Decentralized Connectivity Maintenance for Networked Lagrangian Dynamical Systems

Lorenzo Sabattini, Cristian Secchi and Nikhil Chopra

Abstract—In order to accomplish cooperative tasks, multi-robot systems are required to communicate among each other. Thus, maintaining the connectivity of the communication graph is a fundamental issue. Connectivity maintenance has been extensively studied in the last few years, but generally considering only kinematic agents. In this paper we will introduce a control strategy that, exploiting a decentralized procedure for the estimation of the algebraic connectivity of the graph, ensures the connectivity maintenance for groups of Lagrangian systems. The control strategy is validated by means of analytical proofs and simulation results.

I. INTRODUCTION

In this paper we address the connectivity maintenance issue for multi-robot systems. In [1], [2] we introduced a connectivity maintenance control strategy for single integrator agents, based on the decentralized estimation of the algebraic connectivity of the communication graph. In this paper we extend this control strategy, taking explicitly into account the dynamics of real robotic systems, modeled as Lagrangian systems.

Decentralized and cooperative control has been widely studied in the last few years. Among the main motivations for this interest is the increasing application of autonomous mobile robots, i.e. autonomous ground, aerial or underwater vehicles. Generally speaking, the main goal of decentralized control strategies is to make the robots cooperate, in order to satisfy some globally defined objective, without requiring a central computation unit, or a global knowledge of the entire state of the group. To achieve this objective, robots are required to communicate among each other. As mobile robots have limited communication capabilities, it is fundamental to ensure information exchange can occur among them. The communication architecture among the robots is usually modeled as a graph [3], usually referred to as the communication graph. Thus, to achieve the global objective, it is critical to guarantee the connectivity of the communication graph.

Decentralized control of groups of mobile robots has several applications, such as surveillance, automatic warehouses, exploration of unknown environments, search–and–rescue [4], [5]. In this kind of problems, robots are supposed to coordinate their motion, in order to achieve the global objective. Generally speaking, one of the main challenges is due to the fact that robots move into cluttered environments: unknown terrains to be explored, industrial environments for goods transportation, damaged buildings in search–and–rescue applications. This implies that collisions with obstacles must be avoided, while robots are performing their task. Thus, obstacle avoidance can interfere with the primary task of the robots: problems can arise, for instance, if some robots are trapped. One of the main issues, in this case, is to ensure the connectivity maintenance: due to the limited communication capabilities, in fact, the trapped robots can lose the connectivity with the rest of the group, which implies that the global objective can not be, in general, achieved.

In the literature, several approaches to connectivity maintenance have been proposed. These approaches can be divided into two categories: approaches to maintain the local connectivity, and approaches to maintain the global connectivity.

Maintaining the local connectivity entails designing a controller that ensures that, if a communication link is active at time $t = 0$, it will be active $\forall t \geq 0$. Examples of decentralized algorithms for local connectivity maintenance can be found in [6]–[9]. The main advantage of these control algorithms is that the formal proof of connectivity maintenance is generally shown. Nevertheless, imposing the maintenance of each single communication link is often too restrictive. In case of robots trapped due to the presence of obstacles in the environment, the requirement of local connectivity maintenance could impose to the system a rigid behavior, that is the entire group may get trapped.

To ensure that information exchange among all the robots is possible, it is necessary to guarantee only the global connectivity of the communication graph. Loosely speaking, it is acceptable that a few links are broken, as long as the overall graph is still connected: if necessary, redundant links can be removed, and new ones can be introduced. Thus, imposing the global connectivity maintenance ensures that none of the robots loses connectivity from the rest of the group, while maintaining the ability of the robots to operate in cluttered environments.

To the best of the authors’ knowledge, connectivity maintenance control strategies that can be found in the literature only tackle the problem for purely kinematic systems, i.e. groups of single integrator agents (see for instance the recent survey paper [10]). Nevertheless, in several applications, robots’ dynamics need to be explicitly taken into account. For instance, for industrial or military applications, when dealing with heavy and/or fast mobile robots that interact with the environment, dynamics can not be neglected.
The main contribution of this paper is the introduction of a global connectivity maintenance control strategy for groups of Lagrangian systems.

