2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

On Distributed Maximization of Algebraic Connectivity in Robotic Networks

Andrea Simonetto, Tamás Keviczky, and Robert Babuška

Abstract— **We consider the problem of maximizing the algebraic connectivity of the communication graph in a network of mobile robots by moving them into appropriate positions. We describe the Laplacian of the graph as dependent on the pairwise distance between the robots and formulate an approximate problem as a Semi-Definite Program (SDP). We propose a consistent, non-iterative distributed solution by solving local SDP's which use information only from nearby neighboring robots. Numerical simulations show the performance of the algorithm with respect to the centralized solution.**

I. INTRODUCTION

A robotic network is a collection of mobile units that communicate with one another to achieve a common goal. Such systems are present in several applications ranging from underwater [1] and space exploration [2], to search and rescue [3], fire monitoring [4] and other surveillance applications [5]. Maintaining connectivity between the individual robots and guaranteeing a certain level of communication quality given the environmental constraints and objectives have been considered as key requirements to meet the demands of such applications. Furthermore, several different coordination and control algorithms designed for these networks rely on some type of agreement protocol or consensus process [6]–[8], whose effectiveness is profoundly influenced by the interconnections between the units.

Motivated by the importance of the communication network, we study distributed solutions for maximizing its algebraic connectivity (often denoted as λ_2) in mobile robotic networks. This parameter is the second smallest eigenvalue of the communication graph's Laplacian matrix, and it dictates the convergence properties of consensus protocols [9], [10]. Maximization of λ_2 is also crucial for collaborative target tracking [11], where a network of mobile robots strive for increased accuracy of the joint position estimate of one or more moving objects [12]–[14]. Besides an increase in accuracy, a positive λ_2 also ensures that the network stays connected during the motion. We will focus on distance-based connectivity aspects of robotic networks, as opposed to ensuring line-of-sight connectivity in the presence of obstacles, which is also a relevant related problem actively studied in the literature (see e.g., [15]).

The works of [16]–[21] give a comprehensive overview of distributed algorithms for networks of vehicles that aim at ensuring connectivity (i.e., nonzero λ_2 rather than its maximization). Typically, these are either limited to specific scenarios only, or imply heavy communication requirements. Often the proposed approaches are not obtained by decomposing the centralized problem but constructed using ad-hoc methods that are not directly related to the solution of the centralized version.

In terms of distributed connectivity maximization, the available literature appears very limited. To the best of our knowledge, only the work of [22] investigates a distributed solution for the maximization of λ_2. The authors use a two-step distributed algorithm, which relies on super-gradients and potential functions. The required communication load scales with the square of the graph diameter and can impede fast real-time implementations for large groups of agents.

In this paper, we present a distributed approach for the λ_2 maximization problem as formulated by [11], [23], [24] in a centralized framework. Our perspective is model-based optimization and control, which allows additional constraints (e.g., the dynamics of the robots) to be included explicitly in the problem formulation. Moreover, our approach is consistent, in the sense that *(i)* the local problems are derived via a suitable decomposition of the centralized one, *(ii)* the *linearized* algebraic connectivity of the approximate problem is guaranteed to be monotonically increasing, and *(iii)* the local solutions are feasible with respect to the constraints of the original centralized problem. These properties, especially *(i)* and *(iii)*, appear to be completely absent in the aforementioned literature. Our proposed distributed approach relies on local problems that are solved by each robot using information only from nearby neighbors and, in contrast with [22], it does not require any iterative schemes, making it more suitable for real-time applications. This last property is not a trivial aspect when using common decomposition methods [25], as done in various approaches to distributed control [26], [27]. Finally, the proposed solution can also be extended to incorporate other interesting scenarios, such as collaborative target tracking.

Simulation results support the efficacy of our approach and show interesting properties of the algorithm. For instance, given the nonlinear/nonconvex nature of the problem, in certain scenarios the distributed solutions converge to a higher λ_2 value than the centralized ones obtained from the approximate problem formulation.

The paper's main contribution is to improve and extend our preliminary heuristic approaches presented in [28], and provide proofs of its most important properties. Furthermore, we discuss new insights gained from simulation studies, along with suggestions for further improvements.

The paper is organized as follows. Section II formulates

The authors are with the Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands {a.simonetto, t.keviczky, r.babuska}@tudelft.nl

the approximate centralized problem as suggested by [11], [23], [24]. The proposed distributed approach is described in Section III. Numerical simulations are shown in Section IV to assess the performance of the distributed solutions with respect to centralized schemes. Conclusions and open issues are discussed in Section V.

