Maintaining limited-range connectivity among second-order agents

Giuseppe Notarstefano Ketan Savla Francesco Bullo Ali Jadbabaie

***Abstract*— In this paper we consider ad-hoc networks of robotic agents with double integrator dynamics. For such networks, the connectivity maintenance problems are: (i) do there exist control inputs for each agent to maintain network connectivity, and (ii) given desired controls for each agent, can one compute the closest connectivity-maintaining controls in a distributed fashion? The proposed solution is based on three contributions. First, we define and characterize admissible sets for double integrators to remain inside disks. Second, we establish an existence theorem for the connectivity maintenance problem by introducing a novel state-dependent graph, called the *double-integrator disk graph*. Finally, we design a distributed "flow-control" algorithm to compute optimal connectivity-maintaining controls.**

I. INTRODUCTION

This work is a contribution to the emerging discipline of motion coordination for ad-hoc networks of mobile autonomous agents. This loose terminology refers to groups of robotic agents with limited mobility and communication capabilities. In the not too distant future, these groups of coordinated devices will perform a variety of challenging tasks including search and recovery operations, surveillance, exploration and environmental monitoring. The potential advantages of employing arrays of agents have recently motivated vast interest in this topic. For example, from a control viewpoint, a group of agents inherently provides robustness to failures of single agents or of communication links.

The motion coordination problem for groups of autonomous agents is a control problem in the presence of communication constraints. Typically, each agent makes decisions based only on partial information about the state of the entire network that is obtained via communication with its immediate neighbors. One important difficulty is that the topology of the communication network depends on the agents' locations and, therefore, changes with the evolution of the network. In order to ensure a desired emergent behavior for a group of agents, it is necessary that the group does not disintegrate into subgroups that are unable to communicate with each other. In other words, some restrictions must be applied on the movement of the agents to ensure *connectivity* among the members of the group. In terms of design, it is required to constrain the control input such that the resulting topology maintains connectivity throughout its course of evolution. In [1], a connectivity constraint was developed for a group of agents modeled as first-order discrete time dynamic systems. In [1] and in the related references [2], [3], this constraint is used to solve rendezvous problems. Connectivity constraints for line-of-sight communication are proposed in [4]. Another approach to connectivity maintenance for first-order systems is proposed in [5]. In this paper we fully characterize the set of admissible control inputs for a group of agents modeled as second order discrete time dynamic systems, which would ensure connectivity of the group in the same spirit as described earlier.

The contributions of the paper are threefold. First, we consider a control system consisting of a double integrator with bounded control inputs. For such a system, we define and characterize the admissible set that allows the double integrator to remain inside disks. Second, we define a novel state-dependent graph – the *double-integrator disk graph* – and give an existence theorem for the connectivity maintenance problem for networks of second order agents with respect to an appropriate version of this new graph. Finally, we consider a relevant optimization problem, where given a set of desired control inputs for all the agents it is required to find the optimal set of connectivity-maintaining control inputs. We cast this problem into a standard quadratic programming problem and provide a distributed "flow-control" algorithm to solve it.

II. PRELIMINARY DEVELOPMENTS

We begin with some notations. We let $\mathbb{N}$, $\mathbb{N}_0$, and $\mathbb{R}_+$ denote the natural numbers, the non-negative integer numbers, and the positive real numbers, respectively. For $d \in \mathbb{N}$, we let 0_d and 1_d denote the vectors whose entries are all 0 and 1, respectively. We let $\|p\|$ denote the Euclidean norm of $p \in \mathbb{R}^d$. For $r \in \mathbb{R}_+$ and $p \in \mathbb{R}^d$, we let $\overline{B}(p,r)$ denote the closed ball centered at p with radius r, i.e., $\overline{B}(p,r) = \{q \in \mathbb{R}^d \mid \|p-q\| \leq r\}$. For $x, y \in \mathbb{R}^d$, we let $x \preceq y$ denote component-wise inequality, i.e., $x_k \leq y_k$ for $k \in \{1,\dots,d\}$. We let $f : A \rightrightarrows B$ denote a set-valued map; in other words, for each $a \in A$, $f(a)$ is a subset of B.

A. Maintaining a double integrator inside a disk

For $t \in \mathbb{N}_0$, consider the discrete-time control system in $\mathbb{R}^{2d}$

$$\begin{aligned} p[t+1] &= p[t] + v[t], \\ v[t+1] &= v[t] + u[t], \end{aligned} \tag{1}$$

where the norm of the control is upper-bounded by $r_{\text{ctr}} \in \mathbb{R}_+$, i.e., $u[t] \in \overline{B}(0_d, r_{\text{ctr}})$ for $t \in \mathbb{N}_0$. We refer to this control

Giuseppe Notarstefano is with the the Department of Information Engineering, Università di Padova, Via Gradenigo 6/a, 35131 Padova, Italy, notarste@dei.unipd.it

Ketan Savla and Francesco Bullo are with the Center for Control, Dynamical Systems and Computation, University of California at Santa Barbara, Santa Barbara, CA 93106, ketansavla@umail.ucsb.edu, bullo@engineering.ucsb.edu

Ali Jadbabaie is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, jadbabai@seas.upenn.edu

system as the *discrete-time double integrator* in $\mathbb{R}^d$ or, more loosely, as a second-order system. Given $(p,v) \in \mathbb{R}^{2d}$ and $\{u_\tau\}_{\tau\in\mathbb{N}_0} \subseteq \overline{B}(0_d, r_{\text{ctr}})$, let $\phi(t,(p,v),\{u_\tau\})$ denote the solution of (1) at time $t \in \mathbb{N}_0$ from initial condition (p,v) with inputs $u_1,\dots,u_{t-1}$.

