2018 Annual American Control Conference (ACC)
June 27–29, 2018. Wisconsin Center, Milwaukee, USA

A Connectivity Preserving Framework for Distributed Motion Coordination in Proximity Networks

Hassan Jaleel and Waseem Abbas

***Abstract*— A necessary condition for global performance guarantees in most of the motion coordination algorithms is connectivity of the underlying network topology. In proximity networks, connectivity maintenance becomes critical because the neighborhood set of each agent is dynamic and depends on the locations of all the other agents in the network. We present an efficient framework for distributed motion coordination in proximity networks. The proposed framework relies on identifying agents in a so called weakly connected dominating (WCD) set of the underlying network graph. Maintaining only the edges incident to the agents in WCD, which we call as critical edges, preserves the connectivity of the overall network. The proposed framework is presented in the context of rendezvous problem, which is selected because of its canonical importance in distributed systems with mobile agents. We propose a controller that drives all the agents to a common point by preserving the critical edges only. The proposed scheme is robust to failure of edges that are not critical and nodes that do not belong to WCD. Moreover, it performs well in terms of energy consumption and computational complexity.**

I. INTRODUCTION

In the distributed control and coordination of multiagent systems, maintaining connectivity among agents is one of the primary requirements. In particular, success of various distributed algorithms employed towards achieving formation control, coverage control, and boundary protection deeply depends on the connectivity of the underlying network structure. Moreover, despite the algorithmic advances in coordinating multiagent systems, issues related to energy efficiency and real time implementability of these algorithms are significant and still require thorough investigation.

We propose an efficient framework for multiagent systems comprising mobile robots that ensures connectivity of the underlying network topology in proximity networks. In these networks, two agents are neighbours if and only if they are within the sensing range of each other, which results in a dynamic neighborhood set for each agent. Therefore, maintaining connectivity in proximity graphs requires additional measures. In the proposed framework, we first identify a set of nodes that constitute a *weakly connected dominating (WCD) set* in the underlying network graph. We establish that maintaining only the edges incident to the nodes in WCD, which we refer to as critical edges, preserve the connectivity of the overall network. This weakly connected dominating set is computed offline before the deployment of the system. In order to find a small WCD set, we refer to the well-established theory of domination in graphs (see e.g. [11]), in particular to the weakly connected dominating sets in graphs (see e.g., [13], [14], [15], [16], [17]).

We also design an energy efficient controller that ensures that all the critical edges are maintained, and guarantees a desired global objective. The controller is designed for the consensus problem, in which the objective is to ensure that all agents converge at a common state, which is a position here. Consensus-based controllers have been extensively studied and applied to solve problems related to formation control, coverage control, and distributed estimation to name a few (e.g., [1], [2], [7]). We present a detailed stability analysis for the proposed controller through which we prove that the proposed controller drives all the agents to a common point while guaranteeing that all the critical edges are maintained for all the time. Moreover, we verify the performance of the proposed framework via simulations. The simulations also highlight the energy efficiency and real time implementability of the proposed framework as compared to the existing schemes.

In the existing literature, different schemes have been proposed to ensure connectivity. In [10], the authors proposed a centralized scheme to maximize the second smallest eigenvalue λ_2, called the algebraic connectivity of a graph, using semi definite programming. A graph with N nodes is connected as long as $\lambda_2 > 0$. A distributed solution for this problem was presented in [5]. Based on the same concept, a new scheme was proposed in [8], where each agent gets its estimate of λ_2 in a distributed manner. Edges are allowed to break if the value of λ_2 is greater than some threshold value. The problem with these schemes is their high computational complexity since an optimization problem has to be solved at each decision time either by a centralized authority or by individual agents. This computational complexity severely limits the real time implementability of these algorithms.

Another approach is based on potential functions in which an energy or a potential is associated with each edge to prevent edge loss (see [2], [3], and [4]). This approach is computationally efficient but conservative in a sense that it does not allow any edge to break. Consequently, the motion of the agents can be overly restrictive and the agents are often forced to move in an aggressive manner. Moreover, in the case of rendezvous problem, the constraints on each agent increases further with the formation of each new edge as the agents move close to each other. In this approach, convergence to a rendezvous point can be fast but this fast convergence rate is achieved at a price of excessive energy

H. Jaleel is with the King Abdullah University of Science and Technology (KAUST), Computer, Electrical and Mathematical Sciences and Engineering Division (CEMSE), Thuwal 23955–6900, Saudi Arabia. W. Abbas is with the Department of Electrical Engineering at Information Technology University (ITU), Lahore, Pakistan. Emails: hassan.jaleel@kaust.edu.sa, w.abbas@itu.edu.pk.

consumption.