More specifically, we will extend the work first introduced in [1], [2], where a decentralized control strategy was introduced to proveably guarantee global connectivity for groups of single integrator agents.

The connectivity maintenance control strategy strongly relies on a decentralized estimation procedure to compute the second–smallest eigenvalue of the Laplacian matrix that, as shown in [11], is a measure of the connectivity of a graph.

The outline of the paper is as follows. In Section II we provide some background on graph theory. In Section III the connectivity maintenance control strategy and estimation procedure introduced in [1] for single integrator agents are summarized. In Section IV the connectivity maintenance control strategy for groups of Lagrangian systems proposed in this work is detailed. In Section V we apply the connectivity maintenance control strategy for a group of spacecraft performing a rendezvous application. Simulation results are provided for validation purposes. Concluding remarks are provided in Section VI.

II. BACKGROUND ON GRAPH THEORY

In this section we summarize some of the main notions on graph theory used in the paper. Further details can be found for instance in [12]. Given N mobile robots, we describe the communication architecture among them as an undirected graph. Each robot corresponds to a node of the graph, and each link between two robots corresponds to an edge of the graph. Let NI be the neighborhood of the i–th robot, i.e. the set of robots that can exchange information with the i–th one. The communication graph can be described by means of the adjacency matrix A ∈ R N × N. Each element aij is defined as the weight of the edge between the i–th and the j–th robot, and is a positive number if j ∈ NI, zero otherwise. Since we are considering undirected graphs, we assume aij = aji.

The degree matrix of the graph is defined as D = diag ({di}), where di is the degree of the i–th node of the graph, i.e. di = \sum j=1 N aij. The (weighted) Laplacian matrix of the graph is defined as L = D − A. The unweighted Laplacian matrix, L*, is defined as a special case of Laplacian matrix, where all non–zero entries of the adjacency matrix are equal to one. The Laplacian matrix exhibits some remarkable properties:

1) Let 1 be the column vector of all ones. Then, L1 = 0.
2) Let \lambda_, i = 1 . . . N be the eigenvalues of the Laplacian matrix.
   - The eigenvalues can be ordered such that
     \[ 0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N \] (1)
   - \lambda_2 > 0 if and only if the graph is connected; then, \lambda_2 is defined as the algebraic connectivity of the graph.

III. CONNECTIVITY MAINTENANCE AND ESTIMATION PROCEDURE FOR SINGLE INTEGRATOR AGENTS

In this section we summarize the main results on the connectivity maintenance control strategy first introduced in [1], [2], where we considered a group of N single integrator agents, i.e.:

\[ \dot{p}_i = u_i \] (2)

where pi ∈ ℝm is the position of the i–th agent, and ui is the control input. Let p = [p1T . . . pN T]T ∈ ℝNm be the state vector of the multi–agent system.

A. Connectivity maintenance control strategy

In [1] we introduced the following control law for kinematic agents:

\[ \dot{p}_i = u^c_i \] (3)

where u^c_i is defined as follows:

\[ u^c_i = \text{csch}^2 (\lambda_2 - \epsilon) \frac{\partial \lambda_2}{\partial p_i} \] (4)

where \epsilon is the desired lower–bound for the value of \lambda_2.

As shown in Fig. 1(a), the magnitude of the control action suddenly increases as the connectivity of the graph worsens.

Let R be the maximum communication range for each agent, i.e. the j–th agent is inside NI if \|pi − pj\| ≤ R. The edge–weights aij are defined as follows:

\[ a_{ij} = \begin{cases} e^{-((R^2)/2\sigma^2)} & \text{if } \|p_i - p_j\| \leq R \\ 0 & \text{otherwise} \end{cases} \] (5)

The scalar parameter \sigma is chosen to satisfy the threshold condition e^{-((R^2)/2\sigma^2)} = \Delta, where \Delta is a small predefined threshold. This definition of the edge–weights introduces a discontinuity in the control action, that can be avoided introducing a smooth bump function, as in [13].