In what follows we consider the following problem: assume that the initial position of (1) is inside a disk centered at 0_d, find inputs that keep it inside that disk. This task is impossible for general values of the initial velocity. In what follows we identify assumptions on the initial velocity that render the task possible.

For $r_{\text{pos}} \in \mathbb{R}_+$, we define the *admissible set at time zero* by

$$\mathcal{A}_0^d(r_{\text{pos}}) = \overline{B}(0_d, r_{\text{pos}}) \times \mathbb{R}^d. \tag{2}$$

For $r_{\text{pos}}, r_{\text{ctr}} \in \mathbb{R}_+$, we define the *admissible set* by

$$\begin{aligned}\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}}) = \big\{(p,v) \in \mathbb{R}^{2d} \mid \exists \{u_\tau\}_{\tau\in\mathbb{N}_0} \subseteq \overline{B}(0_d, r_{\text{ctr}}) \\ \text{s.t. } \phi(t,(p,v),\{u_\tau\}) \in \mathcal{A}_0^d(r_{\text{pos}}), \ \ \forall t \in \mathbb{N}_0\big\},\end{aligned} \tag{3}$$

and the *admissible set for m time steps* by

$$\begin{aligned}\mathcal{A}_m^d = \big\{(p,v) \in \mathbb{R}^{2d} \mid \exists \{u_\tau\}_{\tau\in[0,m-1]} \subseteq \overline{B}(0_d, r_{\text{ctr}}) \\ \text{s.t. } \phi(t,(p,v),\{u_\tau\}) \in \mathcal{A}_0^d(r_{\text{pos}}), \ \ \forall t \in [0,m]\big\}.\end{aligned} \tag{4}$$

The following theorem characterizes the admissible set.

Theorem 2.1: For all $d \in \mathbb{N}$ and $r_{\text{pos}}, r_{\text{ctr}} \in \mathbb{R}_+$, the following statements hold:

(i) for all $m \in \mathbb{N}$, $\mathcal{A}_m^d \subseteq \mathcal{A}_{m-1}^d$ and there exists $\bar{m}(r_{\text{pos}}, r_{\text{ctr}})$ such that $\mathcal{A}_m^d = \mathcal{A}_{\bar{m}}^d$, for all $m \geq \bar{m}$;

(ii) $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}}) = \lim_{m\to+\infty} \mathcal{A}_m^d$;

(iii) $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$ is a convex, compact set and is the largest controlled-invariant[1] subset of $\mathcal{A}_0^d(r_{\text{pos}})$;

(iv) $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$ is invariant under orthogonal transformations in the sense that, if $(p,v) \in \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$, then also $(Rp, Rv) \in \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$ for all orthogonal[2] matrices R in $\mathbb{R}^{d\times d}$;

(v) if $0 < r_1 < r_2$, then $\mathcal{A}^d(r_{\text{pos}}, r_1) \subset \mathcal{A}^d(r_{\text{pos}}, r_2)$ and $\mathcal{A}^d(r_1, r_{\text{ctr}}) \subset \mathcal{A}^d(r_2, r_{\text{ctr}})$.

Next, we study the set-valued map that associates to each state in $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$ the set of control inputs that keep the state inside $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$ in one step. We define the *admissible control set* $\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) : \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}}) \rightrightarrows \overline{B}(0_d, r_{\text{ctr}})$ by

$$\begin{aligned}\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p,v) = \{u \in \overline{B}(0_d, r_{\text{ctr}}) \mid \\ (p+v, v+u) \in \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})\},\end{aligned} \tag{5}$$

or, equivalently,

$$\begin{aligned}&\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p,v) \\ &\quad = \overline{B}(0_d, r_{\text{ctr}}) \cap \{w - v \mid (p+v, w) \in \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})\}.\end{aligned}$$

Lemma 2.2: For all $(p,v) \in \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$, the set $\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p,v)$ is non-empty, convex and compact. For generic (p,v), the set $\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p,v)$ does not contain 0_d.

[1] A set is controlled invariant for a control system if there exists a feedback law such that the set is positively invariant for the closed-loop system.

[2] A matrix $R \in \mathbb{R}^{d\times d}$ is orthogonal if $RR^T = R^TR = I_d$.

B. Computing admissible sets

We characterize $\mathcal{A}^d$ for $d = 1$ in the following result and we illustrate the outcome in Figure 1.

Lemma 2.3: For $r_{\text{pos}}, r_{\text{ctr}} \in \mathbb{R}_+$, $\mathcal{A}^1(r_{\text{pos}}, r_{\text{ctr}})$ is the polytope containing the points $(p,v) \in \mathbb{R}^2$ satisfying

$$-\frac{r_{\text{pos}}}{m} - \frac{m-1}{2} r_{\text{ctr}} \le v + \frac{p}{m} \le \frac{r_{\text{pos}}}{m} + \frac{m-1}{2} r_{\text{ctr}},$$

for all $m \in \mathbb{N}$, and $-r_{\text{pos}} \le p \le r_{\text{pos}}$.