Our scheme provides a compromise between the above approaches. We also add edge weights but only to the set of critical edges. These are the edges incident to the nodes belonging to the WCD set. Loss of an edge which is not critical, or a node which does not belong to the WCD set does not affect the connectivity of the network. Hence, it is no longer required to maintain all edges. Since a WCD set has to be computed only once before the deployment of the system, the proposed framework is real time implementable. Moreover, we show through simulations that the proposed controller is energy efficient as compared to the scheme in [2] because of the reduced number of constraints on each agent. Moreover the transmission power level of each agent are adjusted to maintain only the required connections, reducing the energy consumption even further.

One final remark regarding the novelty of the proposed framework. We note that the minimum number of edges that are needed to maintain the connectivity of the underlying network of n nodes is $(n-1)$. In particular, these are the edges that are included in a minimum spanning tree. However, in that case, a node has to identify a subset of its neighbors with which it will maintain edges. In the proximity model for network topology, an agent does not have the authority to select its neighbors. An agent j is a neighbor of agent i if the distance between them is less then the sensing radius of agent i. Our framework handles this restriction of proximity networks. In our model, if a node is responsible for maintaining edges, i.e., if it is included in a WCD set, then it preserves edges with all its neighbors. Thus, we eliminate the requirement to explicitly distinguish between neighbors for the purpose of keeping a track of edges that are included in a spanning tree.

The outline of this paper is as follows. Section II presents graph theoretic concepts used in the paper and describes the system model. Section III discusses the notion of WCD set. Section IV presents the controller for the rendezvous problem. Section V provides a numerical evaluation of the energy consumption, and Section VI concludes the paper.

II. Preliminaries and System Description

Here, we introduce the terms and notations that will be used throughout the paper. We represent the underlying network structure by a graph $\mathcal{G}(V,E)$, in which V is the vertex set, and E is the edge set. The graph is undirected if the edge set is unordered, i.e., $(v_i \sim v_j) \in E \Leftrightarrow (v_j \sim v_i) \in E$, and the undirected graph is *connected* if a path exists between any two vertices. If the edges are directed, we obtain a directed graph, for which we define an *incidence matrix* $\mathcal{I} = [e_{ij}]$ as follows.

$$e_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is the head of the edge } e_j \\ -1 & \text{if } v_i \text{ is the tail of the edge } e_j \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathcal{G}(V,E)$ be an undirected graph, and we assign an arbitrary orientation to its edges to obtain a directed graph. If $\mathcal{I}$ be the incidence matrix of the new directed graph, then we define the Laplacian matrix of the undirected $\mathcal{G}(V,E)$ as $\mathcal{L} = \mathcal{I}\mathcal{I}^T$. We denote the eigen values of $\mathcal{L}$ by $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$. Note that λ_1 is always zero and $\mathbf{1} = [1\ 1\ \cdots\ 1]^T$ is the corresponding eigen vector. Moreover, the number of connected components in $\mathcal{G}$ is exactly same as the multiplicity of zero eigen values of its Laplacian matrix $\mathcal{L}$ (e.g., see [6]).

The standard dynamics to solve the consensus problem are given by the following consensus equation.

$$\dot{x}_i = -\sum_{v_j \in \mathcal{N}(v_i)} (x_i - x_j). \tag{1}$$

Here $\mathcal{N}(v_i)$ is the neighborhood set of node v_i define as $\mathcal{N}(v_i) = \{v_j | (v_i \sim v_j) \in E\}$. We define the close neighborhood of a node v_i as $\mathcal{N}[v_i] = \{v_i \cup \mathcal{N}(v_i)\}$. We consider N agents in $\mathbb{R}^n$ and denote the location of agent i by $x_i^T = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$. Then the dynamics in (1) can be represented in matrix form as

$$\dot{x} = -\mathcal{L}x. \tag{2}$$

where $\mathcal{L}$ is the laplacian matrix corresponding to the graph $\mathcal{G}$.

