Graph-Theoretic Methods for Multi-Agent Coordination

Magnus Egerestedt*

Abstract

By ignoring the geometric constraints that inevitably govern inter-robot interactions in decentralized robot networks, a purely combinatorial description of the network is obtained. In fact, it can be described as a graph, with vertices corresponding to the individual robots, and edges corresponding to the existence of an inter-robot communication (or sensing) link. In this note, we report on some of the recent results that have emerged in the general area of graph-based multi-agent control. Most notably of these might be the consensus equation that allows us to drive a scalar state value to the same value for the different robots, in a completely decentralized fashion.

1 Introduction: Combinatorics vs. Geometry

The emergence of decentralized, mobile multi-agent networks, such as distributed robots, mobile sensor networks, or mobile ad-hoc communications networks, has imposed new challenges when designing control algorithms. These challenges are due to the fact that the individual agents have limited computational, communications, sensing, and mobility resources. In particular, the information flow between nodes of the network must be taken into account explicitly already at the design phase and a number of approaches have been proposed for addressing this problem, e.g. [6, 7, 8, 9, 10, 11, 17].

Regardless of whether the information flow is generated over communication channels or through sensory inputs, the underlying geometry is playing an important role. For example, if an agent is equipped with omnidirectional range sensors, it can only detect neighboring agents if they are located in a disk around the agent. Similarly, if the sensor is a camera, the area becomes a wedge rather than a disk. But, to make the interaction geometry explicit when designing control laws is not an easy task, and an alternative view is to treat interactions as purely combinatorial. In other words, all that matters is whether or not an interaction exists

*School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA. E-mail:magnus@ece.gatech.edu. This work was supported by the U.S. Army Research Office through the grant # 99838.

between agents, and under certain assumptions on the global interaction topology, one can derive remarkably strong and elegant results. (For a representative sample, see [6, 11, 17].) What then remains to be shown is that the actual geometry in fact satisfies the combinatorial assumptions.

This view, i.e. that the geometry is abstracted down to a purely combinatorial relationship allows the control laws to be designed based without explicit dependency on the geometry of the system. Rather, the control laws will end up depending solely on certain topological properties of the network, such as connectivity or balancing. As such, one has effectively traded away a hard problem (the geometric problems are for example addressed in [1, 9]) for a simple problem under some assumptions that may or may not always be valid (e.g. [7]). And, in this note, we discuss some of the main results in the area of graph-based (or combinatorial) multi-agent control in the domain of multi-agent robotics.

2 Algebraic Graph Theory and Proximity Graphs

2.1 Basic Notation

Assume that the multi-robot system consists of N agents, evolving in a d-dimensional state space, i.e. that $x_i \in \Re^d,\ i = 1, \ldots, N$. Let $V = \{1, \ldots, N\}$ be a set of vertices in a graph G, corresponding to the identity of the robots. Moreover, let the graph $G = (V, E)$, where the edge set $E \subset V \times V$ is a set of unordered pairs of vertices. The interpretation is that an edge (v_i, v_j) is in E if agents i and j can interact with each other.[1]

Graphs are combinatorial objects (sets and pairwise relations between elements in he sets). In order to endow these objects with algebraically manipulable items, such as matrices, one has to look at the area of *algebraic graph theory*. (See for example [5].) For example, some standard matrices associated with G are the *degree matrix* D and the *adjacency matrix* A. The degree matrix is a diagonal matrix $D = \text{diag}(\text{deg}(v_1), \ldots, \text{deg}(v_N))$, where $\text{deg}(v_j)$ is the degree of vertex v_j, i.e. the number of vertices adjacent to v_j. We will let N_j denote the set of adjacent, or neighboring nodes, i.e. $N_j = \{v_i \mid (v_i, v_j) \in E\}$ and hence $|N_j| = \text{deg}(v_j)$. The adjacency matrix $A = [a_{ij}] \in \{0, 1\}^{N \times N}$, where

$$a_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise.} \end{cases}$$

Another matrix of fundamental importance in algebraic graph theory is the *graph Laplacian* $L = D - A$, which has the following key properties:

- $L = L^T \succeq 0$, i.e. it is positive semi-definite.
- If the graph is connected, i.e. there exists a path between any two vertices, then

[1] In this note, we will assume that the edges are undirected, i.e. that $(v_i, v_j) \in E \Leftrightarrow (v_j, v_i) \in E$.