II. Problem formulation

We consider a network of N agents. The agents represent mobile robots and the network encodes undirected communication links, meaning that if two agents are connected, they can communicate with each other. As a general notation $a_i(k)$ represents the value of the variable a for agent i at time k while $\delta a_i(k) = a_i(k) - a_i(k-1)$. For simplicity of exposition and without loss of generality, we will assume identical discrete-time agent dynamics of the following form:

$$x_i(k) = x_i(k-1) + v_i(k-1)\Delta t \tag{1}$$

where the agents are limited to move on a 2-D plane, $v_i(k)$ is the velocity control input and Δt the sampling time. Let $x(k) \in \mathbb{R}^{nN}$ be the collection of the agents' positions in a 2-D space, i.e., $x(k) = (x_1^\top(k), \ldots, x_N^\top(k))^\top$. We use graph-theoretical notions to model the network. The set $\mathcal{S}$ contains the indices of the mobile agents (nodes), with cardinality $N = |\mathcal{S}|$. We use $\mathcal{E}$ to indicate the set of communication links, i.e., the edges $\{(i,j)|i,j \in \mathcal{S}\}$. The graph $\mathcal{G}$ is then expressed as $\mathcal{G} = (\mathcal{S}, \mathcal{E})$. Let the graph be connected initially, the agent clocks synchronized, and assume perfect communication (no delays or packet losses). The agents with which agent i communicates are called neighbors and are contained in the set $\mathcal{N}_i$. Note that node i is not included in the set $\mathcal{N}_i$. We define $\mathcal{J}_i = \mathcal{N}_i \cup \{i\}$ and $N_i = |\mathcal{J}_i|$. We define a set of Laplacian matrices $\mathcal{L}$ associated with $\mathcal{G}$ as

$$\mathcal{L} = \{L \in \mathbb{R}^{N\times N} | L = L^\top, \ell_{ij} = 0 \text{ iff } (i,j) \notin \mathcal{E}, L\mathbf{1} = \mathbf{0}\}$$

The entries of a Laplacian matrix L are defined as

$$\ell_{ij} := \begin{cases} 0 & (i,j) \notin \mathcal{E} \\ -w_{ij} & (i,j) \in \mathcal{E}, i \neq j \\ \sum_{l \neq i} w_{il} & i = j \end{cases} \tag{2}$$

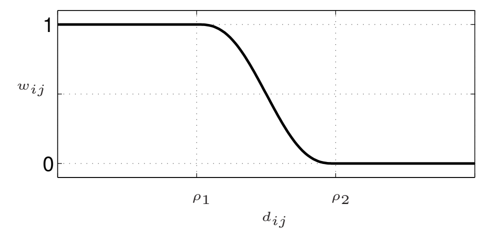
where the positive weights w_{ij} represent the "connection strength" between agent i and j. The weights themselves depend on the physical distance between the agents. For this purpose we introduce the square distance matrix D, whose entries d_{ij} are defined as

$$d_{ij}(k) = ||x_i(k) - x_j(k)||^2 \tag{3}$$

The value of the normalized weights w_{ij} will be 1 representing a "strong connection" if d_{ij} is less than a certain threshold, i.e., $d_{ij} \leq \rho_1$, with $\rho_1 > 0$. On the other hand, agents will not be connected at all ($w_{ij} = 0$) for $d_{ij} > \rho_2$, with $\rho_2 > \rho_1$. For $\rho_1 < d_{ij} \leq \rho_2$ the agents are connected with a connection strength that decreases smoothly with their distance. Some of the typical functions used for the weights w_{ij} can be found in [11], [23], [28], for simulation purposes we will use the polynomial description shown in Table I, which for a suitable choice of the coefficients α_p is both continuous and twice-differentiable.

TABLE I

5-TH ORDER POLYNOMIAL CHOICE FOR THE WEIGHTING FUNCTION.



$$w_{ij} := \begin{cases} 1 & d_{ij} < \rho_1 \\ \sum_{p=0}^{5} \alpha_p d_{ij}^p & \rho_1 \leq d_{ij} < \rho_2 \\ 0 & d_{ij} \geq \rho_2 \end{cases}$$

As a direct consequence of the above definitions, the entries of the Laplacian matrix (2) will depend on the pairwise distance and therefore on the position of the agents, making it state-dependent, which we will denote by $L(x)$. We are interested in the maximization of the algebraic connectivity of the weighted graph by moving the robots to appropriate positions. This goal could be formulated as the following time-invariant optimization problem [24]:

$$\mathbf{P}(L(x)) : \quad \max_{x,\gamma} \quad \gamma \tag{4a}$$
$$\text{s.t.} \quad \gamma > 0 \tag{4b}$$
$$L(x) + \mathbf{1}\mathbf{1}^T \succ \gamma I \tag{4c}$$

where the decision variables are the final robot locations and the optimal value of γ which is the maximum λ_2 for $L(x)$.