We consider multiagent systems with planar agents, that is $x_i \in \mathbb{R}^2$ for each v_i. We define the region in which an agent can sense some event and communicate with others as the *sensor footprint* or the *sensing region*. Since the neighbors of an agent could change as a result of a change in its location, we also study the so-called *dynamic graphs*, in which the edge set changes as agents move in and out of each others' sensing regions. In particular, we study Δ-disk graphs, $\mathcal{G}(V,E,\Delta)$, where $(v_i, v_j) \in E \Leftrightarrow \|x_i - x_j\| \leq \Delta$, for some given $\Delta > 0$. The neighborhood set of such graphs is defined as $\mathcal{N}(v_i) = \{v_j | \|x_i - x_j\| \leq \Delta\}$. If Δ is same for all the agents, then the resulting graph is undirected. We also consider the case in which agents might have different sensing radii resulting in a *directed disk graphs*, where $(v_i, v_j) \in E \Leftrightarrow \|x_i - x_j\| < \Delta_j$.

III. Connectivity Maintenance through Weakly Connected Dominating (WCD) Sets

We start with a group of N planar agents with $\mathcal{G}(V,E,\Delta)$ representing their underlying interaction topology. Our proposed scheme is based on identifying a subset of nodes in a network such that maintaining all the edges incident to these nodes will preserve the connectivity of the overall network. This means instead of maintaining all the edges, as in [2], we only need to maintain a subset of the edges. The subset of nodes is the one that constitute a *Weakly Connected Dominating (WCD)* set in the graph. The notions of dominating sets and its variants have been extensively applied in the domain of wireless sensor and mobile ad hoc networks to exploit the network structure to design energy-efficient efficient routing and clustering schemes (e.g. [18]). A dominating set in a graph is defined as below.

Definition 3.1: For a graph $\mathcal{G}(V, E, \Delta)$, a set of nodes $S \subseteq V$ is a dominating set if and only if $\bigcup_{v_i \in S} \mathcal{N}[v_i] = V$.

Note that for each dominating node v of degree r, we can associate a star graph, $K_{1,r+1}$, in $\mathcal{G}$ where a dominating node v is a central vertex and vertices in $\mathcal{N}(v)$ are the branches. The union of all such stars is a graph, $\mathcal{G}_D \subseteq \mathcal{G}$ with a vertex set V and an edge set $E_D \subseteq E$ where, $E_D = \{v \sim u : v \in D\}$. This $\mathcal{G}_D$ will necessarily span $\mathcal{G}$ by the definition of a dominating set, but it may not be connected as shown in the Figure 1(b). The goal is to extend the dominating set of $\mathcal{G}$ such that the union of star graphs obtained from the nodes in D results into a connected spanning subgraph of $\mathcal{G}$. Such a dominating set is referred to as the *Weakly Connected Dominating* set.

Definition 3.2: A set of nodes D is a *Weakly Connected Dominating (WCD)* set if and only if

(a) D is a dominating set

(b) A graph with a vertex set V and an edge set $E_d \subseteq E$, where $E_D = \{v \sim u : \ v \in D\}$ is a connected spanning subgraph of $\mathcal{G}$.

Definition 3.3: An edge $(v_i \sim v_j)$ is critical if and only if $v_i \in D$.

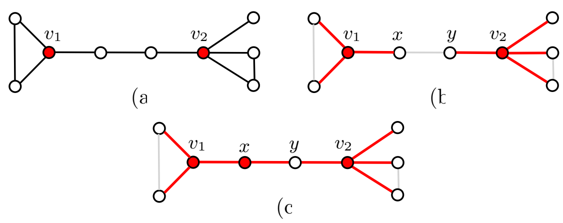


Fig. 1. (a) A graph with a dominating set, $D = \{v_1, v_2\}$. (b) A subgraph $\mathcal{G}_D = \mathcal{G}_{v_1} \cup \mathcal{G}_{v_2}$, where $\mathcal{G}_{v_i}$ is a star graph with a vertex set $\mathcal{N}[v_i]$ and edge set consisting of all edges incident to v_i. Note that $\mathcal{G}_d$ is not connected. (c) vertex x is added to a dominating set, thus giving a connected graph, $\bigcup_{k \in \{v_1, v_2, x\}} \mathcal{G}_k$, that spans original $\mathcal{G}$.

