Applications of Connectivity Graph Processes in Networked Sensing and Control*

Abubakr Muhammad, Meng Ji, and Magnus Egerstedt

School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA
{abubakr,mengji,magnus}@ece.gatech.edu

Summary. This paper concerns the problem of controlling mobile nodes in a network in such a way that the resulting graph-encoding of the inter-node information flow exhibits certain desirable properties. As a sequence of such transitions between different graphs occurs, the result is a graph process. In this paper, we not only characterize the reachability properties of these graph processes, but also put them to use in a number of applications, ranging from multi-agent formation control, to optimal collaborative beamforming in sensor networks.

1 Introduction

As the complexity associated with the control design tasks for many modern engineering systems increases, strategies for managing the complexity have become vitally important. This is particularly true in the areas of networked and embedded control design, where the scale of the system renders classical design tools virtually impossible to employ. However, the problem of coordinating multiple mobile agents is one in which a finite representation of the configuration space appears naturally, namely by using graph-theoretic models for describing the local interactions in the formation. In other words, graph-based models can serve as a bridge between the continuous and the discrete when trying to manage the design-complexity associated with formation control problems. Notable results along these lines have been presented in [1, 2, 3, 4, 5].

The conclusion to be drawn from these research efforts is that a number of questions can be answered in a natural way by abstracting away the continuous dynamics of the individual agents. Several terms such as link graphs, communication graphs and connectivity graphs have been used interchangeably in the literature for graphical models that capture the local limitations

* This work was supported by the U.S. Army Research Office through the grant # 99838.

of sensing and communication in decentralized networked systems. In this paper, we give several applications of connectivity graphs and their dynamics: the so-called connectivity graph processes.

The outline of this paper as follows. In Section 2 we provide various characterizations and computational tools that deal with the realization of connectivity graphs in their corresponding configuration spaces, based on our work in [11, 12]. These studies let us distinguish between valid and invalid graphical configurations and provide a computable way of determining valid transitions between various configurations, which is the topic of Section 3. Moreover, notions such as graph reachability and planning will be given a solid foundation. We moreover develop an optimal control framework, where the configuration space is taken as the space of all connectivity graphs. Finally, Section 4 is devoted to the various applications of this framework. In particular, we study low-complexity formation planning for teams for mobile robots and collaborative beamforming in mobile sensor networks.

2 Formations, Connectivity Graphs and Feasibility

Graphs can model local interactions between agents, when individual agents are constrained by limited knowledge of other agents. In this section we summarize some previous results, found in [6], of a graph theoretic nature for describing formations in which the primary limitation of perception for each agent is the limited range of its sensor. Suppose we have N such agents with identical dynamics evolving on $\mathbb{R}^2$. Each agent is equipped with a range limited sensor by which it can sense the relative displacement of other agents. All agents are assumed to have identical sensor ranges δ. Let the position of each agent be $\mathbf{x}_n \in \mathbb{R}^2$, and its dynamics be given by $\dot{\mathbf{x}}_n = f(\mathbf{x}_n, u_n)$, where $u_n \in \mathbb{R}^m$ is the control for agent n and $f : \mathbb{R}^2 \times \mathbb{R}^m \to \mathbb{R}^2$ is a smooth vector field. The configuration space $\mathcal{C}^N(\mathbb{R}^2)$ of the agent formation is made up of all ordered N-tuples in $\mathbb{R}^2$, with the property that no two points coincide, i.e. $\mathcal{C}^N(\mathbb{R}^2) = (\mathbb{R}^2 \times \mathbb{R}^2 \times \ldots \mathbb{R}^2) - \Delta$, where $\Delta = \{(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) : \mathbf{x}_i = \mathbf{x}_j \text{ for some } i \neq j\}$. The evolution of the formation can be represented as a trajectory $\mathcal{F} : \mathbb{R}_+ \to \mathcal{C}^N(\mathbb{R}^2)$, usually written as $\mathcal{F}(t)(\mathbf{x}_1(t), \mathbf{x}_2(t), \ldots \mathbf{x}_N(t))$ to signify time evolution. The spatial relationship between agents can be represented as a graph in which the vertices of the graph represent the agents, and existence of a pair of vertices on each edge tells us that the corresponding agents are within sensor range δ of each other.

Let $\mathcal{G}_N$ denote the space of all possible graphs that can be formed on N vertices $V = \{v_1, v_2, \ldots, v_N\}$. Then we can define a function $\Phi_N : C^N(\mathbb{R}^2) \to \mathcal{G}_N$, with $\Phi_N(\mathcal{F}(t)) = \mathcal{G}(t)$, where $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t)) \in \mathcal{G}_N$ is the *connectivity graph* of the formation $\mathcal{F}(t)$. $v_i \in \mathcal{V}$ represents agent i at position $\mathbf{x}_i$, and $\mathcal{E}(t)$ denotes the edges of the graph. $e_{ij}(t) = e_{ji}(t) \in \mathcal{E}(t)$ if and only if $\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| \leq \delta$, $i \neq j$. These graphs. The graphs are always undirected because the sensor ranges are identical. The motion of agents in a formation may result in the removal or addition of edges in the graph. Therefore $\mathcal{G}(t)$ is a dynamic structure. Lastly and most importantly, every graph in $\mathcal{G}_N$ is not a connectivity graph. The last observation is not as obvious as the others, and it has been analyzed in detail in [6]. A *realization* of a graph $\mathcal{G} \in \mathcal{G}_N$ is a formation $\mathcal{F} \in \mathcal{C}^N(\mathbb{R}^2)$, such that $\Phi_N(\mathcal{F}) = \mathcal{G}$. An arbitrary graph $\mathcal{G} \in \mathcal{G}_N$ can therefore be *realized* as a connectivity graph in $\mathcal{C}^N(\mathbb{R}^2)$ if $\Phi_N^{-1}(\mathcal{G})$ is nonempty. We denote by $\mathcal{G}_{N,\delta} \subseteq \mathcal{G}_N$, the space of all possible graphs on N agents with sensor range δ, that can be realized in $\mathcal{C}^N(\mathbb{R}^2)$. In [7] we proved the following result.