Let \nu_2 be the eigenvector corresponding to the eigenvalue \lambda_2. Hence, the value of \frac{\partial \lambda_2}{\partial p_i} can be computed as shown in [14]:

\[ \frac{\partial \lambda_2}{\partial p_i} = v^2_2 \frac{\partial L}{\partial p_i} v_2 = \sum_{j \in N_i} \frac{\partial a_{ij}}{\partial p_i} (v^2_j - v^2_i) \] (6)

where v^k_j is the k–th component of v_2.

Given the definition of the edge–weights in Eq. (5), the value of \frac{\partial \lambda_2}{\partial p_i} can be computed as follows:

\[ \frac{\partial \lambda_2}{\partial p_i} = \sum_{j \in N_i} -a_{ij} (v^2_j - v^2_i) \frac{(p_i - p_j)}{\sigma^2} \] (7)

We now define a generalized communication model, that extends the scope of the one introduced in [1], [2]. Let \mu_{ij} = p_i − p_j, and let \hat{H} be some properly defined constant matrix. We suppose \hat{H} ≥ 0.

The matrix \hat{H} is defined such that the j–th agent is inside NI if \mu_{ij}^T \hat{H} \mu_{ij} ≤ R^2, for some parameter \sigma > 0.
The edge-weights $a_{i,j}$, first introduced in Eq. (5), are then re-defined as follows:

$$a_{i,j} = \begin{cases} \frac{p_i^THp_j}{\sigma^2} & \text{if } p_i^THp_j \leq R^2 \\ 0 & \text{otherwise} \end{cases}$$ (8)

The scalar parameter $\sigma$ is chosen to satisfy the threshold condition $e^{-\left(\frac{R^2}{2\sigma^2}\right)} = \Delta$, where $\Delta$ is a small predefined threshold.

Given the definition of the edge-weights in Eq. (8), the value of $\frac{\partial \lambda_2}{\partial p_i}$ can be computed as follows (see also Eq. (7)):

$$\frac{\partial \lambda_2}{\partial p_i} = \sum_{j \in N_i} -a_{ij}\left(\tilde{v}_j^2 - \tilde{v}_i^2\right)^2 \frac{H(p_i - p_j)}{\sigma^2}$$ (9)

B. Estimation of the algebraic connectivity of the graph

The computation of the eigenvectors of the Laplacian matrix is a centralized operation. Hence, the actual values of $\lambda_2$, $v_2$ and $\frac{\partial \lambda_2}{\partial p_i}$ are not available to the agents. As shown in [1], it is possible to exploit a decentralized procedure that allows each agent to obtain an estimate of these values. Inspired by [14], the estimate of $\lambda_2$ is computed by exploiting the estimation of the corresponding eigenvector $v_2$. Specifically, the power iteration procedure described in [15] is used to design the following update law:

$$\dot{\tilde{v}}_i = -k_1 \text{Ave}\left(\{\tilde{v}_j^2\}\right) 1 - k_2 L \tilde{v}_i - k_3 \left(\text{Ave}\left(\{\tilde{v}_j^2\}\right) - 1\right) \tilde{v}_i$$ (10)

where $k_1$, $k_2$, $k_3 > 0$ are the control gains, and $\text{Ave}\left(\cdot\right)$ is the averaging operation. Furthermore, $\tilde{v}_i$ is defined as the $i$–th agent’s estimate of $v_2$, the $i$–th component of the eigenvector $v_2$, and $\tilde{v} = [\tilde{v}_1, \ldots, \tilde{v}_N]^T$. Further details can be found in [14].

To implement the update law in Eq. (10) in a decentralized way, the averaging operation is implemented by means of the PI average consensus estimator described in [16].