– The (ordered), non-negative, real eigenvalues of L satisfies $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_N$.

– $\text{null}(L) = \text{span}\{\mathbf{1}\}$, i.e. $L\mathbf{1} = 0$, where $\mathbf{1} = (1, \ldots, 1)^T$.

These facts about the graph Laplacian will play an important role in subsequent sections.

2.2 Proximity Graphs

The way geometry enters into the picture is through so-called *proximity graphs* [9]. The idea here is that edges in the graph exist when the underlying geometry satisfies certain properties. For example, if the robots, whose positions are $x_1, \ldots, x_N \in \Re^d$, are all equipped with omni-directional range sensor, with an effective range δ, the induced δ-disk proximity graph is $G(t) = (V, E(t))$. Here the vertex set is $V = \{v_1, \ldots, v_N\}$, and $(v_i, v_j) \in E(t) \Leftrightarrow \|x_i(t) - x_j(t)\| \leq \delta$. An example of such a graph is given in Figure 1.

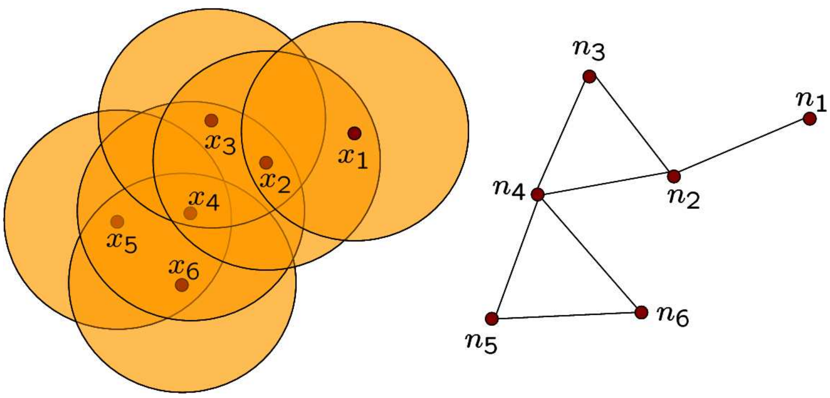

Figure 1: δ-disk proximity graph.

One should note that as the agents move around, neighbors (i.e. adjacent vertices) may be introduced or lost. As such, the graph is no-longer a static structure and this type of graph is referred to as a *dynamic* graph. Other types of recently studied proximity graphs are the Gabriel graphs, Vornoi graphs, and DeLaunay graphs, just to name a few [9, 16].

3 Consensus Problems

3.1 Static Networks

The consensus problem is in essence a problem involving having multiple agents reach an agreement about a scalar state value over a network. This problem is a canonical problem in decentralized coordination and it can be solved quite elegantly using algebraic graph theory.

One instantiation of the consensus problem is the so-called *rendezvous* problem where a collection of agents are supposed to meet at an unspecified common location. In particular, we will assume that each agent is located at x_i and that it can only measure the relative position of its neighboring agents, i.e. it can measure $x_i - x_j,\ \forall j \in N_i$. Moreover, assuming that each agent has single-integrator dynamics, i.e. $\dot{x}_i = u_i$ one reasonable control strategy is to drive each agent towards the centroid of its neighboring set as

$$\dot{x}_i = -\sum_{j \in N_i} (x_i - x_j),$$

where N_i is the set of robots adjacent to robot i. In fact, in the proceeding paragraphs, we will assume that the underlying graph is static. Based on this assumption, the above equation can be rewritten as

$$\dot{x}_i = -\deg(v_i) x_i + \sum_{i=1}^{N} a_{ij} x_j,$$

where, as before, $\deg(v_i)$ is the degree of node i, and a_{ij} is the (i,j):th entry in the adjacency matrix.