Theorem 1. *$\mathcal{G}_{N,\delta}$ is a proper subspace of $\mathcal{G}_N$ if and only if $N \geq 5$.*

Formations can produce a wide variety of graphs for N vertices. This includes graphs that have disconnected subgraphs or totally disconnected graphs with no edges. However the problem of switching between different formations or of finding interesting structures within a formation of sensor range limited agents can only be tackled if no sub-formation of agents is totally isolated from the rest of the formation. This means that the connectivity graph $\mathcal{G}(t)$ of the formation $\mathcal{F}(t)$ should always remain *connected* (in the sense of connected graphs) for all time t.

In [11, 12] we gave a detailed study of feasibility results using semi-definite programming methods and its relation to the Positivstellensatz for semialgebraic sets. In particular, we showed how to setup the feasibility of geometrical constraints in a possible graph as a linear matrix inequality (LMI) problem as follows. Recall that the connectivity graph $(\mathcal{V}, \mathcal{E})$ corresponding to the formation $(\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_N) \in \mathcal{C}^N(\mathbb{R}^2)$ can be described by $N(N-1)/2$ relations of the form

$$\begin{aligned}\delta^2 - (x_i - x_j)^2 - (y_i - y_j)^2 \geq 0,\ \text{if } e_{ij} \in \mathcal{E},\\ (x_i - x_j)^2 + (y_i - y_j)^2 - \delta^2 > 0,\ \text{if } e_{ij} \notin \mathcal{E},\end{aligned}$$

where $1 \leq i < j \leq N$ and $\mathbf{x}_i = (x_i, y_i)$. Therefore the realization problem is equivalent to asking if there exist $x_1, y_1, \ldots, x_N, y_N$ such that these inequality constraints are satisfied. In [11] we showed that the non-feasibility problem is equivalent to asking if the set $X = \{\mathbf{x} \in \mathbb{R}^M \mid \mathbf{x}^T A_{ij} \mathbf{x} \geq 0,\ 1 \leq i < j \leq N,\ e_{ij} \in \mathcal{E}, \mathbf{x}^T B_{lm} \mathbf{x} > 0,\ 1 \leq l < m \leq N,\ e_{lm} \notin \mathcal{E}\}$ is empty for certain A_{ij} and B_{lm} matrices (See [11] for details). Since all semi-algebraic constraints on the set X are quadratic and Moreover, it was also shown that $A_{ij} = A_{ij}^T, B_{lm} = B_{lm}^T$, we can use the celebrated $\mathcal{S}$-procedure to transform the feasibility question into a linear matrix inequality (LMI) problem [8].

Theorem 2. *Given symmetric $n \times n$ matrices $\{A_k\}_{k=0}^m$, the following are equivalent:*

1. *The set $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}^T A_1 \mathbf{x} \geq 0, \mathbf{x}^T A_2 \mathbf{x} \geq 0, \cdots, \mathbf{x}^T A_m \mathbf{x} \geq 0, \mathbf{x}^T A_0 \mathbf{x} \geq 0, \mathbf{x}^T A_0 \mathbf{x} \neq 0\}$ is empty.*
2. *There exist non-negative scalars $\{\lambda_k\}_{k=1}^m$ such that $-A_0 - \sum_{k=1}^m \lambda_k A_k \geq 0$.*

If a solution to this LMI exists then we say that we have a certificate of infeasibility. It was demonstrated how to use a standard LMI-software [9] to effectively solve a wide class of such feasibility problems. Once such infeasibility certificates have been obtained, they can be used in formation planning under constraints of communication and sensory perception.

3 Connectivity Graph Processes For Formation Planning

As mentioned in Section 1, formation switching with limited global information is an important problem in multi-agent robotics. However, little work has been done so far that adequately addresses the problem of formation switching under limited range constraints. Therefore the ability to give exact certificates about what can and cannot be achieved under these constraints is a desirable result. Recall that the connectivity graph of the formation evolves over time as $G(t) = (\mathcal{V}, \mathcal{E}(t)) = \Phi_N(\mathbf{x}(t))$. Under standard assumptions on the individual trajectories of agents, one gets a finite sequence of graphs $\{G_0, G_1, \ldots, G_M\}$ for each finite interval of evolution $[0, T]$, where $\Phi_N(\mathbf{x}(0)) = G_0$, G_i switches to G_{i+1} at time t_i, and $\Phi_N(\mathbf{x}(T)) = G_M$. We will often write this as $G_0 \to G_1 \to \ldots \to G_M$ and call such a sequence a *connectivity graph process*[2]. Such graph processes can be thought of as trajectories on the space $\mathcal{G}_{N,\delta}$. In what follows we discuss the role of *feasible, reachable* and *desirable* sets when generating trajectories on this space.