Since there are two averaging operations in the update law in Eq. (10), two PI consensus estimators must be run:

- the first one, whose input is $\tilde{v}_i^2$, provides $z_1^2$ as the $i$–th agent’s estimate of $\text{Ave}\left(\{\tilde{v}_j^2\}\right)$;
- the second one, whose input is $\{\tilde{v}_j^2\}$, provides $z_2^2$ as the $i$–th agent’s estimate of $\text{Ave}\left(\{\tilde{v}_j^2\}\right)$.

Thus, each agent can run the decentralized version of the update law in Eq. (10):

$$\dot{\tilde{v}}_i = -k_1 z_1^2 - k_2 \sum_{j \in N_i} a_{i,j} \left(\tilde{v}_j^2 - \tilde{v}_i^2\right) - k_3 (z_2^2 - 1) \tilde{v}_i^2$$ (11)

Let $\tilde{\lambda}_2$ be the value that the second smallest eigenvalue of the Laplacian matrix would take if $\tilde{v}_2$ were the corresponding eigenvector. As demonstrated in [14], $\tilde{\lambda}_2$ can be computed as follows:

$$\tilde{\lambda}_2 = \frac{k_3}{k_2} \left[1 - \text{Ave}\left(\{\tilde{v}_j^2\}\right)\right]$$ (12)

Furthermore, $\frac{\partial \tilde{\lambda}_2}{\partial p_i}$ can be computed as in [14]:

$$\frac{\partial \tilde{\lambda}_2}{\partial p_i} = \tilde{v}_i^2 \frac{\partial L}{\partial p_i} \tilde{v}_i = \sum_{j \in N_i} \frac{\partial a_{i,j}}{\partial p_i} \left(\tilde{v}_j^2 - \tilde{v}_i^2\right)^2$$ (13)

Then, from the definition of the edge-weights $a_{i,j}$ given in Eq. (8):

$$\frac{\partial \tilde{\lambda}_2}{\partial p_i} = \sum_{j \in N_i} -a_{i,j} \left(\tilde{v}_j^2 - \tilde{v}_i^2\right)^2 \frac{p_i - p_j}{\sigma^2}$$ (14)

Further details can be found in in [1].

The actual value of $\lambda_2$ can not be computed by each agent. In fact, the real value of Ave $\left(\{\tilde{v}_j^2\}\right)$ is not available. Nevertheless, an estimate of this average, namely $\tilde{z}_2^2$, is available to each agent. Hence, the $i$–th agent can compute its own estimate of $\lambda_2$, namely $\lambda_i^2$, as follows:

$$\lambda_i^2 = \frac{k_3}{k_2} \left(1 - \tilde{z}_2^2\right)$$ (15)

As shown in [1], $\lambda_i^2$ is a good estimate of both $\lambda_2$ and $\tilde{\lambda}_2$. More specifically, it has been proven that $\Xi \leq \lambda_i^2 \leq \lambda_2$ such that

$$\lambda_i^2 - \lambda_2 \leq \Xi \quad \forall i = 1, \ldots, N$$ (16)

From Eq. (16), we can conclude that

$$\left|\lambda_2 - \tilde{\lambda}_2\right| \leq \Xi + \Xi'$$ (17)

The control law introduced in Eq. (4) will then be implemented introducing each agent’s estimates, that is:

$$u_i^c = \text{csch}^2 \left(\lambda_i^2 - \tilde{\epsilon}\right) \frac{\partial \tilde{\lambda}_2}{\partial p_i}$$ (18)

where $\tilde{\epsilon} = \epsilon + 2\Xi + \Xi'$.

As shown in Fig. 1(a), the magnitude of the control action suddenly increases as the connectivity of the graph worsens.