We note that the WCD set, as the name suggests, is a weaker notion as compared to the more widely used concept of *connected dominating (CD)* set. A set of nodes constitute a CD set if they also form a dominating set, and the subgraph induced by the vertices in CD is connected. In the case of WCD, the subgraph induced by the vertices in WCD may not be connected, but the subgraph with a vertex set V and the edge set E_D containing only those edges of $E(\mathcal{G})$ that originates from the vertices in WCD, is connected.

The problem of finding a minimum sized WCD set is NP-hard [12]. In [13], Chen and Liestman proposed centralized algorithms having an approximation ratio $O(\log d_{\max})$ to compute WCD sets. Here, $d_{\max}$ is the maximum degree of the graph. They also provided distributed implementation of their algorithms. In [14], the same authors proposed another algorithm based on the idea of dividing the whole graph into various regions, computing WCD set for each region, and then making adjustments to construct the WCD of the whole graph. Alzoubi et al. proposed an algorithm in [15] that ran in $O(n)$ time, required a message complexity of $O(n \log n)$, and had a constant approximation ratio of 5. Another distributed algorithm with a constant approximation ratio was presented in [17], whose time and message complexities were both $O(n)$. Some other distributed algorithms for a small sized WCD sets were also presented in [16].

An example of a dominating set and a weakly connected dominating set are illustrated in Figure 2. A set of nodes $D = \{v_1, v_4, v_5, v_7, v_{13}\}$ is a dominating set as $\cup_{v_i \in D} \mathcal{N}[v_i] = V$. If E_D is the set of edges incident on the vertices in D, then the subgraph consisting of vertices V and edges E_D is not connected. In fact, it has three components as shown in Figure 2(a). On the other hand, if we consider $D' = D \cup \{v_3, v_{10}\}$, then D' is a WCD as $\cup_{v_i \in D} \mathcal{N}[v_i] = V$, and the subgraph with the vertex set V and the edge set containing edges that are incident on the vertices in D' only, is connected, as shown in Figure 2(b).

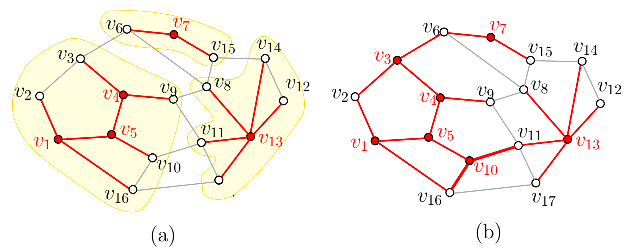


Fig. 2. (a) A graph with a dominating set $D = \{v_1, v_4, v_5, v_7, v_{13}\}$. (b) A graph with a weakly connected dominating set $D' = D \cup \{v_3, v_{10}\}$.

IV. Rendezvous With Connectivity Maintenance Dominating Set

In this section, we design a control law to solve the rendezvous problem using WCD sets. The rendezvous problem is selected because of its fundamental importance to the distributed control of multiagent systems. If the underlying network structure induces an undirected graph that remains connected at all times, the controller in Equation (1) solves the problem. But for Δ-disk graphs, this connectivity condition cannot be guaranteed as shown in [2] and [7]. To address this issue, one approach is to modify Equation (1) by introducing edge weights $w(x_i, x_j)$.

$$\dot{x}_i = -\sum_{v_j \in \mathcal{N}(v_i)} w(x_i, x_j)(x_i - x_j), \tag{3}$$

In the case of an undirected network graph, we consider the edge tension energy for each edge $(v_i \sim v_j)$ as

$$\mathcal{E}_{ij}(x) = \frac{\|x_i - x_j\|^2}{\Delta - \|x_i - x_j\|}. \tag{4}$$

If a graph is initially connected (at $t = 0$), then it was shown in [2] that the controller

$$\dot{x}_i = -\frac{\partial \mathcal{E}(x)}{\partial x_i}^T = -\sum_{v_j \in \mathcal{N}(v_i)} w(x_i, x_j)(x_i - x_j). \tag{5}$$

guarantees that the graph remains connected $\forall t$, while solving the rendezvous problem for Δ-disk proximity graphs.