3.1 Feasible connectivity graph transitions

The connectivity graph processes are generated through through the movement of individual nodes. For a connectivity graph $G_j = (\mathcal{V}_j, \mathcal{E}_j) = \Phi_N(\mathbf{x}(t_j))$ let the nodes be partitioned as $\mathcal{V}_j = \mathcal{V}_j^0 \cup \mathcal{V}_j^m$, where the movement of the nodes in V_j^m facilitates the transition from G_j to the next graph G_{j+1} and V_j^0 is the set of nodes that are stationary. With the positions $\mathbf{x}_j^0 = \{\mathbf{x}_m(t_i)\}_{m \in \mathcal{V}_j^0}$ being fixed, let $\text{Feas}(G_j, \mathcal{V}_j^m, \mathbf{x}_j^0) \subseteq \mathcal{G}_{N,\delta}$ be the set of all connected connectivity graphs that are feasible by an unconstrained placement of positions corresponding to $\mathcal{V}_j^m$ in $\mathbb{R}^2$. (We will often denote this set as $\text{Feas}(G_j, \mathcal{V}_j^m)$, when the the positions $\mathbf{x}_j^0$ are understood from context.) The set $\text{Feas}(G_j, \mathcal{V}_j^m)$ of feasible graph transitions can be computed using the semi-definite programming methods discussed above. It will be appropriate to explain the reason for keeping track of mobile and stationary nodes at each transition. In principle, it is possible to compute However, in order to manage the combinatorial growth in the number of possible graphs, it is desirable to let the transitions be generated by the movements of a small subset of nodes only. In fact, we will investigate the situation where er only move one node at a time.

[2] We borrow this term from Mesbahi [2].

Let $\mathcal{V}_j^0 = \{1, \ldots, k-1, k+1, \ldots, N\}$ and $\mathcal{V}_j^m = \{k\}$. It should be noted that the movement of node k can only result in the addition or deletion of edges that have node k as one of its vertices. Therefore the enumeration of the possible resulting graphs should count all possible combinations of such deletions and additions. This number can be easily seen to be 2^{N-1} for N nodes. Since we are also required to keep the graph connected at all times, this number is actually $2^{N-1}-1$, obtained after removing the graph in which node k has no edge with any other node.

Now, we can use the S-procedure to evaluate whether each of the new graphs resulting from this enumeration is feasible. Since all nodes are fixed except for $\mathbf{x}_k = (x, y)$, the semi-algebraic set we need to check for non-feasibility is defined by $N-1$ polynomial inequalities over $\mathbb{R}[x, y]$. Each of these inequalities has either of the following two forms,

$$\begin{aligned} &\delta^2 - (x - x_i)^2 - (y - y_i)^2 \geq 0, \text{ if } e_{ki} \in \mathcal{E}, \\ &(x - x_i)^2 + (y - y_i)^2 - \delta^2 > 0, \text{ if } e_{ki} \notin \mathcal{E}. \end{aligned}$$

where $2 \leq i \leq N$, $\mathcal{E}$ is the edge set of the new graph and we denote $\mathbf{x}_i(t_j)$ by (x_i, y_i) for $i \neq k$. This computation can be repeated for all N nodes so that we have a choice of $N(2^{N-1}-1)$ graphs. Each of the $N-1$ inequalities can be written as either

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & x_i \\ 0 & -1 & y_i \\ x_i & y_i & \delta^2 - x_i^2 - y_i^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \geq 0, \text{ if } e_{ik} \in \mathcal{E},$$

or

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_i \\ 0 & 1 & -y_i \\ -x_i & -y_i & x_i^2 + y_i^2 - \delta^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} > 0, \text{ if } e_{ik} \notin \mathcal{E}.$$

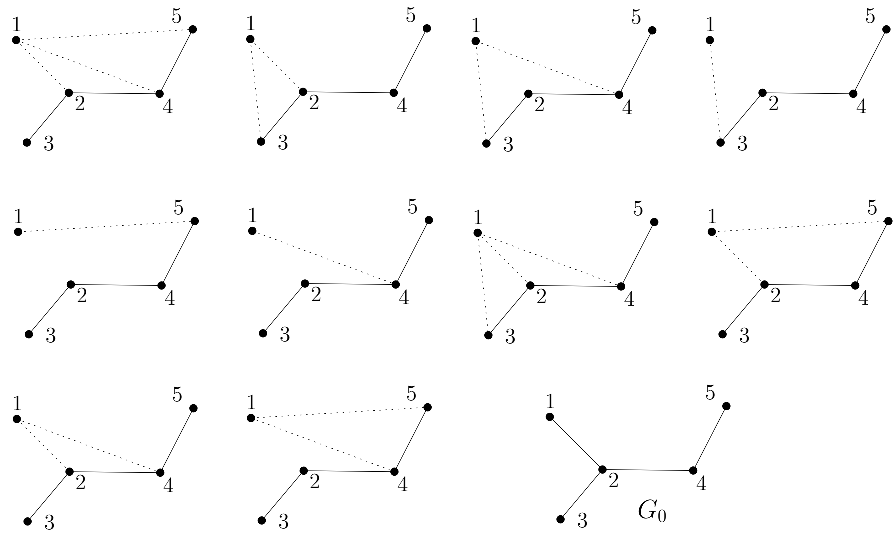
Denoting by

$$A_i = \begin{bmatrix} -1 & 0 & x_i \\ 0 & -1 & y_i \\ x_i & y_i & \delta^2 - x_i^2 - y_i^2 \end{bmatrix}, \qquad B_i = \begin{bmatrix} 1 & 0 & -x_i \\ 0 & 1 & -y_i \\ -x_i & -y_i & x_i^2 + y_i^2 - \delta^2 \end{bmatrix},$$