This problem would be convex if L was the decision variable, but it is non-convex given that we are optimizing over the positions x and the entries of L are nonlinear functions of x. However, we can obtain a time-varying convex approximation of the problem by using first-order Taylor approximations and the dynamical equation (1):

$$d_{ij}(k) = d_{ij}(k-1) + 2(x_i(k-1) - x_j(k-1))^\top (\delta x_i(k) - \delta x_j(k))$$

therefore

$$d_{ij}(k) = -d_{ij}(k-1) + 2(x_i(k-1) - x_j(k-1))^\top (x_i(k) - x_j(k))$$

and in the same way, the weights of the state-dependent Laplacian $L(x)$ are approximated as

$$\begin{aligned} w_{ij}(k) = &\; w_{ij}(k-1) + \left.\frac{\partial w_{ij}}{\partial d_{ij}}\right|_{d_{ij}(k-1)} \left(\frac{\partial d_{ij}}{\partial x_i}\delta x_i(k) + \frac{\partial d_{ij}}{\partial x_j}\delta x_j(k)\right) \\ = &\; w_{ij}(k-1) + 2\left.\frac{\partial w_{ij}}{\partial d_{ij}}\right|_{d_{ij}(k-1)} \times \\ &\; (x_i(k-1) - x_j(k-1))^\top (\delta x_i(k) - \delta x_j(k)) \end{aligned}$$

This allows us to consider the maximization of the algebraic connectivity of L as the following convex optimization problem to be solved at each time step k:

$$\mathbf{P}_k\left(L(x), x(k-1), D(k-1), \rho_1, v_{\max}\right):$$

$$\max_{x(k), D(k), \gamma(k)} \quad \gamma(k) \tag{5a}$$

s.t.

$$\mathcal{Q}_1 : \begin{cases} \gamma(k) > 0 \\ L(x(k)) + \mathbf{1}\mathbf{1}^T \succ \gamma(k) I \end{cases} \tag{5b}$$

$$\mathcal{Q}_2 : \begin{cases} \mathcal{Q}_{2.1}: & d_{ij}(k) + d_{ij}(k-1) - 2(x_i(k-1) - x_j(k-1))^\top \times \\ & \quad (x_i(k) - x_j(k)) = 0 \\ \mathcal{Q}_{2.2}: & d_{ij}(k) > \rho_1, \quad \forall (i,j) \in \mathcal{E} \\ \mathcal{Q}_{2.3}: & ||x_i(k) - x_i(k-1)|| \le v_{\max}\Delta t \quad i = 1, \dots, N \end{cases} \tag{5c}$$

where the constraint $\mathcal{Q}_{2.2}$ is used both to enforce positive distance values and to avoid agents getting too close to each other. This is not ensured by $\mathcal{Q}_{2.1}$ alone, since there $d_{ij}(k)$ is not constrained to be ≥ 0. The constraint $\mathcal{Q}_{2.3}$ on the velocity represents the physical limitations of the agents.

The optimization problem that has been described in this section attempts to solve the connectivity maximization problem in a centralized manner using linearization and an optimization problem at each time k. In realistic application scenarios, computing the desired positions and the corresponding motion commands for the robots cannot be performed in a single centralized location due to computational and communication constraints. In the next section, we describe a solution approach that allows the problem to be solved in a distributed fashion, using local computation and limited communication resources, which increases the flexibility of the robotic network and is thus appealing in practice.

III. The Proposed Distributed Solution

In this section we present a non-iterative distributed and consistent solution to solve (5). We note that this is not a trivial achievement since the most common decomposition methods, such as Jacobi algorithms [25], Primal decomposition [26], Dual decomposition [27], and Augmented Lagrangian Method [25], typically require iterative solutions which may not be amenable to fast real-time implementations. We are currently exploring the use of recent developments in parallel and distributed algorithms to decompose and solve SDPs [29], [30], which may lead to applicable alternative approaches.

Before presenting the main contribution of this paper, we first introduce some notation and definitions. We then proceed to describe our non-iterative distributed solution method and argue why it leads to a consistent algorithm. As described before, this means that solving local problems leads to a monotonically increasing λ_2 of the linearized Laplacian of the entire network.

In order to describe the local problems each agent will be solving, we define subgraphs that correspond to the agents and their neighborhood. Let $\mathcal{M}_i$ denote the *enlarged* neighborhood for each agent i defined as

$$\mathcal{M}_i = \bigcup_{l \in \mathcal{J}_i} \mathcal{J}_l, \qquad i = 1, \dots, N \tag{6}$$

whose cardinality will be M_i. We denote the vector containing all the positions of the agents in this set with $x_{\mathcal{M}_i}$. We define

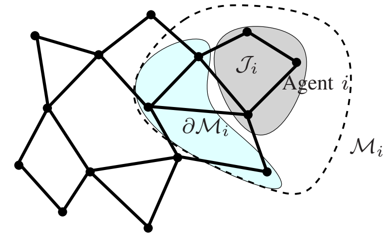


Fig. 1. Notation for the distributed solution

$$\partial\mathcal{M}_i = \{l | l \in \mathcal{M}_i, l \notin \mathcal{J}_i\}, \qquad i = 1, \dots, N \tag{7}$$

as the bordering set of $\mathcal{M}_i$, while we call the set of agents belonging to $\partial\mathcal{M}_i$, the bordering agents of $\mathcal{M}_i$. Figure 1 provides a graphical illustration of this notation. An illustrative example and arguments justifying the choice and role of the enlarged neighborhood set is given in our earlier work [28].