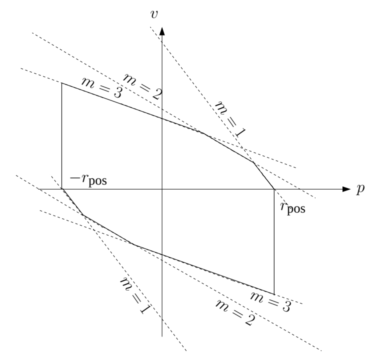

Fig. 1. The admissible set $\mathcal{A}^1$ for generic values of r_{pos} and r_{ctr}

Remark 2.4: The methodology for constructing $\mathcal{A}^1(r_{\text{pos}}, r_{\text{ctr}})$ closely follows the procedure for constructing the so-called *isochronic regions* using principles from discrete-time optimal control, as outlined in [6]. □

Next, we introduce some definitions useful to provide an inner approximation of $\mathcal{A}^d$ when $d \geq 2$. Given $p \in \mathbb{R}^d$ and $v \in \mathbb{R}^d \setminus \{0_d\}$, define $p_\parallel \in \mathbb{R}$ and $p_\perp \in \mathbb{R}^d$ by

$$p = p_\parallel \frac{v}{\|v\|} + p_\perp,$$

where $p_\perp \cdot v = 0$. For $r_{\text{pos}}, r_{\text{ctr}} \in \mathbb{R}_+$, define

$$\begin{aligned}\mathcal{A}_\parallel^d(r_{\text{pos}}, r_{\text{ctr}}) = \big\{(p,v) \in \overline{B}(0_d, r_{\text{pos}}) \times \mathbb{R}^d \mid v = 0_d \text{ or} \\ (p_\parallel, \|v\|) \in \mathcal{A}^1\big(\sqrt{r_{\text{pos}}^2 - \|p_\perp\|^2}, r_{\text{ctr}}\big)\big\}.\end{aligned}$$

Lemma 2.5: For $r_{\text{pos}}, r_{\text{ctr}} \in \mathbb{R}_+$,

(i) $\mathcal{A}_\parallel^d(r_{\text{pos}}, r_{\text{ctr}})$ is a subset of $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$, and

(ii) $\mathcal{A}_\parallel^d(r_{\text{pos}}, r_{\text{ctr}})$ is convex and compact.

Remark 2.6: In what follows we silently adopt the inner approximation $\mathcal{A}_\parallel^d$ for the set $\mathcal{A}^d$ anytime $d \geq 2$. Furthermore, we perform computations by adopting inner polytopic representations for the various compact convex sets. □

C. The double-integrator disk graph

Let us introduce some concepts about state dependent graphs and some useful examples. For a set X, let $\mathbb{F}(X)$ be the collection of finite subsets of X; e.g., $\mathcal{P} \in \mathbb{F}(\mathbb{R}^d)$ is a set of points. For a set X, let $\mathbb{G}(X)$ be the set of undirected graphs whose vertices are elements of X, i.e., whose vertex set belongs to $\mathbb{F}(X)$. For a set X, a *state dependent graph on* X is a map $\mathcal{G} : \mathbb{F}(X) \to \mathbb{G}(X)$ that associates to a finite subset V of X an undirected graph with vertex set V and edge set $\mathcal{E}_{\mathcal{G}}(V)$ where $\mathcal{E}_{\mathcal{G}} : \mathbb{F}(X) \to \mathbb{F}(X \times X)$ satisfies $\mathcal{E}_{\mathcal{G}}(V) \subseteq V \times V$. In other words, what edges exists in $\mathcal{G}(V)$ depends on the elements of V that constitute the nodes.

The following three examples of state dependent graphs play an important role. First, given $r_{\text{pos}} \in \mathbb{R}_+$, the *disk graph* $\mathcal{G}_{\text{disk}}(r_{\text{pos}})$ is the state dependent graph on $\mathbb{R}^d$ defined as follows: for $\{p_1, \dots, p_n\} \subset \mathbb{R}^d$, the pair (p_i, p_j) is an edge in $\mathcal{G}_{\text{disk}}(r_{\text{pos}}) \cdot (\{p_1, \dots, p_n\})$ if and only if $p_i - p_j \in \overline{B}(0_d, r_{\text{pos}})$. Second, given $r_{\text{pos}}, r_{\text{ctr}} \in \mathbb{R}_+$, the *double-integrator disk graph* $\mathcal{G}_{\text{di-disk}}(r_{\text{pos}}, r_{\text{ctr}})$ is the state dependent graph on $\mathbb{R}^{2d}$ defined as follows: for $\{(p_1, v_1), \dots, (p_n, v_n)\} \subset \mathbb{R}^{2d}$, the pair $((p_i, v_i), (p_j, v_j))$ is an edge if and only if the relative positions and velocities satisfy

$$(p_i - p_j, v_i - v_j) \in \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}}).$$

Third, it is convenient to define the disk graph also as a state dependent graph on $\mathbb{R}^{2d}$ by stating that $((p_i, v_i), (p_j, v_j))$ is an edge if and only if (p_i, p_j) is an edge of the disk graph on $\mathbb{R}^d$. We illustrate the first two graphs in Figure 2.