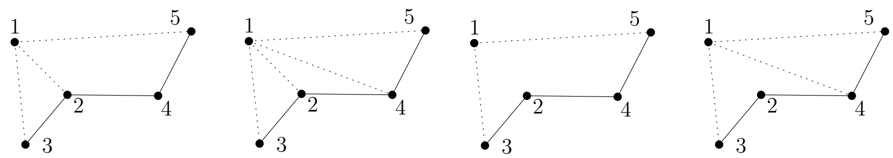
and ignoring the lossy aspect of the S-procedure [8], we need to solve the LMI,

$$-A_{\alpha_1} - \sum_{i \neq 1, e_{\alpha_i k} \in \mathcal{E}} \lambda_{\alpha_i} A_{\alpha_i} - \sum_{j, e_{\alpha_j k} \notin \mathcal{E}} \lambda_{\alpha_j} B_{\alpha_j} \geq 0.$$

An example of such a calculation is given in Figure 1, where $\mathcal{V}^0 = \{2, 3, 4, 5\}$ and $\mathcal{V}^m = \{1\}$. The LMI control toolbox [9] for MATLAB has been used to solve the LMI for each of these graphs in order to get the appropriate certificates.



Set of feasible transitions $Feas(G_0, \{1\})$.



Infeasible transitions.

Fig. 1. Feasible and infeasible graphs by movement of node 1.

3.2 Reachability and Connectivity Graph Processes

Note that $\text{Feas}(G_0, \mathcal{V}_0^m)$ does not depend on the actual movement of the individual nodes. In fact even if $G \in \text{Feas}(G_0, \mathcal{V}_0^m)$, it does not necessarily mean that there exists a trajectory by which $G_0 \to G$ or even that $G_0 \to G_f \ldots \mathcal{G}$. We therefore need some notion of *reachability* on the space $\mathcal{G}_{N,\delta}$. We say that a connectivity graph G_f is *reachable* from an initial graph G_0 if there exists a connectivity graph process of finite length $G_0 \to G_1 \to \ldots G_f$ and a sequence of vertex-sets $\{\mathcal{V}_k^m\}$ such that each $G_{k+1} \in \text{Feas}(\mathcal{G}_k, \mathcal{V}_k^m)$. If $\mathcal{V}_k^m = \mathcal{V}^m$ at each transition, then every $G_k \in \text{Feas}(G_0, \mathcal{V}^m)$. (In particular, $G_f \in \text{Feas}(G_0, \mathcal{V}^m)$.) Consider all such G that are reachable from G_0 with a fixed $\mathcal{V}_m$. We will denote this set by $\text{Reach}(G_0, \mathcal{V}^m)$. It is easy to see that $\text{Reach}(G_0, \mathcal{V}^m) \subseteq \text{Feas}(G_0, \mathcal{V}^m)$.

In the previous paragraphs, it was shown how to determine the membership for the set $\text{Feas}(G_0, \mathcal{V}^m)$. But, determining the membership for $\text{Reach}(G_0, \mathcal{V}^m)$ is not trivial. Under the assumption that individual nodes are

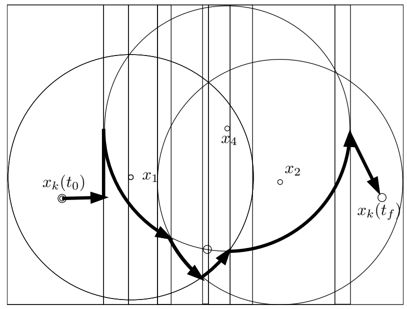


Fig. 2. Selection of Nash cells and the generation of trajectory in a graph process.

globally controllable, a computational tool from algebraic geometry known as *Cylindrical Algebraic Decomposition* [10] or CAD can be used for this purpose, as shown in [12, 13]. For the semialgebraic set S defined by the union of sensing discs of the stationary nodes $\mathcal{V}^0$, CAD provides a decomposition of S into so-called Nash cells. It has been proved in [13] that for two given connectivity graphs $\Phi_N(\mathbf{x}(0))$ and G_f, there exists a finite connectivity graph process $\Phi_N(\mathbf{x}(0)) \to G_1 \to \ldots \to G_f$, and a corresponding trajectory $\mathbf{x}(t) \subseteq C^N(\mathbb{R}^2), t \in [t_0, t_f]$ such that $\mathbf{x}(t_f) \in \Phi_N^{-1}(G_f)$, if and only if there exists a finite collection of Nash cells in the CAD of S such that $x(t)$ belongs to a cell in this collection for all $t \in [t_0, t_f]$. The construction used in proving this result result gives us the trajectories for the actual movement of the nodes. We omit the details of this proof for the sake of brevity. An example of CAD and the trajectory of the mobile node is depicted in Figure 2.

3.3 Global Objectives, Desirable Transitions and Optimality

The whole purpose of a coordinated control strategy in a multi-agent system is to evolve towards the fulfilment of a global objective. This typically requires the minimization (or maximization) of a cost associated with each global configuration. Viewed in this way, a planning strategy should basically be a search process over the configuration space, evolving towards this optimum. If the global objective is fundamentally a function of the graphical abstraction of the formation, then it is better to perform this search over the space of graphs instead of the full configuration space of the system. By introducing various graphical abstractions in the context of connectivity graphs, we have the right machinery to perform this kind of planning. In other words, we will associate a cost or score with each connectivity graph and then work towards minimizing it.