Now, if we for the sake of argument, assume that $x_i \in \Re$, i.e. each robot evolves in a one-dimensional space, then by recalling that $L = D - A$, the above equation can be rewritten as

$$\dot{x} = -Lx,$$

where $x = (x_i, \ldots, x_N)^T$. Note that this is a standard, linear, time-invariant system whose stability properties are entirely given by the eigenvalues to $-L$. But, we already know that as long as G is connected, then L is positive semidefinite, and, as such, $-L$ is negative semidefinite. In fact, we know that $-L$ has a single 0 eigenvalue and all other eigenvalues are negative and real. As such, we have that the system is stable and that x will tend to the null-space of L asymptotically. In other words, $x_i \to \alpha$ as $t \to \infty,\ \forall i$, where $\alpha \in \Re$. This, by now widely utilized consensus-equation has appeared with a number of variations, e.g. [6, 10, 11, 15].

The interpretation here is that all components of x (i.e. the scalar positions of all the robots) will tend to the same value. And hence the rendezvous problem is solved. In fact, it is easy to establish that the centroid

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

is static. And as a consequence $\alpha = \bar{x}$, i.e. *all agents will tend asymptotically to the static centroid of the robot team.* In fact, in [12], it was shown that the rate of convergence to the centroid is given by

$$\|x(t) - \bar{x}\mathbf{1}\| \leq \|x(0) - \bar{x}\mathbf{1}\| e^{-\lambda_2 t},$$

where λ_2 is the second smallest eigenvalue of the graph Laplacian.

If $x_i \in \Re^d$ one can directly note that it is possible to decouple the dynamics along each dimension, i.e. if we let $\text{comp}(x,j) = (x_{1,j},\ldots,x_{N,j})^T$, where $x_i = (x_{i,1},\ldots,x_{i,N})^T$, then we have

$$\frac{d\text{comp}(x,j)}{dt} = -L\text{comp}(x,j),\ \ j = 1,\ldots,d.$$

As such, the previous argument can be applied in this case as well. An example of running the consensus algorithm is shown in Figure 2, where 10 agents have to reach an agreement (or consensus). Note that even though they all start out with different values, they quickly converge to a common value based solely on local information.

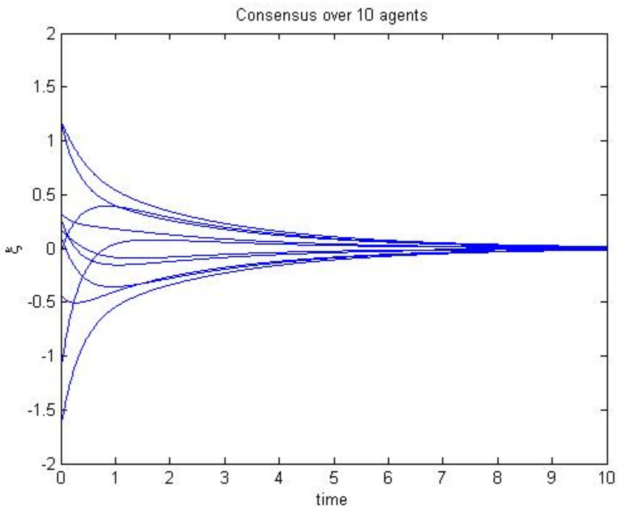

Figure 2: Application of the consensus algorithm.

3.2 Dynamic Networks

Note that so far we have assumed that G is connected and *static*. However, as shown in [6, 7, 8, 17], the above argument still holds as long as the graph stays connected. The connection between combinatorics and geometry is thus made through the assumption that the underlying geometry satisfies this key assumption. However, if we assume that the underlying graph is a disk graph, this assumption may not always hold, as shown below, in Figure 3.