![Fig. 1. Energy function and its derivative, with respect to $\lambda_2$](image_url)

Consider the following control law:

$$\hat{p}_i = u_i^c + u_i^d$$ (19)

where $u_i^c$ is the control term introduced in Eq. (18), while $u_i^d$ is a control term used to obtain some desired behavior. Namely, the control term $u_i^d$ is an unknown bounded function, i.e., $\|u_i^d\| \leq u_M$. As demonstrated in [2], the boundedness of the estimation error is a sufficient condition to guarantee the connectivity maintenance, even in the presence of an external (bounded) control law.
We remark that, as we are exploiting passivity based analysis, dissipative term due to friction, and represents the Coriolis effects, the matrix \( D \) (22) value of \( H \) hence:

\[
M(p_i) \ddot{p}_i + C(p_i, \dot{p}_i) \dot{p}_i + D \dot{p}_i + g(p_i) = u_i \quad (20)
\]

where \( u_i \) is the control input. The matrix \( M(p_i) \) is the symmetric positive definite mass matrix, the matrix \( C(p_i, \dot{p}_i) \) represents the Coriolis effects, the matrix \( D \) represents a dissipative term due to friction, and \( g(p_i) \) is the gravity term. Further details can be in [17].

As in [1], let the energy function be defined as follows:

\[
V(p) = \coth \left( \lambda_2 - \epsilon \right) \quad (21)
\]

We introduce the following control law:

\[
u_i = g(p_i) + u_i^e \quad (22)
\]

We remark that, as we are exploiting passivity based analysis, adaptive gravity compensation is possible. From Eqs. (18)–(21):

\[
u_i^e = -\frac{\partial V}{\partial p_i} \quad (23)
\]

**Proposition 1** Consider the dynamic described by Eq. (20). Let \( \Xi, \Xi' \) be defined according to Eq. (16), and let \( \bar{\epsilon} = \epsilon + 2\Xi + \Xi' \). If the initial value of \( \lambda_2 > \bar{\epsilon} + \Xi + \Xi' \), then the control law defined in Eqs. (22)–(23) ensures that the value of \( \lambda_2 \) never goes below \( \epsilon \).

**Proof:** Let

\[
W(p, \dot{p}) = \frac{1}{2} \sum_{i=1}^{N} p_i^T M(p_i) \ddot{p}_i + V(p) \quad (24)
\]

We compute the time derivative of \( W \):

\[
\dot{W}(p, \dot{p}) = \dot{p}^T \nabla_p W(p) = \sum_{i=1}^{N} p_i^T \frac{\partial W}{\partial p_i} \quad (25)
\]

Hence:

\[
\dot{W}(p, \dot{p}) = \sum_{i=1}^{N} \left( p_i^T M(p_i) \ddot{p}_i + \frac{1}{2} p_i^T \dot{M}(p_i) \dot{p}_i + \dot{p}_i^T \frac{\partial V}{\partial p_i} \right) \quad (26)
\]

From Eqs. (20), (22), (23):

\[
M(p_i) \ddot{p}_i + C(p_i, \dot{p}_i) \dot{p}_i + D \dot{p}_i = -\frac{\partial V}{\partial p_i} \quad (27)
\]

Hence:

\[
M(p_i) \ddot{p}_i = -\frac{\partial V}{\partial p_i} - C(p_i, \dot{p}_i) \dot{p}_i - D \dot{p}_i \quad (28)
\]

Hence, from Eqs. (26), (28):

\[
\dot{W}(p, \dot{p}) = \sum_{i=1}^{N} \left( -\dot{p}_i^T \frac{\partial V}{\partial p_i} - \dot{p}_i^T C(p_i, \dot{p}_i) \dot{p}_i - \dot{p}_i^T D \dot{p}_i + \frac{1}{2} \dot{p}_i^T \dot{M}(p_i) \dot{p}_i + \dot{p}_i^T \frac{\partial V}{\partial p_i} \right) \quad (29)
\]

Hence:

\[
\dot{W}(p, \dot{p}) = \sum_{i=1}^{N} \left( \frac{1}{2} \dot{p}_i^T \left( \dot{M}(p_i) - 2C(p_i, \dot{p}_i) \right) \dot{p}_i - \dot{p}_i^T D \dot{p}_i \right) \quad (30)
\]

With a slight abuse of notation, hereafter we will refer to \( W(t) \) and \( V(t) \), even though \( W(\cdot) \) and \( V(\cdot) \) are not explicit functions of time.