Given $\text{Reach}(G_0, \mathcal{V}^m)$, a decision need to be taken regarding what $G_f \in \text{Reach}(G_0, \mathcal{V}^m)$ the system should switch to. For this we define a cost function $\Psi : G_{N,\delta} \to \mathbb{R}$ and we choose the transition through

$$G_f = \underset{G \in \mathrm{Reach}(G_0, \mathcal{V}^m)}{\arg\min} \Psi(G)$$

Here Ψ is analogous to a terminal cost in optimal control. If, in addition, we also take into account the cost associated with every transition in the graph process $G_0 \to G_1 \to \ldots \to G_M = G_f$ that takes us to G_f, then we would instead consider the minimization of the cost

$$J = \Psi(G_f) + \sum_{i=0}^{M-1} \beta(i) L(G_i, G_{i+1}),$$

where $L : \mathcal{G}_{N,\delta} \times \mathcal{G}_{N,\delta} \to \mathbb{R}$ is the analogue of a discrete Lagrangian, $\beta(i)$ are weighting constants, and $G_{i+1} \in \mathrm{Reach}(G_i, \mathcal{V}^m)$ at each step i. The Lagrangian lets us control the transient behavior of the system during the evolution of the graph process. As an example, let $G_i = (\mathcal{V}_i, \mathcal{E}_i)$ and define

$$L(G_i, G_{i+1}) = |(\mathcal{E}_{i+1} \setminus \mathcal{E}_i) \cup (\mathcal{E}_i \setminus \mathcal{E}_{i+1})|,$$

where $|.|$ gives the cardinality of a set. This Lagrangian is the symmetric difference of the sets of edges in the two graphs. Here, we penalize the addition or deletion of edges at each transition. The resulting connectivity graph process takes G_0 to G_f with minimal structural changes. As another example, if we let $L(G_i, G_{i+1}) = |\mathcal{E}_i| - |\mathcal{E}_{i+1}|$, then the desired G_f is generated while maximizing connectivity during the graph process.

4 Applications of Connectivity Graph Processes

We now give some concrete applications of the connectivity graph processes.

4.1 Production of Low-Complexity Formations

In [14], we have presented a complexity measure for studying the structural complexity of robot formations. The structural complexity is based on the number of local interactions in the system due to perception and communication. When designing control strategies for distributed, multi-agent systems, it is vitally important that the number of prescribed local interactions is managed in a scalable manner. In other words, it should be possible to add new robots to the system without causing a significant increase in the communication and computational burdens of the individual robots. On the other hand, an additional requirement when designing multi-agent coordination strategies should be that enough local interactions are present in order to ensure the proper execution of the task at hand. It turns out that the notion of structural complexity is the right measure to compare these conflicting requirements for the system. Therefore, it would be desirable to obtain graph processes that transform a formation with a high structural complexity to one with a lower complexity (and vice versa.) In [14] we defined the structural complexity of a connectivity graphs G as

$$C(G) = \sum_{v_i \in \mathcal{V}} \left(\deg(v_i) + \sum_{v_j \in V, v_i \neq v_j} \frac{\deg(v_j)}{d(v_i, v_j)} \right),$$

where $d : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^+$ is some distance function defined between vertices. It was also observed that if G is a connected connectivity graph then the complexity of G is bounded above and below by

$$C(\delta_N) \leq C(G) \leq C(\mathcal{K}_N),$$

where δ_N is the δ-chain on N vertices, and $\mathcal{K}_N$ is the complete graph. The lowest complexity graphs or δ-chains (which are the line graphs or the Hamiltonian paths on all vertices), are important for formations that require minimal coordination. Therefore coming up with a planning mechanism to produce low-complexity formations from an arbitrary initial formation is a useful result in multi-agent coordination. Using the concepts from previous sections we define $\Psi(G) = C(G)$ and $L(G_i, G_{i+1}) = |(\mathcal{E}_{i+1} \setminus \mathcal{E}_i) \cup (\mathcal{E}_i \setminus \mathcal{E}_{i+1})|$, where $G_i = (\mathcal{V}_i, \mathcal{E}_i)$. In this way we tend to produce formations that are lower in complexity by generating a graph process that adds or deletes a small number of edges at each transition. The results of one such simulation has been shown in Figure 3. Here, the star-like graph in the upper left corner is the initial connectivity graph (with a higher structural complexity than a δ-chain on 5 vertices). The graph process evolves from left to right and then continues onto the lower rows in the same manner until it reaches the δ-chain in the lower right corner. The process involves transitions to various intermediate graphs. In this example, the mobile node is labelled as 1. As predicted by the CAD decomposition, it first slides up to make an edge with node 2 and then rotates about node 2. It then passes by node 5 making various intermediate graphs, till it comes in the vicinity of node 4. Finally it makes a rotation about node 5 to form the δ-chain. Also note that due to the choice of the Lagrangian described above, the mobile node makes (or breaks) only a minimum number of edges at each graph transition. In this example, it can be seen that this number is always 1.

Simulations for a relatively large number of nodes and more complex formations have also been done. It should be noted that the optimal trajectories thus obtained are only locally minimizing. A scheme for globally optimal behavior is currently under study. We are also working towards extending the number of mobile nodes by a decomposition of CAD into non-overlapping regions for each mobile node.