Although Equation (5) preserves all the edges and solves the rendezvous problem, it is over restrictive. In order to maintain all the edges, agents are forced to move in an aggressive manner at high velocity which is costly in terms of energy consumption. Here, we first propose an energy efficient controller to solve the rendezvous problem using WCD set, denoted by D, and then prove the stability and convergence of the proposed controller. By definition (3.2), a graph $\mathcal{G}$ is connected if all of its edges $(v_i \sim v_j)$ such that $v_i \in D$, exist. The task of the controller is thus to ensure that these edges are preserved for all time. We start with an undirected network where the footprints of all the agents are Δ. For simplicity, we consider each undirected edge $(v_i \sim v_j)$ as a collection of two directed edges. With this in mind, the edge tension energy is defined as

$$\mathcal{E}_{ij}(x) = \begin{cases} \frac{\|x_i - x_j\|^2}{\Delta - \|x_i - x_j\|} & \text{if } v_i \in D \text{ and } (v_i \sim v_j) \in E, \\ \frac{1}{2}\|x_i - x_j\|^2 & \text{if } v_i \in V \backslash D \text{ and } (v_i \sim v_j) \in E, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Lemma 4.1: *If $\mathcal{G}(V, E, \Delta)$ is an undirected Δ-disk graph that is connected at $t = 0$, i.e., $\|x_i(0) - x_j(0)\| < (\Delta - \epsilon)$, for some, $0 < \epsilon < \Delta$ and for all $(v_i, v_j) \in E(0)$; then the graph $\mathcal{G}(V, E, \Delta)$ remains connected $\forall\, t > 0$ under the following control law.*

$$\dot{x}_i = - \sum_{v_j \in \mathcal{N}(v_i)} \frac{\partial \mathcal{E}_{ij}(x)}{\partial x_i}^T, \quad (7)$$

where $\mathcal{E}_{ij}(x)$ is defined in Equation (6).

Proof: The total energy of the system is

$$\mathcal{E}(x) = \sum_{i=1}^{N} \sum_{j=1}^{N} \mathcal{E}_{ij}(x). \quad (8)$$

Then

$$\dot{\mathcal{E}}(x) = \sum_{i=1}^{N} \frac{\partial \mathcal{E}(x)}{\partial x_i} \dot{x}_i,$$

where $\dot{x}_i$ is as in Eq. (7), which can be rewritten as follows.

$$\dot{x}_i = - \frac{\partial \mathcal{E}(x)}{\partial x_i}^T + \sum_{v_j \in \mathcal{N}(v_i)} \frac{\partial \mathcal{E}_{ji}(x)}{\partial x_i}^T,$$

where

$$\frac{\partial \mathcal{E}(x)}{\partial x_i}^T = \sum_{v_j \in \mathcal{N}(v_i)} \frac{\partial \mathcal{E}_{ij}(x)}{\partial x_i}^T + \sum_{v_j \in \mathcal{N}(v_i)} \frac{\partial \mathcal{E}_{ji}(x)}{\partial x_i}^T.$$

$$\dot{\mathcal{E}}(x) = - \left\| \frac{\partial \mathcal{E}(x)}{\partial x} \right\|^2 + \sum_{i=1}^{N} \frac{\partial \mathcal{E}(x)}{\partial x_i} \sum_{v_j \in \mathcal{N}(v_i)} \frac{\partial \mathcal{E}_{ij}(x)}{\partial x_i}^T.$$

Since

$$\left| \sum_{v_j \in \mathcal{N}(v_i)} \frac{\partial \mathcal{E}_{ij}(x)}{\partial x_i}^T \right| < \left| \frac{\partial \mathcal{E}(x)}{\partial x_i}^T \right|,$$

we conclude that

$$\begin{aligned} \dot{\mathcal{E}}(x) &< - \left\| \frac{\partial \mathcal{E}(x)}{\partial x} \right\|^2 + \left\| \frac{\partial \mathcal{E}(x)}{\partial x} \right\|^2, \\ \dot{\mathcal{E}}(x) &< 0. \end{aligned}$$

The above inequality implies that the total energy of the system is always decreasing. We know that initially $\|x_i - x_j\| < \Delta$ for all edges. Therefore, for any critical edge to break, there has to be a time when $\|x_i - x_j\| = \Delta$ making the corresponding edge energy infinite. But we started with a finite energy and the energy is always decreasing. This proves that no critical edge can break and thus connectivity will always be maintained by controller (9). ■

In Lemma (4.1), we have proved that under the proposed controller (7), the network always remains connected. The next step is to show that the controller indeed solves the rendezvous problem.