Hence, \( \forall t \geq 0, W(t) \leq W(0), \forall t \geq 0 \). From Eq. (24), it follows that \( V(t) \leq W(t), \forall t \geq 0 \). Thus, we can conclude that \( V(t) \leq W(0), \forall t \geq 0 \).

Given the definition of \( V(t) \) provided in Eq. (21), we know that \( V \) is monotonically decreasing with respect to \( \lambda_2 \), \( \forall \lambda_2 > \bar{\epsilon} \). According to Eq. (17), the fact that the initial value of \( \lambda_2 \) is greater than \( \bar{\epsilon} + \Xi + \Xi' \) ensures that the initial value of \( \lambda_2 \) is greater than \( \bar{\epsilon} \).

Thus, we can conclude that \( \exists \lambda_2 > \bar{\epsilon} \) such that \( \lambda_2(t) \geq \lambda_2, \forall t \geq 0 \).

Hence, according to Eq. (17), we conclude that \( \lambda_2 \geq \epsilon = \bar{\epsilon} - 2\Xi - \Xi' \).

**B. Connectivity and external control laws**

In this section, the presence of external control laws is explicitly considered:

\[
u_i = g(p_i) + u_i^e + u_i^d \quad (31)
\]

Specifically, we consider the case where \( u_i^d \) is the gradient of an appropriately designed potential function, that is:

\[
u_i^d = -\frac{\partial U(p)}{\partial p_i} \quad (32)
\]

where we suppose that \( U(p) \) is a positive definite potential function.

**Proposition 2** Consider the dynamic described by Eq. (20). Let \( \Xi, \Xi' \) be defined according to Eq. (16), and let \( \bar{\epsilon} = \epsilon + 2\Xi + \Xi' \). If the initial value of \( \lambda_2 > \bar{\epsilon} + \Xi + \Xi' \), then the control law defined in Eqs. (23)–(31)–(32) ensures that the value of \( \lambda_2 \) never goes below \( \epsilon \).

**Proof:** Let

\[
T(p, \dot{p}) = \frac{1}{2} \sum_{i=1}^{N} p_i^T M(p_i) \dot{p}_i + V(p) + U(p) \quad (33)
\]

We compute the time derivative of \( T \):

\[
\dot{T}(p, \dot{p}) = \dot{p}^T \nabla_p T(p) = \sum_{i=1}^{N} p_i^T \frac{\partial T}{\partial p_i} \quad (34)
\]
Hence:
\[
\dot{T}(p, \dot{p}) = \sum_{i=1}^{N} \left( p_i^T M(p_i) \dot{p}_i + \frac{1}{2} p_i^T \dot{M}(p_i) \dot{p}_i + p_i^T \frac{\partial V}{\partial p_i} \dot{p}_i \right)
\]
From Eqs. (20), (31), (23), (32):
\[
M(p_i) \dot{p}_i + C(p_i, \dot{p}_i) \dot{p}_i + D \ddot{p}_i = -\frac{\partial V}{\partial p_i} - \frac{\partial U}{\partial p_i}
\]
Thus:
\[
M(p_i) \dot{p}_i - \frac{\partial V}{\partial p_i} - \frac{\partial U}{\partial p_i} - C(p_i, \dot{p}_i) \dot{p}_i - D \ddot{p}_i = 0
\]
Hence, from Eqs. (35), (37):
\[
\dot{T}(p, \dot{p}) = \sum_{i=1}^{N} \left( -p_i^T \frac{\partial V}{\partial p_i} - p_i^T \frac{\partial U}{\partial p_i} - p_i^T C(p_i, \dot{p}_i) \dot{p}_i - \frac{1}{2} p_i^T \dot{M}(p_i) \dot{p}_i + \frac{1}{2} p_i^T \dot{M}(p_i) \dot{p}_i + p_i^T \frac{\partial V}{\partial p_i} + p_i^T \frac{\partial U}{\partial p_i} \right)
\]
Consequently:
\[
\dot{T}(p, \dot{p}) = \sum_{i=1}^{N} \left( -\frac{1}{2} p_i^T \dot{M}(p_i) \dot{p}_i - 2C(p_i, \dot{p}_i) \dot{p}_i - p_i^T \dot{M}(p_i) \dot{p}_i \right)
\]
\[
= \sum_{i=1}^{N} (-\frac{1}{2} p_i^T \dot{M}(p_i) \dot{p}_i) \leq 0
\]