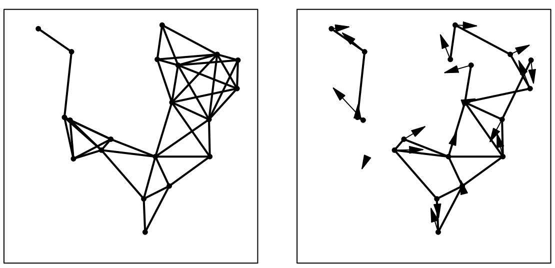

Fig. 2. The disk graph and the double-integrator disk graph in $\mathbb{R}^2$ for 20 agents with random positions and velocities.

III. Connectivity constraints among second-order agents

In this section we state the model, the notion of connectivity, and a sufficient condition that guarantees connectivity can be preserved.

A. Networks of robotic agents with second-order dynamics and the connectivity maintenance problem

We begin by introducing the notion of *network of robotic agents with second-order dynamics* in $\mathbb{R}^d$. Let n be the number of agents. Each agent has the following computation, motion control, and communication capabilities. For $i \in \{1, \dots, n\}$, the ith agent has a processor with the ability of allocating continuous and discrete states and performing operations on them. The ith agent occupies a location $p_i \in \mathbb{R}^d$, moves with velocity $v_i \in \mathbb{R}^d$, according to the discrete-time double integrator dynamics in (1), i.e.,

$$\begin{aligned} p_i[t+1] &= p_i[t] + v_i[t], \\ v_i[t+1] &= v_i[t] + u_i[t], \end{aligned} \tag{6}$$

where the norm of all controls $u_i[t]$, $i \in \{1, \dots, n\}$, $t \in \mathbb{N}_0$, is upper-bounded by $r_{\text{ctr}} \in \mathbb{R}_+$. The processor of each agent has access to the agent location and velocity. Each agent can transmit information to other agents within a distance $r_{\text{cmm}} \in \mathbb{R}_+$. We remark that the control bound r_{ctr} and the communication radius r_{cmm} are the same for all agents.

We now state the control design problem of interest.

Problem 3.1 (Connectivity Maintenance): Choose a state dependent graph $\mathcal{G}_{\text{target}}$ on $\mathbb{R}^{2d}$ and design (state dependent) control constraints sets with the following property: if each agent's control takes values in the control constraint set, then the agents move in such a way that the number of connected components of $\mathcal{G}_{\text{target}}$ (evaluated at the agents' states) does not increase with time. $\square$

This objective is to be achieved with the limited information available through message exchanges between agents.

B. A known result for agents with first-order dynamics

In [1], a connectivity constraint was developed for a set of agents modeled by first-order discrete-time dynamics:

$$p_i[t+1] = p_i[t] + u_i[t].$$

Here the graph whose connectivity is of interest, is the disk graph $\mathcal{G}_{\text{disk}}(r_{\text{cmm}})$ over the vertices $\{p_1[t], \dots, p_n[t]\}$. Network connectivity is maintained by restricting the allowable motion of each agent. In particular, it suffices to restrict the motion of each agent as follows. If agents i and j are neighbors in the r_{cmm}-disk graph $\mathcal{G}_{\text{disk}}(r_{\text{cmm}})$ at time t, then their positions at time $t+1$ are required to belong to $\overline{B}\Big(\frac{p_i[t]+p_j[t]}{2}, \frac{r_{\text{cmm}}}{2}\Big)$. In other words, connectivity between i and j is maintained if

$$\begin{aligned} u_i[t] &\in \overline{B}\Big(\frac{p_j[t]-p_i[t]}{2}, \frac{r_{\text{cmm}}}{2}\Big), \\ u_j[t] &\in \overline{B}\Big(\frac{p_i[t]-p_j[t]}{2}, \frac{r_{\text{cmm}}}{2}\Big). \end{aligned}$$

The constraint is illustrated in Figure 3.

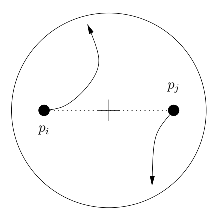

Fig. 3. Starting from p_i and p_j, the agents are restricted to move inside the disk centered at $\frac{p_i+p_j}{2}$ with radius $\frac{r_{\text{cmm}}}{2}$.

Note that this constraint takes into account only the positions of the agents; this fact can be attributed to the *first-order dynamics* of the agents. We wish to construct a similar constraint for agents with second order dynamics. It is reasonable to expect that this new constraint will depend on positions as well as velocities of the neighboring agents.

C. An appropriate graph for the connectivity maintenance problem for agents with second-order dynamics

We begin working on the stated problem with a negative result regarding two candidate target graphs.

Lemma 3.2: Consider a network of n agents with double integrator dynamics (6) in $\mathbb{R}^d$. Let r_{cmm} be the communication range and let r_{ctr} be the control bound. Let $\mathcal{G}_{\text{target}}$ be either $\mathcal{G}_{\text{disk}}(r_{\text{cmm}})$ on $\mathbb{R}^{2d}$ or $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, 2r_{\text{ctr}})$. There exist states $\{(p_i, v_i)\}_{i\in\{1,\dots,n\}}$ such that

(i) the graph $\mathcal{G}_{\text{target}}$ at $\{(p_i, v_i)\}_{i\in\{1,\dots,n\}}$ is connected, and

(ii) for all $\{u_i\}_{i\in\{1,\dots,n\}} \subseteq \overline{B}(0_d, r_{\text{ctr}})$, the graph $\mathcal{G}_{\text{target}}$ at $\{(p_i + v_i, v_i + u_i)\}_{i\in\{1,\dots,n\}}$, is disconnected.