Theorem 4.2: *Consider an undirected and initially connected Δ-disk graph, in which edge lengths are less than $(\Delta - \epsilon)$ for some $0 < \epsilon < \Delta$. Then, the system asymptotically converges to the weighted initial centroid of the network under the controller in Equation (7).*

Proof: Consider the energy function defined in Equation (6). Then for any agent v_i the controller in (7) is

$$\dot{x}_i = \begin{cases} - \sum_{v_j \in \mathcal{N}(v_i)} \frac{2\Delta - \|x_i - x_j\|}{(\Delta - \|x_i - x_j\|)^2} (x_i - x_j) & \text{if } v_i \in D, \\ - \sum_{v_j \in \mathcal{N}(v_i)} (x_i - x_j) & \text{if } v_i \in V \backslash D\ . \end{cases} \quad (9)$$

Equation (9) shows that for all agents, the controller (7) can be written in the form of weighted consensus equation.

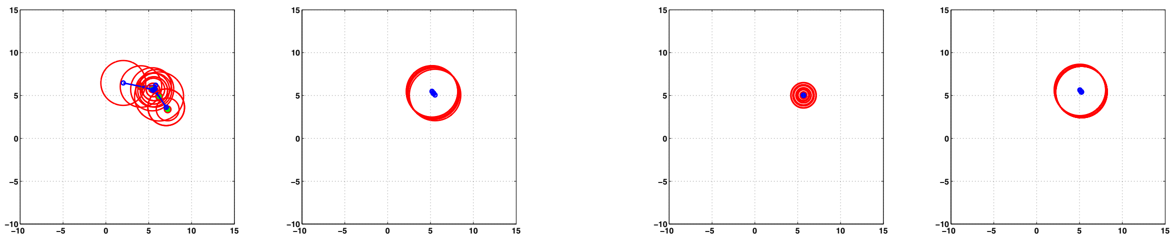
$$\dot{x}_i = - \sum_{v_j \in \mathcal{N}(v_i)} w(x_i, x_j)(x_i - x_j).$$

$$w_{ij} = \begin{cases} \frac{2\Delta - \|x_i - x_j\|}{(\Delta - \|x_i - x_j\|)^2} & \text{if } (v_i \sim v_j) \in E \text{ and } v_i \in D, \\ 1 & \text{if } (v_i \sim v_j) \in E \text{ and } v_i \in V - D, \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

We can also write this controller in terms of a weighted Laplacian.

$$c(x, j) = -\mathcal{L}_w c(x, j) \ \ \forall j = 1, 2 \quad (11)$$

From [9], it is known that the controller (11) drives all the agents asymptotically to the weighted centroid as long as the network stays connected. The connectivity is guaranteed from Lemma 4.1, and the proof follows. ■



(a) Time $t = 1.45$ (b) Time $t = 3$

Fig. 4. Comparison between the proposed scheme (*left* in both plots) and the controller (5) (*right* in both figures) for network in Figure 3. For both cases $N = 15$ and $\Delta = 3$. The controller (5) achieves rendezvous in $t = 1.45$ while our proposed controller achieves it in $t = 3$. The footprints in *left* plots are adjusted according to (12).

A. Power conservation by adjusting transmission power

In the next section we will show through simulations that the controller (9) drives the agents to their destination in an energy efficient manner. However this energy is related to mobility. As mentioned in the introduction, energy is also consumed in sensing and communication. For sensors using RF or radar based omni directional antennas, the power transmitted for sensing and communication is directly related to the square of the radius of sensors, footprint.

$$P_T \propto \Delta^2.$$

We have shown that to solve the rendezvous problem, the controller (9) only needs to maintain edges $(v_i \sim v_j)$ such that $v_i \in D$. So an additional controller can be applied for the transmission power level which will ensure that

1) For agents $v_i \in D$, all the edges are maintained.
2) For agents $v_i \in V \backslash D$, the edges with the neighbors v_j such that $v_j \in D$ are maintained.