With a slight abuse of notation, hereafter we will refer to $T(t)$, $V(t)$ and $U(t)$, even though $T(\cdot)$, $V(\cdot)$ and $U(\cdot)$ are not explicit functions of time.

Hence, $\forall t \geq 0$, $T(t) \leq T(0)$, $\forall t \geq 0$. From Eq. (33), it follows that $V(t) \leq T(t)$, $\forall t \geq 0$. Thus, we can conclude that $V(t) \leq T(0)$, $\forall t \geq 0$.

Given the definition of $V(t)$ provided in Eq. (21), we know that $V$ is monotonically decreasing with respect to $\lambda_2$, $\forall \lambda_2 > \epsilon$. According to Eq. (17), the fact that the initial value of $\lambda_2$ is greater than $\epsilon + \Xi + \Xi'$ ensures that the initial value of $\lambda_2$ is greater than $\epsilon$.

Thus, we can conclude that $\exists \lambda_2 > \epsilon$ such that $\lambda_2(t) \geq \lambda_2$, $\forall t \geq 0$.

Hence, according to Eq. (17), we conclude that $\lambda_2 \geq \epsilon = \Xi + \Xi'$.

V. APPLICATION: RENDEZVOUS FOR FULLY ACTUATED LAGRANGIAN SYSTEMS

In this section we describe the application of our connectivity maintenance control strategy for a group of fully actuated Lagrangian systems performing a rendezvous task.

A. Dynamics and control law

We consider a group of 6-degree-of-freedom spacecraft vehicles, whose dynamics are described in [18].

Specifically, the configuration of these vehicles is described by the following state vectors:
\[
p_i = [x_i^T \theta_i^T]^T \quad \dot{p}_i = [v_i^T \omega_i^T]
\]
where $x_i \in \mathbb{R}^3$ represents the position of the $i$-th robot, and $\theta_i$ represents the rotation of the $i$-th robot, expressed in terms of Euler parameters [19]. $v_i \in \mathbb{R}^3$ and $\omega_i \in \mathbb{R}^3$ are the linear and angular velocity of the $i$-th robot, respectively.

The following relationship holds:
\[
\dot{x}_i = v_i \quad \dot{\theta}_i = T(p_i) \omega_i
\]
where $T(p_i)$ is a properly defined transformation matrix.

Referring to Eq. 20, the matrices that describe the dynamics of each spacecraft vehicle are defined as follows:
\[
M(p_i) = \begin{bmatrix} m_s & I_3 \\ 0_{3 \times 3} & J_s(p_i) \end{bmatrix} \\
C(p_i, \dot{p}_i) = \begin{bmatrix} C_t(x_i, \dot{x}_i) & 0_{3 \times 3} \\ 0_{3 \times 3} & C_r(\theta_i, \omega_i) \end{bmatrix} \\
g(p_i) = \begin{bmatrix} g_0(x_i) \\ 0_{3 \times 1} \end{bmatrix} \\
D = 0_{3 \times 3}
\]
where $0_{3 \times 3}$ is a zero matrix with $3$ rows and $3$ columns, and $I_3$ is the identity matrix of size $3$. The value $m_s$ represents the mass of the spacecraft, while $J_s(p_i)$ is the matrix representing the moments of inertia.