Remark 3.3: The result in Lemma 3.2 on the double integrator graph has the following interpretation. Assume that agent i has two neighbors j and k in the graph $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, r_{\text{ctr}})$. By definition of the neighboring law for this graph, we know that there exists bounded controls for i and j to maintain the $((p_i, v_i), (p_j, v_j))$ link and that there exists bounded controls for i and k to maintain the $((p_i, v_i), (p_k, v_k))$ link. The lemma states that, for some states of the agents i, j, and k, there might not exist controls that maintain both links simultaneously. □

The following theorem suggests that an appropriate scaling of the control bound is helpful in identifying a suitable state dependent graph for Problem 3.1.

Theorem 3.4: [A scaled double-integrator disk graph is suitable] Consider a network of n agents with double integrator dynamics (6) in $\mathbb{R}^d$. Let r_{cmm} be the communication range and let r_{ctr} be the control bound. Take

$$\nu \in \left]0, \frac{2}{\sqrt{d}(n-1)}\right].$$

Then the following statements hold:

(i) For all states $\{(p_i, v_i)\}_{i\in\{1,\dots,n\}}$ such that the graph $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu r_{\text{ctr}})$ is connected at $\{(p_i, v_i)\}_{i\in\{1,\dots,n\}}$, there exists $\{u_i\}_{i\in\{1,\dots,n\}} \subseteq \overline{B}(0_d, r_{\text{ctr}})$, such that the graph $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu r_{\text{ctr}})$ is also connected at $\{(p_i + v_i, v_i + u_i)\}_{i\in\{1,\dots,n\}}$.

(ii) Let T be a spanning tree of $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu r_{\text{ctr}})$ at $\{(p_i, v_i)\}_{i\in\{1,\dots,n\}}$. Then, there exists $\{u_i\}_{i\in\{1,\dots,n\}} \subseteq \overline{B}(0_d, r_{\text{ctr}})$, such that, for all edges $((p_i, v_i), (p_j, v_j))$ of T, it holds that $((p_i+v_i, v_i+u_i), (p_j+v_j, v_j+u_j))$ is an edge in $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu r_{\text{ctr}})$ at $\{(p_i + v_i, v_i + u_i)\}_{i\in\{1,\dots,n\}}$.

These results are based upon Shostak's Theory for systems of inequalities, as exposed in [7]. We postpone the proof to a forthcoming technical report.

Remark 3.5: [One-hop distributed computation of connectivity] Each agent can compute its neighbors in the graph $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu r_{\text{ctr}})$ just by communicating with its neighbors in $\mathcal{G}_{\text{disk}}(r_{\text{cmm}})$ and exchanging with them position and velocity information. □

Remark 3.6: If $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu r_{\text{ctr}})$ at $\{(p_i, v_i)\}_{i\in\{1,\dots,n\}}$ is not connected, then the above theorem applies to its connected components. Without loss of generality, we assume this graph to be connected in what follows. □

D. The control constraint set and its polytopic representation

We now concentrate on two agents with indices i and j. For $t \in \mathbb{N}_0$, we define the relative position, velocity and control by $p_{ij}[t] = p_i[t] - p_j[t]$, $v_{ij}[t] = v_i[t] - v_j[t]$ and $u_{ij}[t] = u_i[t] - u_j[t]$, respectively. It is easy to see that

$$\begin{aligned} p_{ij}[t+1] &= p_{ij}[t] + v_{ij}[t], \\ v_{ij}[t+1] &= v_{ij}[t] + u_{ij}[t]. \end{aligned}$$

Assume that agents i, j are connected in $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu r_{\text{ctr}})$ at time t. By definition, this means that the relative state $(p_{ij}[t], v_{ij}[t])$ belongs to $\mathcal{A}^d(r_{\text{cmm}}, \nu r_{\text{ctr}})$. If this connection is to be maintained at time $t+1$, then the relative control at time t must satisfy

$$u_i[t] - u_j[t] \in \mathcal{U}^d(r_{\text{cmm}}, \nu r_{\text{ctr}}) \cdot (p_{ij}[t], v_{ij}[t]). \quad (7)$$

Also, implicit are the following bounds on individual control inputs $u_i[t]$ and $u_j[t]$:

$$u_i[t] \in \overline{B}(0_d, r_{\text{ctr}}), \quad u_j[t] \in \overline{B}(0_d, r_{\text{ctr}}). \quad (8)$$

This discussion motivates the following definition.

Definition 3.7: Given $r_{\text{cmm}}, r_{\text{ctr}}, \nu \in \mathbb{R}_+$ and given a set E of edges in $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu r_{\text{ctr}})$ at $\{(p_i, v_i)\}_{i\in\{1,\dots,n\}}$, the *control constraint set* is defined by

$$\begin{aligned} &\mathcal{U}_E^d(r_{\text{cmm}}, r_{\text{ctr}}, \nu) \cdot (\{p_i, v_i\}_{i\in\{1,\dots,n\}}) \\ &= \{(u_1, \dots, u_n) \in \overline{B}(0_d, r_{\text{ctr}})^n \mid \forall((p_i, v_i), (p_j, v_j)) \in E, \\ &\qquad u_i - u_j \in \mathcal{U}^d(r_{\text{cmm}}, \nu r_{\text{ctr}}) \cdot (p_i - p_j, v_i - v_j)\}. \end{aligned}$$

Remark 3.8: The control constraint set for an edge set E is the set of controls for each agent with the property that all edges in E will be maintained in one time step. □

Remark 3.9: We can now interpret the results in Theorem 3.4 as follows.