3.3 Linear Formation Control

One reason why the consensus idea is so powerful is that even though the rendezvous problem may be of limited use per se, we can still apply the same thinking to a number of problem

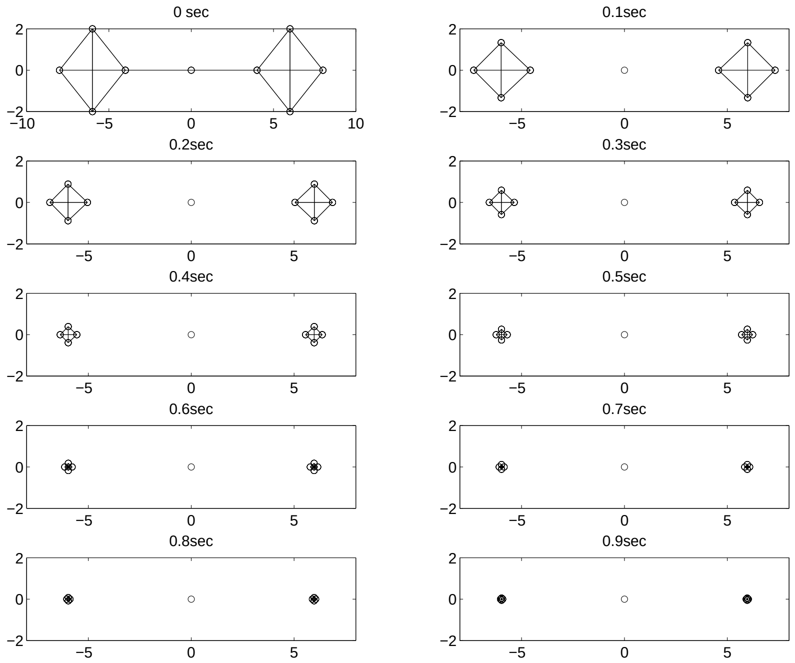

Figure 3: A progression is shown where connectedness is lost even though the initial graph is connected.

formulations, including coverage control (e.g. [1, 2]), containment control [4], distributed filtering [13], and *formation control*, e.g. [3, 7, 15].

In this context, formations are specified in terms of a collection of desired inter-agent distances $d_{ij},\ (i,j) \in D \subset \{1,\dots,N\}^2$. If we assume that these distances are feasible, i.e. there exist points $y_1,\dots,y_N$ such that $\|y_i - y_j\| = d_{ij},\ \forall (i,j) \in D$, we can let the relative errors be given by $\gamma_i = x_i - y_i$.

Now, by observing that if we can drive all relative errors to the same value, we have achieved a pure translation of the points $y_1,\dots,y_N$. As such, one can attempt to solve the formation control problem by simply running a consensus equation over the relative errors [3, 7], as

$$\dot{\gamma}_i = -\sum_{j \in N_i} (\gamma_i - \gamma_j).$$

But, noting that $\dot{\gamma}_i = \dot{x}_i$ and $\gamma_i - \gamma_j = x_i - x_j - (y_i - y_j)$, we get that each agent should move as

$$\dot{x}_i = -\sum_{j \in N_i} (x_i - x_j - \zeta_{ij}),$$

where $\zeta_{i,j} = y_i - y_j$.

One should keep in mind that even though these are elegant results, they all hinge on the fact that connectivity is preserved. And, as shown above, this is not always the case. Recently, a number of papers dealing with the issue of preserving connectedness through nonlinear weights have appeared [3, 7].

4 Controllability and Anchor Networks

Another area where graph-based methods for multi-agent coordination have proved useful are for networks where some agents take on special, so-called leader (or anchor) roles [18].