From Eq. (42) it’s easy to see that translations and rotations are decoupled, and can be independently controlled. Hence, hereafter we will consider only the translational dynamics of the system. The matrix $C_t(x_i, \dot{x}_i)$ is a Coriolis–like skew–symmetric matrix, and is defined as follows:
\[
C_t(x_i, \dot{x}_i) = 2m_s \dot{v}_i \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}
\]
The gravity term $g_t(x_i)$ is defined as follows:
\[
g_t(x_i) = m_f \begin{bmatrix} \frac{\mu}{r_s^3} - \dot{r}_s^2 & -\dot{v}_i & 0 \\ \dot{v}_i & \frac{\mu}{r_s^3} - \dot{r}_s^2 & 0 \\ 0 & 0 & \frac{\mu}{r_s^3} \end{bmatrix} x_i
\]
where $r_s$ is the average radius of the orbit of the spacecraft. Let $G$ be the universal constant of gravity, and let $M_s$ be the mass of the Earth: then, $\mu \approx GM_s$.

We consider the following connectivity model: two robots can communicate if their Euclidean distance is less than or equal to $R$. More specifically, we define the matrix $H$ in Eq. (8) as follows:
\[
H = \begin{bmatrix} I_3 & 0_{3 \times 4} \\ 0_{3 \times 4} & 0_{1 \times 4} \end{bmatrix}
\]
With this definition of $H$, we obtain that $p_{ij}^T H p_{ij}$ is exactly the Euclidean distance between the $i$-th and the $j$-th robot. According to the definition of the edge–weights introduced in Eq. (8), the $i$-th and the $j$-th agents are neighbors if their Euclidean distance is less than or equal to $R$. Furthermore, given the definition of the matrix $H$ provided in Eq. (45), we can state that only the first three components of the control action $u_i$ will be different from zero.
the video) that, starting from random initial positions, are placed them in randomly chosen initial positions.

In order to make the robots perform a rendezvous task, we introduce an additional control law $u_i^d$ defined as in Eq. (32), where the potential field $U(p)$ is defined as follows:

$$U_i = \sum_{j \in N_i} \frac{1}{2} K_r (x_i - x_j)$$

where $K_r > 0$ is a properly defined constant. It’s easy to prove that, as long as the communication graph is connected, this control law yields to the rendezvous of the multi–robot system.

B. Simulations

In order to validate the control strategy presented in this paper, we implemented several Matlab simulations. We have varied the number of agents, from $N = 3$ to $N = 20$, and placed them in randomly chosen initial positions.

The accompanying video clip shows a typical run of a rendezvous simulation, with six robots (colored pyramids in the video) that, starting from random initial positions, are supposed to converge to a common point, while avoiding collisions with randomly placed point obstacles (blue dots in the video). For obstacle avoidance purposes, a repulsive artificial potential field has been added (see e.g. [20]). Without the connectivity maintenance controller, the connectivity is lost quite soon: the obstacle avoidance action obstructs the desired movement of some agents, that are thus trapped and lose connectivity with the other ones. As expected, using the connectivity maintenance control action the connectivity of the graph is always preserved, as shown also in Fig. 2.

VI. CONCLUSIONS

In this paper we have presented a decentralized control strategy for the maintenance of the connectivity of the communication graph for groups of Lagrangian systems. This control strategy is an extension of the work introduced in [1], [2], where the connectivity maintenance issue is addressed for single integrator kinematic agents. The algorithm relies on a decentralized estimation procedure, that allows each robot to compute a local estimate of the algebraic connectivity of the graph, exploiting only local information. Connectivity maintenance has been formally demonstrated by means of analytical proofs.

Simulations have been also provided to validate the control strategy for a group of spacecraft satellites performing a rendezvous task, while moving through randomly placed obstacles.

Current work aims at the implementation of the connectivity maintenance control strategy for under–actuated Lagrangian systems. The presence of nonholonomic constraints will be investigated as well.

REFERENCES