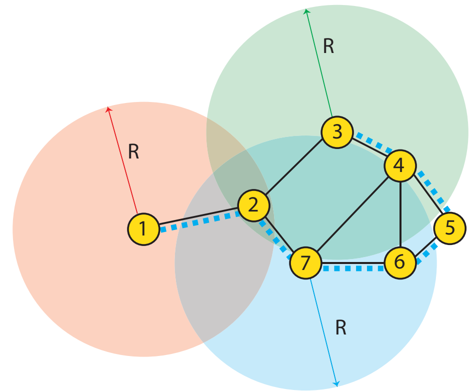

Fig. 2: The Euclidean minimum spanning tree (EMST) of a given proximity network shown by the dashed blue line. The connectivity sets of vertex 2 imposed by the EMST and another spanning tree are depicted by the intersections of circles with radius R, i.e., intersections of the red and blue circles, and the green and red circles, respectively

Proof: Assume that a connected spanning subgraph $\mathcal{G}_S$ exists which provides the largest connectivity set on average for every agent, and it is not the EMST. As $\mathcal{G}_S$ imposes the least conservative constraints for ensuring the connectivity, it must have the minimum number of edges. Then, $\mathcal{G}_S$ is a spanning tree. In addition, since $\mathcal{G}_S$ provides on average the largest set for every agent, the average distance between every two neighbors must be the shortest one, i.e.,

$$\bar{w}(\mathcal{G}_S) \leq \bar{w}(\mathcal{G}_{ST}). \tag{14}$$

Multiplying both sides by the number of edges in a spanning tree, we have

$$(N-1)\bar{w}(\mathcal{G}_S) \leq (N-1)\bar{w}(\mathcal{G}_{ST}).$$

Each side of (16) represents the total weight of the corresponding graph, i.e.,

$$\sum_{(i,j)\in\mathcal{E}_S} w_{ij} \leq \sum_{(i,j)\in\mathcal{E}_{ST}} w_{ij}. \tag{15}$$

Hence, $\mathcal{G}_S$ must be the EMST, and this is a contradiction. ■

Another advantage of using EMST is the existence of efficient distributed algorithms that can obtain it. In this work, we employ the Gallager, Humblet, and Spira (GHS) algorithm [25] which is reported briefly in Algorithm 2.

Fig. 3 illustrates a proximity graph (communication graph) with 50 agents and the EMST (interaction graph) with (vertex) graph coloring of the EMST.

We update the EMST every τ time steps in order to make the interaction graph adaptive. Hence, the switching times will be at $t = k\tau, k \in \mathbb{N}$.

Proposition 1: Provided the switch from the interaction graph at time $t-1$, which is the EMST at time $(k-1)\tau$, to the EMST at time $t = k\tau$ is admissible for every agent (in the sense of Lemma 3), then the network can update the

Algorithm 2 Gallager, Humblet, and Spira

1. EXECUTE $WakeUp$
2. RESPOND TO (FIRST COME FIRST SERVED)
   - $Connect(L)$ ON EDGE e
   - $Initiate(L, F, S)$ ON EDGE e
   - $Test(L, F)$ ON EDGE e
   - $Accept$ ON EDGE e
   - $Reject$ ON EDGE e
   - $Report(w)$ ON EDGE e
   - $ChangeRoot$

interaction graph at time t, and the connectivity constraints for every agent through the time interval $\big[k\tau, (k+1)\tau\big)$ is

$$\|p_i(t) - p_j(t)\| \leq R,\ \forall j \in \mathcal{N}_{i,EMST}(k\tau). \quad (16)$$

Notice that if the switching to the new topology is not admissible at time $t = k\tau$, one solution is to postpone the switch for another τ steps, and then recheck the admissibility.

Remark 1: Switching between spanning trees is devised to enhance the performance of the overall system when the connectivity control algorithm is integrated with a high-level control algorithm (e.g., formation control). When the system reaches an equilibrium (velocity mismatch of the network becomes zero), one can switch from the EMST to any other spanning tree available for the network if it further improves the performance.

IV. Simulation Results

To show the effectiveness of the proposed algorithm, we present simulation results with a swarm of 20 agents. We assume each agent has a double integrator dynamics and is controlled with a PD

$$\Phi = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ -T_s k_p & 0 & 1 - T_s k_d & 0 \\ 0 & -T_s k_p & 0 & 1 - T_s k_d \end{bmatrix},\ G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ T_s k_p & 0 \\ 0 & T_s k_p \end{bmatrix} \quad (17)$$

where $T_s = 0.1$, $k_p = 12$, and $k_d = 5$, and the eigenvalues of the controlled systems are $0.75 \pm 0.24i$ and $0.75 \pm 0.24i$. The parameters in (7) are $\delta = 0.1$ and $k_0 = 10$.