It is to be noted here that this controller for transmission power level will operate while maintaining connectivity, thus keeping all the previous analysis valid. The graph $\mathcal{G}(V, E')$, where E' is the set of critical edges, will still be connected with a slight modification that Δ_i will be the footprint radius for agent v_i. Now the requirements (1 and 2) can be accomplished using a simple linear feedback controller.

$$\dot{P}_{T_i}(t) = -u(x, \Delta_i) P_{T_i}(t) \quad \text{for all } v_i \in V, \tag{12}$$

where Δ_i is the radius of the footprint of agent v_i and

$$u(x, \Delta_i) = \begin{cases} \Delta_i - (\max\limits_{v_j \in \mathcal{N}(v_i)} \|x_i - x_j\| + \epsilon) & v_i \in D, \\ \\ \Delta_i - (\max\limits_{v_j \in D \cap \mathcal{N}(v_i)} \|x_i - x_j\| + \epsilon) & v_i \in V \backslash D. \end{cases} \tag{13}$$

Since Δ is directly related with P_T, as P_{T_i} decreases so does Δ_i. One final note is that the controller (12) is one of the many controllers that can be used to accomplish the same task. We selected controller (12) owing to its simplicity.

V. SIMULATIONS

In this section, we establish through Matlab simulations that our proposed scheme is efficient in terms of energy consumption. We will consider energy consumption due to mobility and sensing and communication. For mobility, based on the model presented in [19], energy consumption is related directly to acceleration. This mobility model is intuitive since although moving at a constant velocity consumes energy, it is more costly to frequently accelerate and decelerate. That is why fuel consumption is more in cities than on highways. The total energy consumed is

$$E_{tot} = \alpha E_{acc} + \beta E_{tran} \tag{14}$$

Where α and β are the constants. In this simulation both the constants are taken to be 1. E_{acc} is the energy consumed due to mobility and E_{tran} is the energy consumed due to sensing and communication. In the simulation, we are assuming that energy consumed is directly related to acceleration, i.e., $E_{acc} \propto acceleration$. This assumption is logical but simplistic because higher acceleration will result in higher fuel consumption but this relation is not necessarily linear. However for an initial analysis, this assumption provides a good insight.

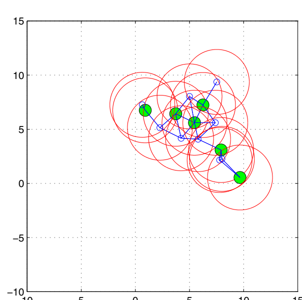
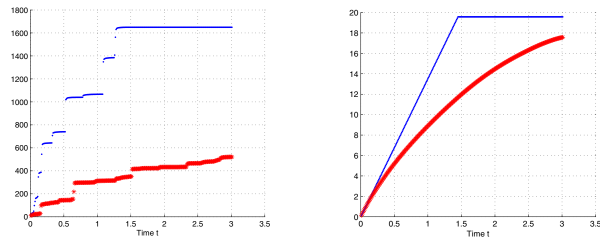


Fig. 3. Initial network topology with $N = 15$ and $\Delta = 3$. The agents represented by big circles constitute WCD. In this example the number of agents in WCD is 6.



(a) (b)

Fig. 5. Comparison of energy consumption between our proposed scheme and the controller (5). The left plot is the energy consumed due to Mobility (acceleration). The right plot is the energy consumed due to sensing and communication. $\alpha = 1$ and $\beta = 1$. In both cases power consumed under our scheme is represented by the bold line. The power consumed by the scheme proposed in (5) is represented by the thin line. In both cases our scheme is consumes less energy.

We start with the system in Figure 3 with $N = 15$ and $\Delta = 3$, and compare the proposed controller (9) with the existing controller (5) as presented in [2]. The reason for this comparison is that, out of the various connectivity maintenance schemes presented in Section I, the one in [2], i.e., controller (5) using edge weights, is computationally efficient and is easy to implement on practical systems. Now, Figure 4(a) and Figure 4(b) show that the controller (5) achieves rendezvous in almost half the time as our proposed scheme. However, this faster convergence comes at a cost of high energy consumption. Figure 5(a) gives a comparison of energy consumption due to mobility in both the cases. It is evident that our scheme outperforms the controller in (5) by a huge gap. It is interesting to analyse the jumps in the plot of controller (5). Since the controller does not allow any edge to break and all the agents are running the weighted consensus, the agents are forced to move in an extremely aggressive manner. The jumps correspond to the formation of new edges which increase the weights and thus the velocity of the corresponding agents. On the other hand the plot for our scheme is smoother as only the nodes in WCD are running weighted consensus while remaining nodes are running standard consensus with weight 1. As a result the formation of new edges does not effect the agents velocity in the drastic manner as is observed in the other plot.