Finally, we will denote the graph Laplacian associated with subgraph $\mathcal{M}_i$ as L_i with corresponding distance matrix D_i. We also introduce a scaled minimum distance $\tilde{\rho_1}_{ij}$ and a scaled maximum velocity $\tilde{v}_{\max,i}$ defined as

$$\tilde{\rho_1}_{ij} = \left(\sum_{p \in \mathcal{M}_i \cap \mathcal{M}_j} \frac{1}{N_p} \right)^{-1} \times \left(\rho_1 + d_{ij}(k-1) \left(\sum_{p \in \mathcal{M}_i \cap \mathcal{M}_j} \frac{1}{N_p} - 1 \right) \right), \quad \forall (i,j) \in \mathcal{E} \tag{8}$$

$$\tilde{v}_{\max,i} = \left(\sum_{j \in \mathcal{M}_i} \frac{1}{N_j} \right)^{-1} v_{\max}, \quad i = 1, \dots, N \tag{9}$$

whose values vary from agent to agent. These quantities will be used to tighten the global constraints in such a way that the global solution constructed from the local ones satisfies the original $\mathcal{Q}_{2.2}$ and $\mathcal{Q}_{2.3}$ constraints.

Our algorithm consists of two steps. First, each agent solves the problem $\mathbf{P}_{k,i}$ defined as

$$\mathbf{P}_k(L_i(x_{\mathcal{M}_i}), x_{\mathcal{M}_i}(k-1), D_i(k-1), \tilde{\rho_1}_{ij}, \tilde{v}_{\max,i}) \tag{10a}$$

$$\text{s.t.} \quad \mathcal{Q}_3: \quad x_j(k) = x_j(k-1), \qquad \text{for } j \in \partial\mathcal{M}_i \tag{10b}$$

computing the solution $\hat{x}_{\mathcal{M}_i}(k)$, which is composed of $\hat{x}_{ij}(k)$ for each $j \in \mathcal{M}_i$. Thus, we will call $\hat{x}_{ij}(k)$ the position of agent j as computed by agent i. Note that the extra constraint $\mathcal{Q}_3$ is an important requirement to guarantee consistency as will be explained later in this section.

As the second step, the solutions $\hat{x}_{\mathcal{M}_i}(k)$ are shared within the enlarged neighborhood $\mathcal{M}_i$ and averaged according to

$$x_i(k) = x_i(k-1) + \sum_{j \in \mathcal{M}_i} \frac{1}{N_j} \delta\hat{x}_{ji}(k), \qquad i = 1, \dots, N \tag{11}$$

Algorithm 1 summarizes the method.

Remark 1: We emphasize again that although $\mathcal{Q}_3$ keeps some agent positions fixed in the local solutions, they will not remain stationary when constructing the global solution due to the averaging step (11).

Algorithm 1 λ_2 Maximization.

1: Input: $x_i(k-1)$, $x_j(k-1)$, $j \in \mathcal{M}_i$
2: Compute: $d_{ij}(k-1)$ from input based on (3)
3: Solve: $\mathbf{P}_{k,i}$ in (10) computing $\hat{x}_{ij}(k)$, $j \in \mathcal{M}_i$
4: Communicate: $\hat{x}_{ij}(k)$ among members of $\mathcal{M}_i$
5: Average: $x_i(k) = x_i(k-1) + \sum_{j \in \mathcal{M}_i} \frac{1}{N_j} \delta\hat{x}_{ji}(k)$
6: Output: $x_i(k)$

We claim that the above algorithm leads to a consistent solution, i.e., if we consider the resulting global position vector $x(k) = (x_1^\top(k), \dots, x_N^\top(k))^\top$, then

(C1) the algebraic connectivity of the corresponding global linearized Laplacian $L(x(k))$ is monotonically increasing in each iteration;

(C2) all the constraints of the global problem are met.

We will prove these claims in two steps: Theorem 1 establishes (C1), by establishing a link between the average value (11) and the algebraic connectivity through the linear dependence of the linearized Laplacian on x. The constraint $\mathcal{Q}_3$ plays a crucial role here to ensure the feasibility of the local solutions. Theorem 2 guarantees (C2), by showing how the tightened constraints (8)-(9) of the local problems ensure that the global solution, obtained via the average (11), satisfies the global constraints.

Consider the local problem $\mathbf{P}_{k,i}$ and its solution comprised of $\hat{x}_{ij}(k)$ for all $j \in \mathcal{M}_i$. Construct the vector

$$\hat{x}^{(i)}(k) = (x_1^\top(k-1), \dots, \hat{x}_{ij}^\top(k), \dots, x_N^\top(k-1))^\top \tag{12}$$

where we keep those agent positions that have not been optimized fixed, and we update the rest from the solution of the local problem.