4.2 Collaborative Beamforming in Sensor Networks

Another promising application of the framework presented in Section 3 is collaborative beamforming. In ad-hoc and wireless sensor networks, long range

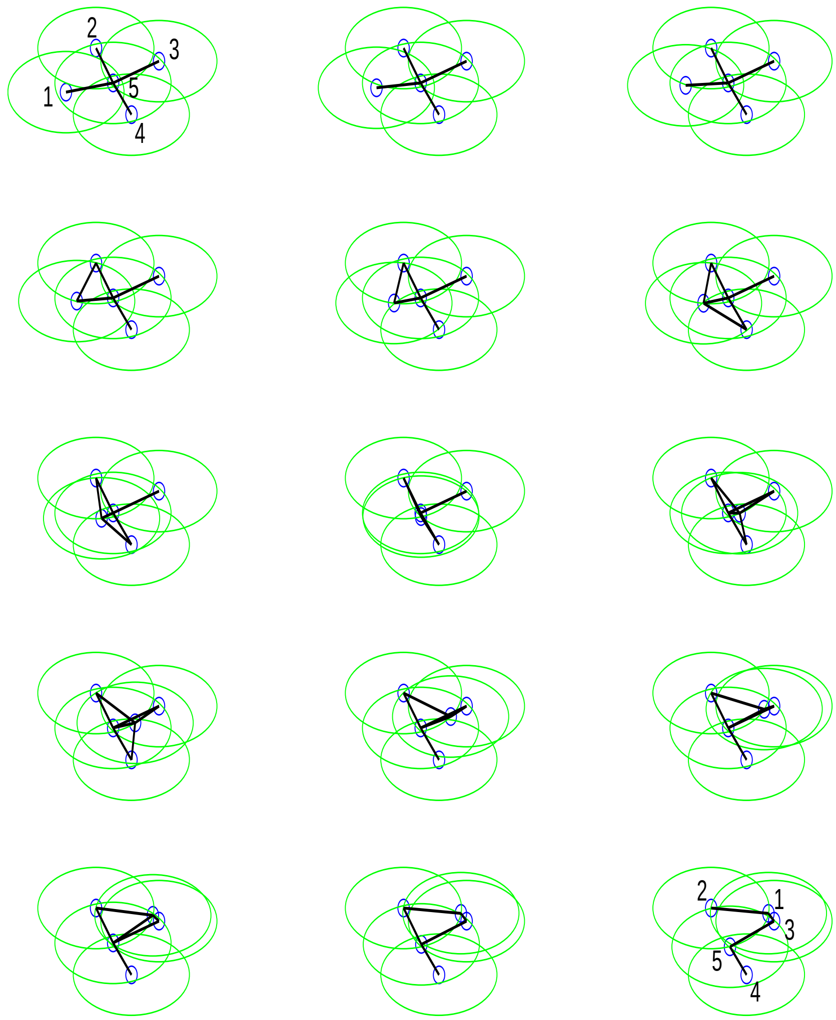


Fig. 3. A connectivity graph process that generates a δ-chain.

communication between clusters is always expensive due to limitations on power and communication channels. Collaborative beamforming is one way to solve this problem without dramatically increasing the complexity compared to non-collaborative strategies [15, 16].

With this method, a cluster of sensors synchronize their phases and collaboratively transmit or receive data in a distributed manner. By properly designing the array factor, one can shape and steer the beam pattern in such a way that the array has either a high power concentration in the desired direction with little leakage (when transmitting), or a high gain in the direction of arrival (DOA) of the signal of interest with significant attenuation in the direction of interference (when receiving). These properties enables Space-Division Multiplex Access (SDMA) among clusters. In most beamforming applications, the array geometry is assumed to be fixed and the optimal beam pattern is formed by optimally weighing the signals received at individual nodes [17]. In this work, we further optimize the beam pattern by altering the geometry of the sensor array using connectivity graph processes.

It should be mentioned that finding an optimal geometry is a difficult design problem in array signal processing. Most designs favor a regular equispaced geometry such as linear, circular, spherical and rectangular grid arrays over random geometries[17]. If one follows the design philosophy in the array processing community, one would tend to drive all nodes to a regular geometry for obtaining better beam patterns. However, more exotic geometries have also been designed for particular applications. Moreover, the placement of nodes in a sensor network is not merely to optimize network communication, but also to maximize some benefit associated with distributed sensing. Therefore, it seems beneficial to optimize the geometries over some cumulative function of both the communication performance as well as the sensing performance of the network. We present this approach below.

Let us first study the beamforming for an arbitrary geometry. Following the standard notation in the array signal processing literature, we describe the positions of the individual nodes, signal and interference in polar coordinate system. The signal is located at a distance A and azimuthal angle ϕ_0 while the interference is at an angle ϕ_i as shown in Fig. 4. The position $\mathbf{x}_k = (x_k, y_k)$ of the k-th node is given by (r_k, θ_k), where $r_k = \sqrt{x_k^2 + y_k^2}$ and $\theta_k = \tan^{-1}(y_k/x_k)$. Given the position of the sensor array $r = [r_1, r_2, \ldots, r_N]$, $\theta = [\theta_1, \theta_2, \ldots, \theta_N]$, we adopt the beamforming algorithm presented in [15], where the gain in direction ϕ is given by the norm of the array factor θ

$$F(\phi|r,\theta) = e^{j\frac{2\pi}{\lambda}A}\frac{1}{N}\sum_{k=1}^{N} e^{j\frac{2\pi}{\lambda}r_k[\cos(\phi_s-\theta_k)-\cos((\phi-\theta_k))]}.$$

For known signal and interference directions, the objective of beamforming is to obtain high signal to interference ratio (SIR) and fine resolution, i.e. we would like to keep the main lobe of the beam pattern as thin as possible while minimizing the power in the interference direction. Let $\Delta\phi_H$ be the half power beam width (HPBW) of the main lobe as depicted in Figure 5. The power concentrations (accumulated gain) in the direction of signal and interference are respectively given by

$$P_s(r,\theta) = \int_{\phi_0-\Delta\phi_H/2}^{\phi_0+\Delta\phi_H/2} |F(\phi|r,\theta)|^2 d\phi, \quad P_i(r,\theta) = \int_{\phi_i-\Delta\phi_H/2}^{\phi_i+\Delta\phi_H/2} |F(\phi|r,\theta)|^2 d\phi.$$