VI. Conclusions

In this paper we presented a framework for distributed motion coordination in multiagent systems with network topology represented by proximity graphs. In the proposed framework, to guarantee the connectivity of the underlying network topology, we identified a subset of edges, which we called critical edges. These were the edges incident on the agents in a weakly connected dominating set. We showed that maintaining only the edges incident to the nodes in WCD is sufficient to enusre connectivity, thereby proposing a controller that preserved those edges. Finally, we demonstrated the efficiency of the proposed scheme in terms of energy consumption through simulations. It is important to note that although the controller designed in this work was for rendezvous problem, the proposed framework is general and it can be applied to address a wide class of distributed motion coordination algorithms for proximity networks in which connectivity of the underlying network is required.

References

[1] J. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom., Leuven, Belgium*, 1998, pp. 2864 - 2869.

[2] M. Ji and M. Egerstedt, "Distributed Coordination Control of Multi-agent Systems While Preserving Connectedness," *IEEE Transactions on Robotics*, vol. 23, issue 4, Aug. 2007, pp. 693 - 703.

[3] H. Jaleel and M. Egerstedt, " Power-Aware Rendezvous with Shrinking Footprints," in *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, San Francisco, CA. Sept 25-30, 2011.

[4] H. Jaleel, S.D. Bopardikar, and M. Egerstedt, "Towards Power-Aware Rendezvous," in *IEEE Conference on Decision and Control*, Orlando, FL, Dec. 2011

[5] M. C. De Gennaro and A. Jadbabaie." Decentralized control of connectivity for multiagent systems," In *Proc. of the IEEE International Conf. on Decision and Control*, page 3628, 2006.

[6] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods for Multiagent Networks*, Princeton University Press, Princeton, NJ, Sept. 2010.

[7] M. M. Zavlanos, H. G. Tanner, A. Jadbabaie and G. J. Pappas. "Hybrid Control for Connectivity Preserving Flocking," *IEEE Transactions on Automatic Control*, Vol. 54, No. 12, pp. 2869-2875, December 2009.

[8] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar. "Decentralized estimation and control of graph connectivity for mobile sensor networks," in *Automatica,* Vol 46, pp. 390396, 2010.

[9] R. O. Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," in *IEEE Trans. Autom. Control*, vol. 49, no. 9, Sep. 2004, pp. 1520 - 1533.

[10] Y. Kim and M. Mesbahi, "On Maximizing the Second Smallest Eigenvalue of a State-Dependent Graph Laplacian," in *IEEE Trans. Autom. Control*, vol. 51, no. 1, Jan. 2006, pp. 116 - 120.

[11] T.W. Haynes, S.T. Hedetniemi and P.J. Slater, *Fundamentals of domination in graphs*, Mercell Dekker, New York, 1998

[12] J.E. Dunbar, J.W. Grossman, J.H. Hattingh, S.T. Hedetniemi, and A.A. McRae, "On weakly connected domination in graphs," *Discrete Mathematics*, vol. 167, pp. 261–269, 1997.

[13] Y.P. Chen and A.L. Liestman,"Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks," In *Proc. of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, Lausanne, Switzerland, 2002.

[14] Y.P. Chen and A.L. Liestman, "A zonal algorithm for clustering ad hoc networks," *International Journal of Foundations of Computer Science*, vol. 14, no. 2, 2003.

[15] K.M. Alzoubi, P.-J. Wan, and O. Frieder, "Maximal independent set, weakly connected dominating set, and induced spanners for mobile ad hoc networks," *International Journal of Foundations of Computer Science*, vol. 14, no. 2, pp. 287–303, 2003.

[16] D. Dubhashi, A. Mei, A. Panconesi, J. Radhakrishnan, and A. Srinivasan, "Fast distributed algorithms for (weakly) connected dominating sets and linear-size skeletons," *Journal of Computer and System Sciences*, vol. 71, pp. 467–479, 2005.

[17] B. Han and W. Jia, "Clustering wireless ad hoc networks with weakly connected dominating set," *Journal of Parallel and Distributed Computing*, vol. 67, no. 6, pp. 727–737, 2007.

[18] J. Yu, N. Wang, G. Wang, and D. Yu, "Connected dominating sets in wireless ad hoc and sensor networks A comprehensive survey," *Computer Communications*, vol. 36, pp.121–134, 2013.

[19] P. Tokekar, N. Karnad, and V. Isler, "Energy-optimal trajectory planning for car-like robots," *Autonomous Robots*, pp. 1-22, 2013