(i) To maintain connectivity between any pair of connected agents, we should simultaneously handle constraints corresponding to *all* edges of $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu r_{\text{ctr}})$. This might render the control constraint set empty.

(ii) However, if we only consider constraints corresponding to edges belonging to *a* spanning tree T of $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu r_{\text{ctr}})$, then the set $\mathcal{U}_T^d(r_{\text{cmm}}, \nu r_{\text{ctr}}) \cdot (\{p_i, v_i\}_{i\in\{1,\dots,n\}})$ is guaranteed to be nonempty. □

Let us now provide a concrete representation of the control constraint set. Given a pair i, j of connected agents, the admissible control set $\mathcal{U}^d(r_{\text{cmm}}, \nu r_{\text{ctr}}) \cdot (p_{ij}, v_{ij})$ is convex and compact (Lemma 2.2). Hence, we can fit a polytope with N_{poly} sides inside it. This approximating polytope leads to the following tighter version of the constraint in (7):

$$(C_{ij}^\eta)^T (u_i - u_j) \le w_{ij}^\eta, \qquad \eta \in \{1, \dots, N_{\text{poly}}\}, \quad (9)$$

for some appropriate vector $C_{ij}^\eta \in \mathbb{R}^d$ and scalar $w_{ij}^\eta \in \mathbb{R}$. Similarly, one can compute an inner polytopic approximation of the ball $\overline{B}(0_d, r_{\text{ctr}})$ and write the following linear vector inequalities:

$$(C_{i\theta}^\eta)^T u_i \le w_{i\theta}^\eta, \qquad \eta \in \{1, \dots, N_{\text{poly}}\}, \quad (10)$$

where the symbol θ has the interpretation of a fictional agent.

In this way, we have cast the original problem of finding a set of feasible control inputs into a satisfiability problem for a set of linear inequalities.

Remark 3.10: Rather than a network-wide control constraint set, one might like to obtain decoupled constraint sets for each individual agent. However, (1) it is not clear how to design a distributed algorithm to perform this computation,

(2) such an algorithm will likely have large communication requirements, and (3) such a calculation might lead to a very conservative estimate for the decoupled control constraint sets. Therefore, rather than explicitly decoupling the control constraint sets, we next focus on a distributed algorithm to search the control constraint set for feasible controls that are optimal according to some criterion. □

IV. Distributed computation of optimal controls

In this section we formulate and solve the following optimization problem: given an array of desired control inputs $U_{\text{des}} = (u_{\text{des},1}, \dots, u_{\text{des},n})^T \in (\mathbb{R}^d)^n$, find, via local computation, the array $U = (u_1, \dots, u_n)$ belonging to the control constraint set, that is *closest* to the desired array U_{des}. To formulate this problem precisely, we need some additional notations. Let E be a set of edges in the undirected graph $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu r_{\text{ctr}})$ at $\{(p_i, v_i)\}_{i\in\{1,\dots,n\}}$. To deal with the linear inequalities of the form (9) and (10) associated to each edge of E, we introduce an appropriate *multigraph*. A *multigraph* (or *multiple edge graph*) is, roughly speaking, a graph with multiple edges between the same vertices. More formally, a multigraph is a pair (V, E), where V is the vertex set and the edge set E contains numbered edges of the form (i, j, η), for $i, j \in V$ and $\eta \in \mathbb{N}$, and where E has the property that if $(i, j, \eta) \in E$ and $\eta > 1$, then also $(i, j, \eta-1) \in E$. Two edges e_1 and e_2 of a multigraph are *parallel* if $e_1 = (v_i, v_j, \eta_1)$ and $e_2 = (v_i, v_j, \eta_2)$, i.e., if they share the same vertices. With this definition in hand, let $E_{\text{mult}} = \{(i, j, \eta) \in \{1, \dots, n\}^2 \times \{1, \dots, N_{\text{poly}}\} \mid ((p_i, v_i), (p_j, v_j)) \in E,\ i > j\}$ and define $G_{\text{mult}} = (\{1, \dots, n\}, E_{\text{mult}})$. Note that to each element $(i, j, \eta) \in E_{\text{mult}}$ is associated the inequality $(C^\eta_{ij})^T (u_i - u_j) \le w^\eta_{ij}$. We are now ready to formally state the optimization problem at hand in the form of the following quadratic programming problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\sum_{i=1}^{n} \|u_i - u_{\text{des},i}\|^2, \\ \text{subj. to} \quad & (C^\eta_{ij})^T (u_i - u_j) \le w^\eta_{ij}, \ \text{ for } (i, j, \eta) \in E_{\text{mult}}, \\ & (C^\eta_{i\theta})^T u_i \le w^\eta_{i\theta}, \\ & \qquad \text{for } i \in \{1, \dots, n\}, \eta \in \{1, \dots, N_{\text{poly}}\}. \end{aligned} \tag{11}$$

Here, somehow arbitrarily, we have adopted the 2-norm to define the cost function.