Theorem 1: (C1) The algebraic connectivity of the global linearized Laplacian $L(x(k))$ is monotonically increasing in each iteration, meaning $L(x(k)) \succeq L(x(k-1))$, where $x(k)$ is computed by the average (11).

Proof. Due to constraint $\mathcal{Q}_3$, $L(\hat{x}^{(i)}(k)) - L(x(k-1)) \succeq 0$, meaning that the new positions $\hat{x}^{(i)}(k)$ do not decrease the algebraic connectivity of the Laplacian matrix. In fact, $L(\hat{x}^{(i)}(k)) - L(x(k-1) = L(\delta\hat{x}^{(i)}(k)))$, which, up to a renumbering of the agents, leads to:

$$\left[\begin{array}{c|c} L(\delta\hat{x}_{\mathcal{M}_i}(k)) & 0 \\ \hline 0 & 0 \end{array}\right] \succeq 0$$

which is positive semi-definite due to the nature of the local solution.

The previous property implies $(L(\hat{x}^{(i)}(k)) - L(x(k-1)))/N_i \succeq 0$ for all i. Thus summing over all agents leads to

$$\sum_{i=1}^{N} \frac{1}{N_i}(L(\hat{x}^{(i)}(k)) - L(x(k-1))) \succeq 0$$

Considering the weighted sum $x_i(k)$ in (11), and the associated global vector $x(k)$, it can be shown that

$$L(\delta x(k)) = \sum_{i=1}^{N} \frac{1}{N_i} L(\delta\hat{x}^{(i)}(k)) \tag{13}$$

which leads to the desired consistency property

$$L(x(k)) \succeq L(x(k-1))$$

To show the correctness of (13), let us consider the entry (i,j) of the Laplacian L on both sides of the expression. For the right side, ℓ_{ij} is

$$\ell_{ij} = a_{ij} \sum_{p \in \mathcal{M}_i \cap \mathcal{M}_j} \frac{\delta\hat{x}_{pi}(k) - \delta\hat{x}_{pj}(k)}{N_p}$$

where a_{ij} is a constant depending on $x(k-1)$.

For the left side,

$$\ell_{ij} = a_{ij}\,(\delta x_i(k) - \delta x_j(k)) = a_{ij}\left(\sum_{p \in \mathcal{M}_i} \frac{1}{N_p}\delta\hat{x}_{pi}(k) - \sum_{p \in \mathcal{M}_j} \frac{1}{N_p}\delta\hat{x}_{pj}(k)\right)$$

the last expression can be divided in three parts: $p \in \mathcal{M}_i \cap \mathcal{M}_j$, $p \in \mathcal{M}_i \wedge p \notin \mathcal{M}_j$, and $p \in \mathcal{M}_j \wedge p \notin \mathcal{M}_i$. Since $a_{ij} \neq 0$ only if (i,j) are first-order neighbors, we make the key observation that: $\{p | p \in \mathcal{M}_i \wedge p \notin \mathcal{M}_j\} \subseteq \partial\mathcal{M}_i$ and $\{p | p \in \mathcal{M}_j \wedge p \notin \mathcal{M}_i\} \subseteq \partial\mathcal{M}_j$ which leads to:

$$\ell_{ij} = a_{ij} \sum_{p \in \mathcal{M}_i \cap \mathcal{M}_j} \frac{\delta\hat{x}_{pi}(k) - \delta\hat{x}_{pj}(k)}{N_p} + \underbrace{a_{ij} \sum_{p \in \mathcal{M}_i \wedge p \notin \mathcal{M}_j} \frac{\delta\hat{x}_{pi}(k)}{N_p}}_{=0} - \underbrace{a_{ij} \sum_{p \in \mathcal{M}_j \wedge p \notin \mathcal{M}_i} \frac{\delta\hat{x}_{pj}(k)}{N_p}}_{=0}$$

where the last two terms are 0 due to the constraint set $\mathcal{Q}_3$. Since both sides of the expression (13) are the same, then the consistency property (C1) holds. □

Theorem 2: (C2) The global vector $x(k)$ computed with the average (11), satisfies the constraint sets $\mathcal{Q}_1$ and $\mathcal{Q}_2$.