4.1 Leader-Follower Structures

Assume that a single agent (let's say robot N) is stationary while the others are executing the consensus equation discussed in the previous section, as

$$\begin{aligned} \dot{x}_i &= -\textstyle\sum_{j \in N_i}(x_i - x_j), \; i = 1, \ldots, N-1 \\ \dot{x}_N &= 0, \end{aligned}$$

one can wonder if rendezvous (or consensus) is still achieved. The answer to this question is yes, and, as long as the network stays connected, all agents will converge to the static leader (or anchor) agent, as shown in [18].

This fact is interesting since it essentially allows us to control the network by moving the leader agent around. In fact, as long as it moves slow enough (as compared to the convergence rate of the consensus equation) we can expect the other agents to follow the leader agent rather closely.

Moreover, if we have a number of stationary leader agents, it was shown in [4] that the remaining agents will in fact converge to the convex hull spanned by the leader agents. This observation moreover allows us to exert boundary value control of the network by changing the shape of this convex hull, as was the case in [4], and as is illustrated in Figure 4.

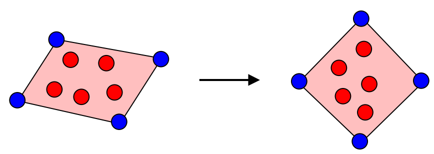

Figure 4: The containment problem: The leaders move in such a way that the followers remain in the convex leader-polytope for all times.

4.2 Graph-Based Controllability

If we assume that the network is static and that the first $N - N_l$ agents are followers, and the last N_l agents are leaders, we can decompose the graph Laplacian as

$$L = \left[\begin{array}{c|c} L_l & \ell \\ \hline \ell^T & L_f \end{array} \right].$$

Here $L_l \in \Re^{(N-N_l)\times(N-N_l)}$, $\ell \in \Re^{(N-N_l)\times N_l}$, and $L_f \in \Re^{N_l \times N_l}$. If we as before, without loss of generality, assume that $x_i \in \Re,\ i = 1, \ldots, N$, we can now let $x = (x_1, \ldots, x_{N-N_l})^T$. Moreover, if we assume that the followers are executing the consensus equation while we can control the position of the leaders directly (or it's velocities - it does not matter from a controllability point-of-view), we can let $u = (x_{N-N_l+1}, \ldots, X_N)^T$. The corresponding control system thus becomes

$$\dot{x} = -L_l x - \ell u.$$

One can thus ask the following question: For what topologies does the above equation correspond to a completely controllably system? In other words, if (L_l, ℓ) was a controllable pair, we could drive the followers to whatever position we would like. And, as it turns out, tools from algebraic graph theory once again help us understanding this issue.

Below, in Figure 5, are given three different graphs. The first two are not controllable and the reason for this is that the followers are somehow symmetric with respect to the leader, i.e. if $x_1(0) = x_2(0)$ then $x_1(t) = x_2(t),\ \forall t \geq 0$. This is not the case in the third case.

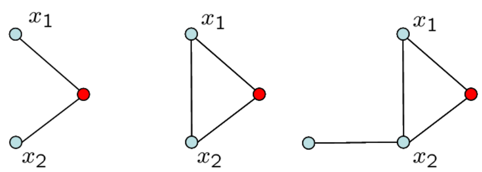

Figure 5: Two uncontrollable graph structures (left and middle) and one controllable (right).

Technially, what happened here is that in the uncontrollable cases, it was possible to relabel the non-leader agents while still maintaining the same edge relations. Technically speaking, such an adjacency preserving vertex permutation is called a *graph automorphism*. In other words, $\psi : V \to V$ is a rgaph automorphism if $(v_i, v_j) \in E \ \Leftrightarrow\ (\psi(v_i), \psi(v_j)) \in E$. And, in [14], it was shown that a sufficient (and in some cases necessary and sufficient) condition for a single leader network to be uncontrollable is that there exists a non-trivial (not the identity) graph automorphism, with $\psi(v_N) = v_N$.

For multiple leaders, things become more complicated, but analogous sufficient conditions for uncontrollability have been found in [14] based on so-called *equitable partitions* of the graph. (Interested readers are referred to [14].)