We assume that the local constraint is defined as

$$\begin{bmatrix} -2 \\ -2 \end{bmatrix} \leq v_i(t) \leq \begin{bmatrix} 2 \\ 2 \end{bmatrix}. \quad (18)$$

At time $t = 0$, the agents are distributed in a region $[2\text{ m} \times 2\text{ m}]$. The communication distance is $R = 2$ m, and we assume that the communication graph is initially connected.

The scenario is to deploy all agents on a circle with the radius of 3 m and center at origin. Each agent is provided with its final desired position on the circle. The swarm should expand itself while the connectivity constraints and the local constraints must be satisfied. Figure 4(a-e) illustrate the configuration of the swarm at difference time steps. It was observed that all the local constraints were satisfied. The algebraic connectivity through the experiment is shown in Fig. 4(f). The algebraic connectivity starts from 20 and

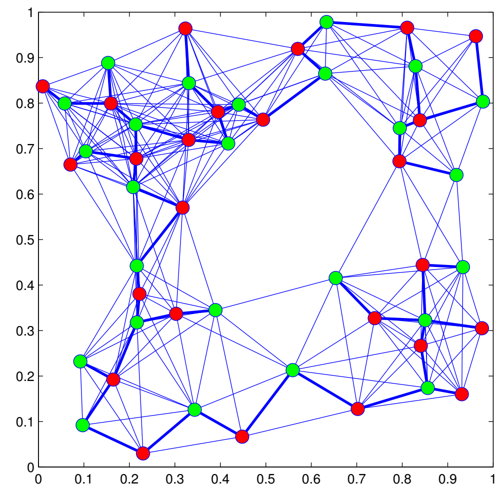

Fig. 3: A proximity graph with 50 agents and with neighborhood distance 0.30 randomly generated in a $[1 \times 1]$ region. The thin lines illustrates the edges in the proximity graph while the tick lines illustrates the edges in the EMST. The EMST is colored by two colors: green and red.

reaches 0.48 at the steady state. The minimum value of the algebraic connectivity is 0.35 occurring at the time step 16 confirming that the communication graph was maintained connected through the deployment.

V. Conclusion

In this paper, the problem of preserving connectivity for dynamically decoupled agents with linear discrete-time dynamics subject to local constraints was tackled, and a novel distributed connectivity control algorithm making use of a receding horizon scheme was proposed. The results showed the effectiveness of the method. In the future, we intend to study the convergence properties of the scheme and its extension to the case of nonlinear agents.

Acknowledgment

This work was partially supported by the European Research Council through the ERC Advanced Grant "E-SWARM: Engineering Swarm Intelligence Systems" (contract 246939). Marco Dorigo acknowledges support from the "Fonds de la Recherche Scientifique - FNRS" of the Fédération Wallonie Bruxelles, of which he is a research director.

References

[1] M. Mesbahi, "On State-dependent dynamic graphs and their controllability properties," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 387–392, 2005.

[2] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[3] 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.