Proof. The previous proof guarantees that the averaged solution satisfies the constraint $\mathcal{Q}_1$. With regards to $\mathcal{Q}_2$, we begin with the constraints $\mathcal{Q}_{2.1}$ and $\mathcal{Q}_{2.2}$. In subproblem $\mathbf{P}_{k,p}$ with $p \in \mathcal{M}_i \cap \mathcal{M}_j$ the distance variable is written as

$$d_{ij}(k) = d_{ij}(k-1) + 2(x_i(k-1) - x_j(k-1))^\top \times (\delta\hat{x}_{pi}(k) - \delta\hat{x}_{pj}(k)) > \tilde{\rho_1}_{ij}$$

or

$$2(x_i(k-1) - x_j(k-1))^\top(\delta\hat{x}_{pi}(k) - \delta\hat{x}_{pj}(k)) > \tilde{\rho_1}_{ij} - d_{ij}(k-1).$$

When considering the use of the average $x(k)$, following similar arguments as in the previous proof leads to

$$\begin{aligned} & d_{ij}(k-1) + 2(x_i(k-1) - x_j(k-1))^\top \times \sum_{p \in \mathcal{M}_i \cap \mathcal{M}_j} \frac{1}{N_p}(\delta\hat{x}_{pi}(k) - \delta\hat{x}_{pj}(k)) > \\ & d_{ij}(k-1) + (\tilde{\rho_1}_{ij} - d_{ij}(k-1)) \sum_{p \in \mathcal{M}_i \cap \mathcal{M}_j} \frac{1}{N_p} > \\ & \sum_{p \in \mathcal{M}_i \cap \mathcal{M}_j} \frac{1}{N_p}\tilde{\rho_1}_{ij} - d_{ij}(k-1)\left(\sum_{p \in \mathcal{M}_i \cap \mathcal{M}_j} \frac{1}{N_p} - 1\right) \end{aligned}$$

which for the chosen $\tilde{\rho_1}_{ij}$ in (8) further leads to

$$d_{ij}(k) = d_{ij}(k-1) + 2(x_i(k-1) - x_j(k-1))^\top \times \sum_{p \in \mathcal{M}_i \cap \mathcal{M}_j} \frac{1}{N_p}(\delta\hat{x}_{pi}(k) - \delta\hat{x}_{pj}(k)) > \rho_1.$$

Thus $x(k)$ satisfies $\mathcal{Q}_{2.2}$.

Considering now $\mathcal{Q}_{2.3}$, for each subproblem we have

$$||\delta\hat{x}_{ii}(k)|| < \tilde{v}_{\max,i} = \left(\sum_{j \in \mathcal{M}_i} \frac{1}{N_j}\right)^{-1} v_{\max}$$

and for the global problem

$$||\delta x_i(k)|| < \sum_{j \in \mathcal{M}_i} \frac{1}{N_j} ||\delta\hat{x}_{ji}(k)|| < v_{\max}$$

Thus $x(k)$ satisfies also $\mathcal{Q}_{2.3}$ and (C2) is established. □

Possible further improvements

A natural way to try to improve the proposed solution is to consider S sub-steps for each discrete time step, e.g. from $k-1$ to k. At each sub-step the agents are allowed to communicate to each other to average their local solutions according to (11). Therefore, in this case, the final position vector will be

$$x(k) = x(k-1) + \frac{1}{S}\sum_{p=1}^{S}\sum_{j \in \mathcal{M}_i} \frac{\delta\hat{x}_{ji}^{[p]}}{N_j}$$

where we indicate with $\hat{x}_{ji}^{[p]}$ the solution at the p-th sub-step.

However, given the nonlinear/nonconvex nature of the problem, it is not straightforward to prove that increasing S would lead to a better solution. Although simulation results show that in practice choosing bigger S helps to be closer to the centralized algorithm, at least when the initial configurations are the same, a rigorous characterization of this property is still an open question.

IV. Simulation Results

In this section, we present numerical simulation results to illustrate how the algorithms perform with respect to the centralized scheme. We use a benchmark problem motivated by [23]. This scenario starts with $N = 6$ agents forming a connected graph. The initial position vector is $x_i(0) = [1 + 1.05(i-1),\, y_i]^\top$, with $y_i \sim \aleph(0, \sigma)$, meaning that y_i is drawn from a Gaussian distribution $\aleph(0, \sigma)$, with mean 0 and standard deviation $\sigma = 0.5$. Randomness is added to test the algorithm's sensitivity to slightly different initial conditions. The other simulation parameters include the weighting function of Table I, $\rho_1 = 0.5$, $\rho_2 = 2$, velocity bound of 0.3, and final time $T = 75$. We collected 50 simulation runs. As a general notation, we will denote with $S = 1$ the standard distributed algorithm as expressed in Algorithm 1, while with $S = s$ the distributed algorithm with s sub-steps.

In Figure 2, an example of the trajectories using the centralized and the distributed solutions for the $S = 1$ case is depicted, respectively. The initial positions are marked with squares. The final positions are marked with circles. The bold lines represent the final communication graph and the thin lines the agent trajectories. The values of $\sqrt{\rho_1}$ and $\sqrt{\rho_2}$ are also depicted for comparison and illustration of scale.

Figure 3 shows, in the same simulation, the algebraic connectivity as a function of the sampling time k for the cases $S = 1, 2, 4$. Although omitted from Figure 2 for readability purposes, the $S = 2, 4$ cases lead qualitatively to the same final network configuration as $S = 1$. Figure 3 clearly illustrates the nonlinear/nonconvex nature of the problem. In fact, although $S = 4$ seems better at the beginning, it has slower convergence than $S = 2$. We can see also how $S = 4$ leads to a slightly better final λ_2 than the centralized case.