We choose the following metric to evaluate the performance of a sensor array geometry.

$$J'(r,\theta) \triangleq \frac{P_s(r,\theta)}{P_i(r,\theta)\Delta\phi_H} = J'(\mathbf{x}),$$

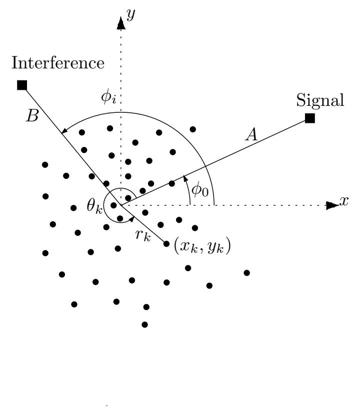


Fig. 4. Sensor array geometry.

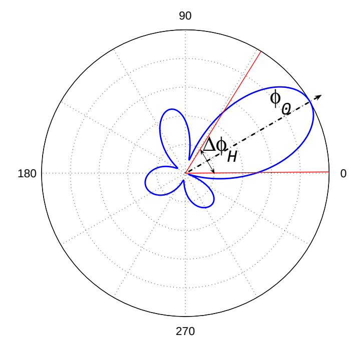


Fig. 5. Beam pattern and HPBW.

where $\mathbf{x}$ is the sensor array geometry in cartesian coordinates. We wish to maximize the value of this metric over various geometries. To elaborate this point further, we give some example geometries and their respective beam patterns in Figure 7. Here, we assume that the signal direction ϕ_0 is 0 degrees and the interference is coming at an azimuthal angle $\phi_i = 90$ degrees. For comparison, the values of the metric have also been given on top of the beam patterns. Note that the linear array has a narrower beam but a large leakage in the interference direction. Similarly, the circular geometry in the bottom has low interference but a fat beam (i.e a large $\Delta\phi_H$) in the direction of signal. The irregular patterns in the middle have a higher benefit, although they lie in between the two extremes of beam width and interference nullification. Moreover, as described in the above paragraphs, the metric to extremize may not be a function of the beamforming performance alone. Therefore it is reasonable to search over all possible geometries, rather than driving all nodes to a pre-determined regular geometry.

Note that this metric may be different for different realizations of a particular connectivity graph. We make use of the cylindrical algebraic decomposition (CAD) algorithm for computing reachability to get a representative geometry r, θ for the connectivity graph G. [3] In this way we let $J'(\mathbf{x}) = J'(G)$, where $\Phi_N(\mathbf{x}) = G$. If $J''(G)$ is some other performance metric associated with the function of the sensor network, then using the notation in Section 3, $\Psi(G) = \nu_1 J'(G) + \nu_2 J''(G)$. Similarly we chose $L(G_i, G_{i+1})$ according to a desired transient behavior in the network. We give a snapshot from one such simulation for a particular choice of cost metric in Figure 7. Here, we have purposely chosen a small number of nodes and a relatively less relative displacement to demonstrate the value of the graph processes. The signal and interference directions are the same as in Figure 6. As the graph process

[3] The details of this computation have been omitted for the sake of brevity.

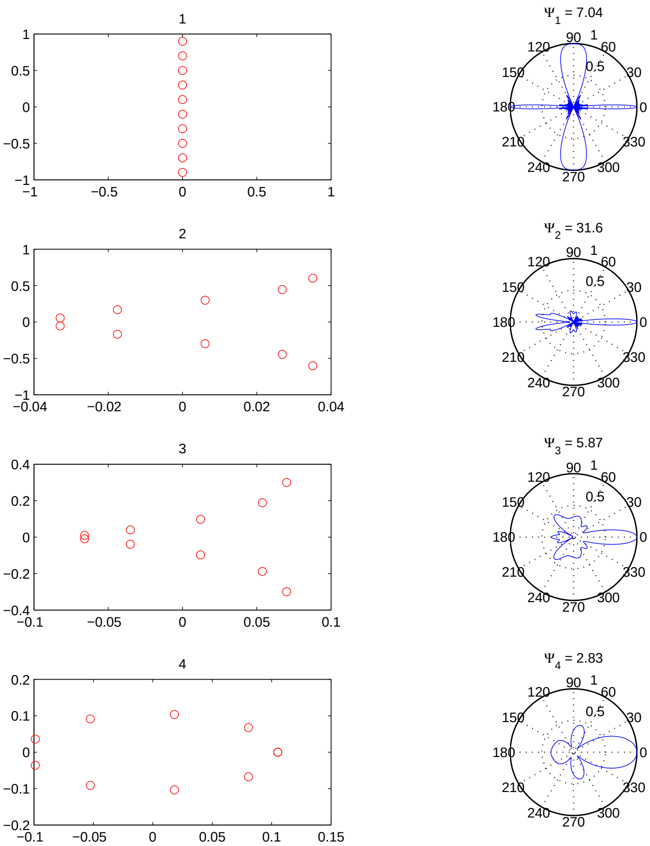


Fig. 6. Beamforming performance for various geometries.

evolves, notice the thinning of the main lobe in the signal direction. More pronounced is the attenuation in the interference direction, thus increasing the value of the metric at each transition.