5 Conclusions

In this note, we report on some of the recent results that have emerged in the general area of graph-based multi-agent control. In fact, by focusing purely on the combinatorial nature of the network (and thus ignoring the geometric constraints on the inter-robot interactions) a number of powerful results can be obtained. Most notably of these might be the consensus equation that allow us to drive a scalar state value to the same value for the different robots, in a completely decentralized fashion. This is possible as long as the network stays connected, which is an assumption that one may or may not always be justified in making.

References

[1] J. Cortes, S. Martinez, and F. Bullo. Robust Rendezvous for Mobile Autonomous Agents via Proximity Graphs in Arbitrary Dimensions. *IEEE Transactions on Automatic Control*, Vol. 51, No. 8, pp. 1289-1298, 2006.

[2] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage Control for Mobile Sensing Networks. *IEEE Transactions on Robotics and Automation*, Vol. 20, No. 2, pp. 243–255, 2004.

[3] D.V. Dimarogonas and K.J. Kyriakopoulos. On the Rendezvous Problem for Multiple Nonholonomic Agents. *IEEE Transactions on Automatic Control*, Vol. 52, No. 5, pp. 916-922, May 2007.

[4] G. Ferrari-Trecate, M. Egerstedt, A. Buffa, and M. Ji. Laplacian Sheep: A Hybrid, Stop-Go Policy for Leader-Based Containment Control. *Hybrid Systems: Computation and Control*, Springer-Verlag, pp. 212-226, Santa Barbara, CA, March 2006.

[5] C. Godsil and G. Royle. *Algebraic graph theory*, Springer, 2001.

[6] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, Vol. 48, No. 6, pp. 988-1001, June 2003.

[7] M. Ji and M. Egerstedt. Distributed Coordination Control of Multi-Agent Systems While Preserving Connectedness. *IEEE Transactions on Robotics*, Vol. 23, No. 4, pp. 693-703, Aug. 2007.

[8] Z. Lin, B. Francis, and M. Maggiore. Necessary and Sufficient Graphical Conditions for Formation Control of Unicycles. *IEEE TRansactions on Automatic Control*, pp. 121-127, 2005.

[9] S. Martinez, J. Cortes, and F. Bullo. Motion Coordination with Distributed Information. *IEEE Control Systems Magazine*, Vol. 27, No. 4, pp. 75-88, 2007.

[10] M. Mesbahi. On state-dependent dynamic graphs and their controllability properties. *IEEE Transactions on Automatic Control*, Vol. 50, No. 3, pp. 387- 392, 2005.

[11] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and Cooperation in Networked Multi-Agent Systems. *Proceedings of the IEEE*, Vol. 95, No. 1, pp. 215-233, Jan. 2007.

[12] R. Olfati-Saber. Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory. *IEEE Trans. on Automatic Control*, Vol. 51, No. 3, pp. 401-420, Mar. 2006.

[13] R. Olfati-Saber. Distributed Kalman Filtering for Sensor Networks. *IEEE Conference on Decision and Control*, Dec. 2007.

[14] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt. Controllability of Multi-Agent Systems from a Graph-Theoretic Perspective. *SIAM Journal on Control and Optimization*. To appear.

[15] W. Ren and R.W. Beard. *Distributed Consensus in Multi-Vehicle Cooperative Control*, Communication and Control Engineering Series, Springer Verlag, New York, 2007.

[16] B. Shucker, T.D. Murphey, and J. Bennett. Switching rules for decentralized control with simple control laws. *American Control Conference*, 2007.

[17] H.G. Tanner, A. Jadbabaie, and G.J. Pappas. Flocking in Fixed and Switching Networks. *Transactions on Automatic Control*, Vol. 52, No. 5, pp. 863-868, 2007.

[18] H.G. Tanner. On the Controllability of Nearest Neighbor Interconnections. *IEEE Conference on Decision and Control*, Dec. 2004.