It appears also that the local problems grow in size as the connectivity increases. Random algorithms have been proposed to handle this growth in [28], but they do not possess the desired consistency properties.

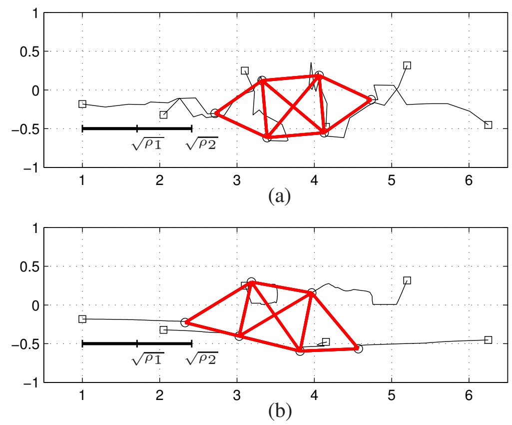


Fig. 2. Trajectories for the centralized solution (a) and for the distributed approach using $S = 1$ (b). The initial positions are marked with squares. The final positions are marked with circles. The bold lines represent the final communication graph and the thin lines the agent trajectories.

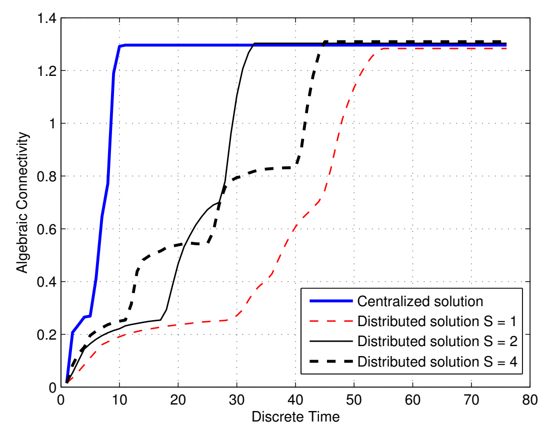


Fig. 3. Algebraic connectivity as a function of time k for both the centralized and the distributed solutions.

Table II shows the ratio between the final connectivity of the distributed solution and the centralized one for $S = 1$, in the 50 simulation runs. We can observe that although the distributed solution with $S = 1$ may even converge to a higher λ_2 value than the centralized one in some instances, or get stuck for a while in local minima, it has performance

TABLE II

RATIO BETWEEN THE FINAL CONNECTIVITY OF THE DISTRIBUTED SOLUTION AND THE CENTRALIZED ONE FOR $S = 1$.

Ratio	$(0.3-0.5]$	$(0.5-0.8]$	$(0.8-1.0]$	$(1.0-1.1]$
♯ of Cases	2	20	22	5

comparable to the centralized solution in about half of the cases. Furthermore, although the network size is rather small, the distributed solution provides on average a reduction of 20% in communication and of 30% in computational time of each agent solving the local optimization problems with respect to the centralized approximation.

V. CONCLUSIONS

We have presented a distributed solution for the maximization of algebraic connectivity in a network of mobile robots, which allows greater modularity and resilience compared to calculating the solution in a central location. The method is optimization-based, it is consistent with the approximate centralized solution and we have presented simulation results for different iteration strategies to assess the performance of the method. We highlighted cases in which the distributed solution converges to a higher λ_2 value than the centralized scheme, and cases in which it converges to local minima.

Although simulation results confirm the consistency of our distributed approach and show its practical applicability, several open issues still remain and will be the focus of our future research. In particular, a more realistic dynamical model for the agents, suboptimality of the distributed solution, implications due to the choice of the enlarged neighborhood are current research topics along with experimental validations.

REFERENCES

[1] N. E. Leonard, D. A. Paley, R. E. Davis, D. M. Fratantoni, F. Lekien, and F. Zhang, "Coordinated Control of an Underwater Glider Fleet in an Adaptive Ocean Sampling Field Experiment in Monterey Bay," *Journal of Field Robotics*, 2010, in press.

[2] D. Izzo and L. Pettazzi, "Autonomous and Distributed Motion Planning for Satellite Swarm," *Journal of Guidance, Control and Dynamics*, vol. 30, no. 2, pp. 449 – 459, 2007.

[3] H. Y. K. Lau and A. W. Y. Ko, "Coordination of Cooperative Search and Rescue Robots for Disaster Relief," in *Proceedings of IFAC World Congress*, Seul, Korea, July 6-11 2008.

[4] D. Casbeer, D. B. Kingston, R. W. Beard, and T. W. McLain, "Cooperative Forest Fire Surveillance using a Team of Small Unmanned Air Vehicles," *International Journal of Systems Science*, vol. 37, no. 6, pp. 351 – 360, 2006.