Remark 4.1: If E is a spanning tree of $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu r_{\text{ctr}})$ at a connected configuration $\{(p_i, v_i)\}_{i\in\{1,\dots,n\}}$, then the control constraint set $\mathcal{U}^d_E(r_{\text{cmm}}, r_{\text{ctr}}, \nu) \cdot (\{p_i, v_i\}_{i\in\{1,\dots,n\}})$ is guaranteed to be non-empty by Theorem 3.4. In turn, this implies that the optimization problem (11) is feasible. □

The above problem can be written in a compact form as:

$$\begin{aligned} \text{minimize} \ & \frac{1}{2}\|U - U_{\text{des}}\|^2, \\ \text{subj. to} \ & B^T_{\text{mult}} U \preceq w, \end{aligned} \tag{12}$$

for appropriately defined matrix B_{mult} and vector w.

A. Solution via duality: Linearized projected Jacobi method

The quadratic programming problem stated in (12) can be written into its equivalent standard form as:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}U^T U - U^T_{\text{des}} U, \\ \text{subj. to} \quad & B^T_{\text{mult}} U \preceq w. \end{aligned}$$

The solution to this problem relies on using duality theory and is obtained by employing a technique which is known as the *linearized projected Jacobi method* in the literature on network flow control problems ([8], Section 3.4). Accordingly, let λ^* be the value of Lagrange multipliers at optimality. Then the global minimum for U is achieved at

$$U^* = U_{\text{des}} - B_{\text{mult}}\lambda^*. \tag{13}$$

The linearized projected Jacobi iteration for each component of λ is given by

$$\begin{aligned} \lambda_\alpha(t+1) = \max\Big\{ & \lambda_\alpha(t) - \frac{\tau}{(B^T_{\text{mult}} B_{\text{mult}})_{\alpha\alpha}} \cdot \\ & \Big((B^T_{\text{mult}} U_{\text{des}} - w)_\alpha + \sum_{\beta=1}^{N_{\text{poly}}(e+n)} (B^T_{\text{mult}} B_{\text{mult}})_{\alpha\beta}\lambda_\beta(t)\Big), 0\Big\}, \end{aligned} \tag{14}$$

where $\alpha \in \{1, \dots, N_{\text{poly}}(e + n)\}$ and τ is the step-size parameter. We can select $\tau = \frac{1}{N_{\text{poly}}(e+n)}$ to guarantee convergence.

B. A distributed implementation of the dual algorithm

Because of the particular structure of the matrix $B^T_{\text{mult}} B_{\text{mult}}$, the iterations (14) can be implemented in a distributed way over the original graph G. To highlight the distributed structure of the iteration we denote the components of λ by referring to the nodes that they share and the inequality they are related to. In particular for each edge in G_{mult}, the corresponding Lagrange multiplier will be denoted as λ^η_{ij} if the edge goes from node i to node j, $i > j$, and it is associated to the inequality constraint $C^\eta_{ij}(u_i - u_j) \le w^\eta_{ij}$. This makes up the first $N_{\text{poly}}e$ entries of the vector λ. To be consistent with this notation, the next $N_{\text{poly}}n$ entries will be denoted as $\lambda^1_{1\theta}, \dots, \lambda^{N_{\text{poly}}}_{1\theta}, \dots, \lambda^1_{n\theta}, \dots, \lambda^{N_{\text{poly}}}_{n\theta}$. Additionally, define $\mathcal{N}(i) = \{j \in \{1, \dots, n\} \mid \{p(i), p(j)\} \in E\} \cup \{\theta\}$. The symbol θ has the interpretation of a fictional node.

Defining $\lambda^\eta_{ij} := \lambda^\eta_{ji}$ for $i < j$, we can write equations (13) and (14) in components as follows. Equation (13) reads, for $i \in \{1, \dots, n\}$,

$$u^*_i = u_{\text{des},i} + \sum_{k\in\mathcal{N}(i)} \sum_{\eta=1}^{N_{\text{poly}}} C^\eta_{ik}\lambda^\eta_{ik}. \tag{15}$$

One can easily work an explicit expression for matrix product $B^T_{\text{mult}} B_{\text{mult}}$ in (14). Then, equation (14) reads, for

$(i,j,\eta) \in E_{\text{mult}}$,

$$\lambda_{ij}^{\eta}(t+1) = \max\Bigg\{0, \lambda_{ij}^{\eta}(t) - \frac{\tau}{2(C_{ij}^{\eta})^T C_{ij}^{\eta}} \cdot$$

$$\Bigg(\sum_{k\in\mathcal{N}(i)} \sum_{\sigma=1}^{N_{\text{poly}}} \Big((C_{ij}^{\eta})^T C_{ik}^{\sigma}\lambda_{ik}^{\sigma}\Big) + \sum_{k\in\mathcal{N}(j)} \sum_{\sigma=1}^{N_{\text{poly}}} \Big((C_{ij}^{\eta})^T C_{jk}^{\sigma}\lambda_{jk}^{\sigma}\Big)$$

$$+ (C_{ij}^{\eta})^T(u_{\text{des},i} - u_{\text{des},j}) - w_{ij}^{\eta}\Bigg)\Bigg\},$$

together with, for $i = \{1,\dots,n\}$, $\eta = \{1,\dots,N_{\text{poly}}\}$

$$\lambda_{i\theta}^{\eta}(t+1) = \max\Big\{0, \lambda_{i\theta}^{\eta}(t)$$

$$- \frac{\tau}{(C_{i\theta}^{\eta})^T C_{i\theta}^{\eta}}\Big(\sum_{k\in\mathcal{N}(i)} \sum_{\sigma=1}^{N_{\text{poly}}}(C_{ik}^{\sigma}\lambda_{ik}^{\sigma}) + (C_{i\theta}^{\eta})^T u_{\text{des},i} - w_{i\theta}^{\eta}\Big)\Big\},$$