4.3 Other Applications

In principle, the framework developed in Section 3 can be used for any application that required optimization over connectivity graphs. In a recent work [18, 19], it has been shown that the geometrical problem of determining coverage loss in a sensor network can be studied by looking at certain topological invariants of the topological spaces induced from the connectivity graph

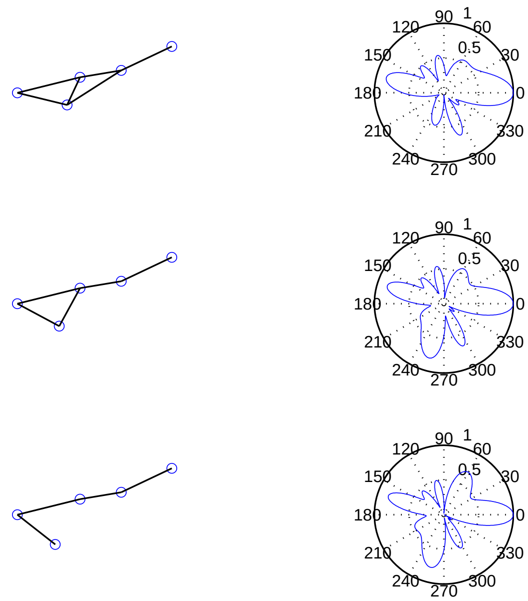


Fig. 7. Evolution of beam pattern in a connectivity graph process.

of the sensor network. This characterization associates a number with every connectivity graph that measures the number of coverage holes in the network. One can therefore use the connectivity graph processes to reduce the coverage holes in the network by setting up the appropriate terminal cost and the Lagrangian. A full investigation of this application (and many others) is a subject of current research.

Conclusions

We have presented a generic framework for connectivity graph processes. The concepts of feasibility and reachability are useful for obtaining optimal trajectories on the space of connectivity graphs. These graphical abstractions are computable using the techniques of semi-definite programming and CAD, as demonstrated by various simulation results. This framework can be used for a wide range of applications. In particular, the problems of producing low complexity formations and collaborative beamforming has be studied by using connectivity graph processes.

Acknowledgments

The authors wish to thank Mehran Mesbahi for discussing dynamic graphs and graph controllability concepts.

References

1. R.Saber and R. Murray, “Agreement Problems in Networks with Directed Graphs and Switching Topology,” in *Proc. IEEE Conference on Decision and Control*, 2003.
2. M. Mesbahi, “On a Dynamic Extension of the Theory of Graphs,” in *Proc. American Control Conference*, May 2002.
3. A. Jadbabaie, J. Lin, and A. Morse, “Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules,” *IEEE Transactions on Automatic Control*, Vol. 48, No. 6, pp. 988-1001, 2003.
4. H. Tanner, A. Jadbabaie and G. Pappas, Flocking in Teams of Nonholonomic Agents, *Cooperative Control*, Lecture Notes in Control and Information Sciences , Vol. 309, Pages 229-239, Springer, 2003.
5. J. Corts, S. Martnez, and F. Bullo, “Spatially-Distributed Coverage Optimization and Control with Limited-Range Interactions,” *ESAIM. Control, Optimisation & Calculus of Variations*, Vol. 11, Pages 691-719, 2005.
6. A. Muhammad and M. Egerstedt, “Connectivity Graphs as Models of Local Interactions.” in *Proc. IEEE Conference on Decision and Control*, 2004.
7. A. Muhammad and M. Egerstedt, “Connectivity Graphs as Models of Local Interactions,” *Journal of Applied Mathematics and Computation* , Vol. 168, Issue 1, Pages 243-269, September 2005.
8. S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan, *Linear Martix Inequalities in Systems and Control Theory*, SIAM Studies in Applied Mathematics, 1994.
9. The Math- Works Inc., “LMI Control Toolbox,” Version 1.0.7, May 2001.
10. S. Basu, R. Pollack and M. Roy, *Algorithms in Real Algebraic Geometry*, Algorithms and Computation in Mathematics Series, Vol. 10, Springer, 2003.
11. A. Muhammad and M. Egerstedt., Positivstellensatz Certificates for Non-Feasibility of Connectivity Graphs in Multi-agent Coordination, in *16th IFAC World Congress*, Prague, 2005.
12. A. Muhammad and M. Egerstedt, “Feasibility, Reachability and Optimal Control of Connectivity Graph Processes,” Submitted to *SIAM Journal on Control and Optimization.*
13. Abubakr Muhammad and Magnus Egerstedt, “Network Configuration Control Via Connectivity Graph Processes,” 2006 American Control Conference, Minneapolis. (Submitted)
14. A. Muhammad and M. Egerstedt, “On the Structural Complexity of Multi-Agent Agent Formations,” in *Proc. American Control Conference*, Boston, Massachusetts, USA, 2004.
15. H. Ochiai, P. Mitran, H. Poor and V. Tarokh, “Collaborative Beamforming for Distributed Wireless Ad Hoc Sensor Networks,” *IEEE Transactions on Signal Processing*. To appear.
16. J. Agre and L. Clare, “An Integrated Architecture for Cooperative Sensing and Networks,” *Computer*, Vol. 33, Pages 106-108, May 2000.
17. B. D. Van Veen and K. M. Buckley, “Beamforming: a versatile approach to spatial filtering,” *IEEE ASSP Magazine*, vol. 5, Apr. 1988.
18. R. Ghrist and A. Muhammad, “Coverage and Hole-Detection in Sensor Networks via Homology,” *The Fourth International Conference on Information Processing in Sensor Networks* (IPSN’05), UCLA, Los Angeles, CA, 2005.
19. V. de Silva, R. Ghrist and A. Muhammad, “Blind Swarms for Coverage in 2-D,” *Robotics: Science and Systems*, MIT, Cambridge, MA, 2005.