[5] G. Mathews and H. Durrant-Whyte, "Decentralised Optimal Control for Reconnaissance," in *Proceedings of the Conference on Information, Decision and Control*, Adelaide, Australia, February 12-14 2007.

[6] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*, ser. Applied Mathematics Series. Princeton University Press, 2008.

[7] T. Keviczky and K. H. Johansson, "A Study on Distributed Model Predictive Consensus," in *Proceedings of IFAC World Congress*, Seul, Korea, July 6-11 2008.

[8] W. Ren and R. Beard, *Distributed Consensus in Multi-Vehicle Cooperative Control: Theory and Applications*. Springer-Verlag London, 2008.

[9] R. Olfati-Saber, , and R. Murray, "Consensus Problems in Networks of Agents with Switching Topology and Time-Delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520 – 1533, September 2004.

[10] J. Cortés, "Distributed Algorithms for Reaching Consensus on General Functions," *Automatica*, vol. 44, pp. 726 – 737, 2008.

[11] J. Derenick, J. Spletzer, and A. Hsieh, "An Optimal Approach to Collaborative Target Tracking with Performance Guarantees," *Journal of Intelligent Robotic Systems*, vol. 56, pp. 47–67, 2009.

[12] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte, "Information-Theoretic Coordinated Control of Multiple Sensor Platforms," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 14-19 2003.

[13] T. H. Chung, V. Gupta, J. W. Burdick, and R. M. Murray, "On a Decentralized Active Sensing Strategy using Mobile Sensor Platforms in a Network," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, Paradise Island, the Bahamas, December 14-17 2004.

[14] S. Martínez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, pp. 661 – 668, 2006.

[15] D. A. Anisi, X. Hu, and P. Ögren, "Connectivity constrained multi-ugv surveillance," in *Proceedings of the 17th IFAC World Congress, Seoul, Korea*, July 2008.

[16] D. Spanos and R. M. Murray, "Robust Connectivity of Networked Vehicles," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, December 2004.

[17] J. Cortés, S. Martínez, and F. Bullo, "Robust Rendezvous for Mobile Autonomous Agents via Proximity Graphs in Arbitrary Dimensions," *IEEE Transaction on Automatic Control*, vol. 51, no. 8, pp. 1289 – 1298, August 2006.

[18] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar, "Decentralized Estimation and Control of Graph Connectivity in Mobile Sensor Networks," in *Proceedings of the American Control Conference*, Seattle, WA, USA, June 11-13 2008.

[19] M. M. Zavlanos and G. J. Pappas, "Distributed Connectivity Control of Mobile Networks," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1416 – 1428, December 2008.

[20] M. M. Zavlanos, H. G. Tanner, and A. Jadbabaie, "Hybrid Control for Connectivity Preserving Flocking," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2869 – 2875, December 2009.

[21] M. Schuresko and J. Cortés, "Distributed Motion Constraints for Algebraic Connectivity of Robotic Networks," *Journal of Intelligent Robotics Systems*, vol. 56, pp. 99 – 126, 2009.

[22] M. C. De Gennaro and A. Jadbabaie, "Decentralized Control of Connectivity for Multi-Agent Systems," in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, December 13-15 2006.

[23] Y. Kim and M. Mesbahi, "On Maximizing the Second Smallest Eigenvalue of a State-Dependent Graph Laplacian," *IEEE Transactions of Automatic Control*, vol. 51, no. 1, pp. 116–120, January 2006.

[24] S. Boyd, "Convex Optimization of Graph Laplacian Eigenvalues," in *Proceedings of the International Congress of Mathematicians*, Madrid, Spain, 2006, pp. 1311 – 1319.

[25] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, Belmont, Massachusetts, 1997.

[26] C. Langbort, L. Xiao, R. D'Andrea, and S. Boyd, "A Decomposition Approach to Distributed Analysis of Networked Systems," in *Proceedings of the 43rd IEEE Conference on Decision and Control, Bahamas*, 2004.

[27] A. Rantzer, "Dynamic Dual Decomposition for Distributed Control," in *Proceedings of American Control Conference*, St. Louis, MO, USA, June 10 - 12 2009.

[28] A. Simonetto, T. Keviczky, and R. Babuška, "Distributed Algebraic Connectivity Maximization for Robotic Networks: A Heuristic Approach," in *Proceedings of the 10th International Symposium on Distributed Autonomous Robotics Systems*, Lausanne, Switzerland, November 2010.

[29] I. D. Ivanov and E. de Klerk, "Parallel Implementation of a Semidefinite Programming Solver based on CSDP on a Distributed Memory Cluster ," *Optimization Methods and Software*, vol. 25, no. 3, pp. 405–420, June 2010.

[30] X. Zhao, D. Sun, and K. C. Toh, "A Newton-CG Augmented Lagrangian Method for Semidefinite Programming," *SIAM Journal on Optimization*, vol. 20, pp. 1737–1765, 2010.