[4] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

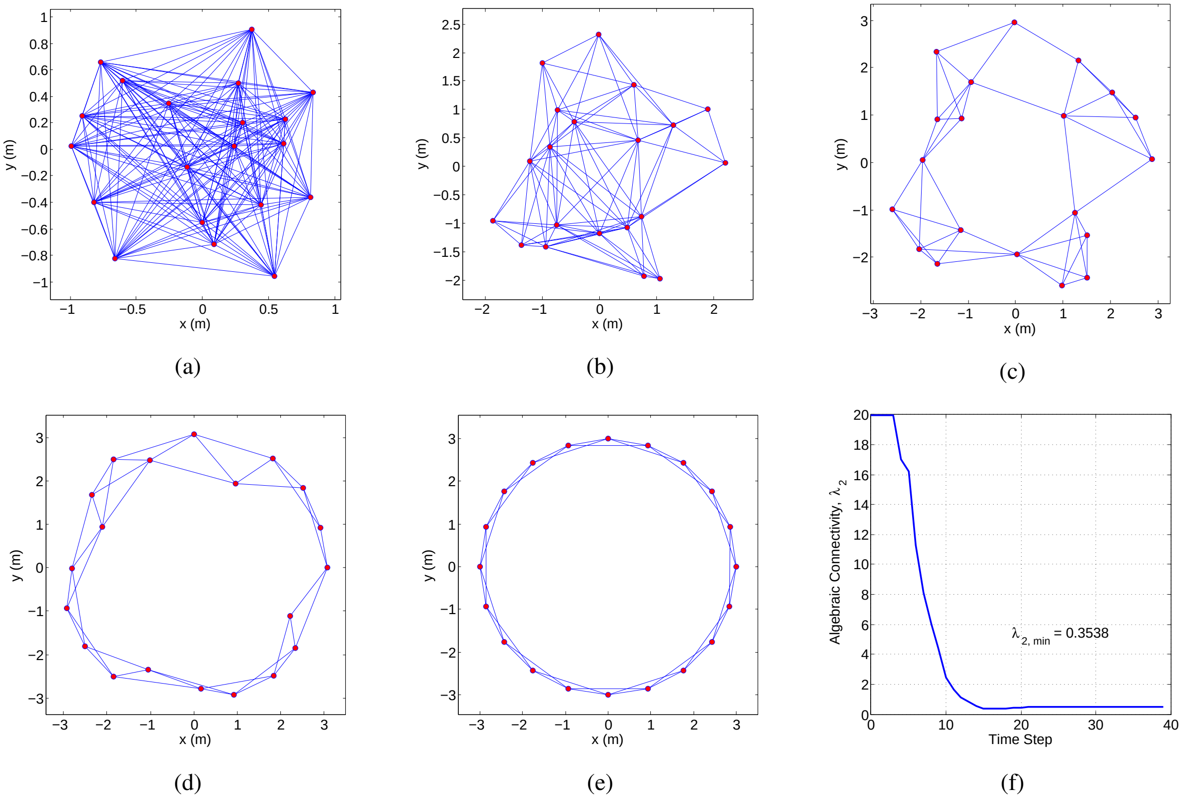

Fig. 4: (a-e) Configurations of the swarm at the time steps 0, 10, 14, 19, and 39 respectively. (f) The algebraic connectivity through the experiment with its minimum occurring at the time step 16.

[5] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo, "Distributed algorithms for environment partitioning in mobile robotic networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 8, pp. 1834–1848, 2011.

[6] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, 2006.

[7] M. De Gennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *IEEE Conference on Decision and Control (CDC)*, pp. 3628–3633, Dec 2006.

[8] M. M. Zavlanos and G. J. Pappas, "Controlling connectivity of dynamic graphs," in *IEEE Conference on Decision and Control (CDC)*, pp. 6388–6393, 2005.

[9] M. M. Zavlanos and G. J. Pappas, "Potential fields for maintaining connectivity of mobile networks," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 812–816, 2007.

[10] D. Dimarogonas and K. Johansson, "Decentralized connectivity maintenance in mobile networks with bounded inputs," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1507–1512, May 2008.

[11] A. Ajorlou, A. Momeni, and A. G. Aghdam, "A Class of bounded distributed control strategies for connectivity preservation in multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2828–2833, 2010.

[12] M. M. Zavlanos and G. J. Pappas, "Distributed connectivity control of mobile networks," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1416–1428.

[13] R. Aragues, C. Sagues, and Y. Mezouar, "Triggered minimum spanning tree for distributed coverage with connectivity maintenance," in *European Control Conference (ECC)*, pp. 1881–1887, June 2014.

[14] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1525–1540, 2011.

[15] M. M. Zavlanos, H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Hybrid control for connectivity preserving flocking," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2869–2875, 2009.

[16] K. Savla, G. Notarstefano, and F. Bullo, "Maintaining limited-range connectivity among second-order agents," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 187–205, 2009.

[17] A. Simonetto, T. Keviczky, and R. Babuška, "Constrained distributed algebraic connectivity maximization in robotic networks," *Automatica*, vol. 49, no. 5, pp. 1348–1357, 2013.

[18] A. Ajorlou and A. Aghdam, "Connectivity preservation in nonholonomic multi-agent systems: A bounded distributed control strategy," *IEEE Transactions on Automatic Control*, vol. 58, pp. 2366–2371, Sept 2013.

[19] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. J. Balas, "Decentralized receding horizon control and coordination of autonomous vehicle formations," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 19–33.

[20] A. Casavola, E. Garone, and F. Tedesco, "Scalability and performance improvement of distributed sequential command governor strategies via graph colorability theory," in *IFAC World Congress*, pp. 9400–9405, 2014.

[21] C. D. Godsil, G. Royle, and C. Godsil, *Algebraic graph theory*, vol. 207. Springer New York, 2001.

[22] A. Bemporad, A. Casavola, and E. Mosca, "Nonlinear control of constrained linear systems via predictive reference management," *IEEE Transactions on Automatic Control*, vol. 42, no. 3, pp. 340–349, 1997.

[23] T. R. Jensen and B. Toft, *Graph coloring problems*. John Wiley & Sons, 2010.

[24] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, "Distributed MPC strategies with application to power system automatic generation control," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1192–1206, 2008.

[25] R. G. Gallager, P. A. Humblet, and P. M. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Transactions on Programming Languages and systems (TOPLAS)*, vol. 5, no. 1, pp. 66–77, 1983.