Remark 4.2: We distribute the task of running iterations for these $N_{\text{poly}}(e+n)$ Lagrange multipliers among the n agents as follows: an agent i carries out the updates for all quantities $\lambda_{i\theta}^{\eta}$ and all λ_{ij}^{η} for which $i > j$. By means of this partition and one-hop communication, we find the global solution for the optimization problem (11) in a distributed fashion over the double integrator disk graph. □

V. Simulations

To illustrate our analysis we focus on the following scenario. For the two dimensional setting, i.e. for $d = 2$, we assume that there are $n = 5$ agents with (randomly chosen) initial condition and such that they are connected according to $\mathcal{G}_{\text{di-disk}}$. The bound for the control input is $r_{\text{ctr}} = 2$ and the communication radius is $r_{\text{cmm}} = 10$.

We assigned to one of the agents a derivative feedback control $u_x[p,v] = (v_x - 2)$, $u_y[p,v] = (v_y - 5)$ as desired input. For the other agents the desired input is set to zero. The interesting aspect of this simulation, which opens the perspective to new results in the area of flocking, is that the maintenance of connectivity can lead to the accomplishments of a coordination task as the flocking. We show the velocities (v_x and v_y) of the agents with respect to time, see Figure 4, and the distances between agents which are neighbors in the spanning tree, see Figure 5. Notice that the agents flock reaching a configuration in which all of them are at the limit distance $r_{\text{cmm}} = 10$.

VI. Conclusion

We provide some distributed algorithms to enforce connectivity among networks of agents with double-integrator dynamics. Future directions of research include (i) evaluating the communication complexity of the proposed distributed dual algorithm and possibly designing faster ones, (ii) studying the relationship between the connectivity maintenance problem and the platooning and mesh stability problem, and (iii) investigating the flocking phenomenon and designing flocking algorithms which do not rely on a blanket assumption of connectivity.

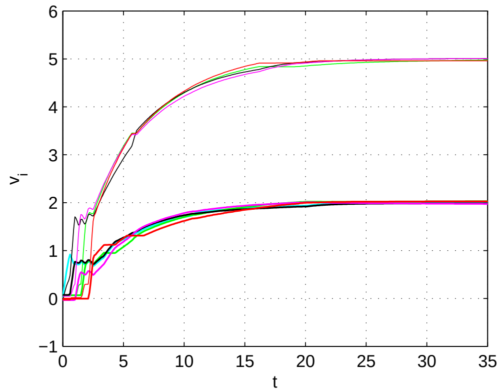

Fig. 4. Velocities (v_x and v_y) of 5 agents for the flocking task

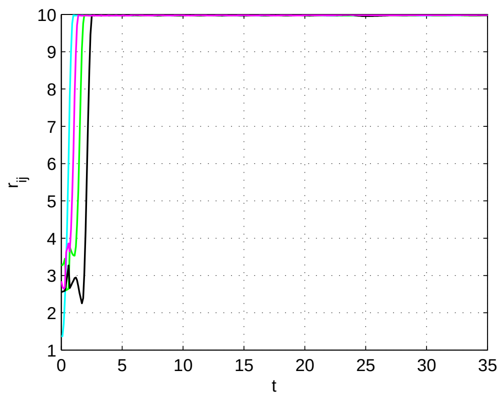

Fig. 5. Inter-agent distances of neighbors in the spanning tree for the flocking task

Acknowledgments

This material is based upon work supported in part by ONR YIP Award N00014-03-1-0512, ARO MURI Award W911NF-05-1-0219 and RECSYS Project IST-2001-37170. Giuseppe Notarstefano and Francesco Bullo thank Ruggero Frezza for his kind support.

References

[1] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, "Distributed memoryless point convergence algorithm for mobile robots with limited visibility," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 5, pp. 818–828, 1999.

[2] J. Lin, A. S. Morse, and B. D. O. Anderson, "The multi-agent rendezvous problem," in *IEEE Conf. on Decision and Control*, (Maui, HI), pp. 1508–1513, Dec. 2003.

[3] J. Cortés, S. Martínez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Transactions on Automatic Control*, vol. 51, no. 6, 2006. To appear.

[4] A. Ganguli, J. Cortés, and F. Bullo, "On rendezvous for visually-guided agents in a nonconvex polygon," in *IEEE Conf. on Decision and Control*, (Seville, Spain), pp. 5686–5691, Dec. 2005.

[5] D. P. Spanos and R. M. Murray, "Motion planning with wireless network constraints," in *American Control Conference*, (Portland, OR), pp. 87–92, June 2005.

[6] Z. Gao, "On discrete time optimal control: A closed-form solution," in *American Control Conference*, (Boston, MA), pp. 52–58, 2004.

[7] B. Aspvall and Y. Shiloach, "A polynomial time algorithm for solving systems of linear inequalities with two variables per inequality," *SIAM Journal on Computing*, vol. 9, no. 4, pp. 827–845, 1980.

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