Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, July 24-28, 2006

Hierarchical Containment Control in Heterogeneous Mobile Networks

M. Ji[†], M. Egerstedt[†], G. Ferrari-Trecate[‡,⋆], and A. Buffa[∘]

***Abstract*— In this paper, the multi-agent containment problem is investigated. In particular, we focus our efforts on a continuation of our previous work on leader-follower networks, by imposing a hierarchical structure on the network topology. The main idea is to organize the team of autonomous, mobile agents in a multi-layered structure, which enables us to understand the tradeoffs between performance and complexity of the hierarchical topology, and we achieve this in the setting where we drive the collection of mobile agents to a target location while guaranteeing containment. In other words, the movement should be such that each layer stays within the convex hull of its parent layer while keeping its child layer inside its own convex hull. Such a criteria is to ensure that the movement of all the agents is constrained to a limited area while the group is moving. The application of this arises, for example, in surveillance, demining, and hazardous materials removing.**

***Keywords*— Multi-agent control, Network topology, Hierarchial structure**

I. Introduction

The research on multi-agent coordination can be characterized as falling into one of two camps depending on whether or not the robot group has designated leaders. In this paper, we strike a compromise between these two extremes through heterogeneous, hierarchical robot networks, where the robots are designated as leaders with respect to certain agents and followers with respect to others. This is achieved through a layering of the interaction network. The purpose for this is to study the potential tradeoffs between the structural complexity associated with the hierarchical network, and the performance it can achieve. In particular, we continue of work in [1] on containment control in which the agents in a particular layer L are to ensure that the agents in the next layer are to remain in the convex hull spanned by the agents in layer L. This is an issue that for instance arises when the team is asked to transport hazardous materials between areas in an orderly manner (confined to a particular area).

In this paper, we propose a multi-layer, hybrid Stop-Go policy for the hierarchical robot network and show how containment control can be achieved. We moreover define complexity and performance measures in terms of the information flow and algebraic connectivity in the network, and relate these to the number of layers in the network. The outline of the paper is as follows: in Section II we introduce some mathematical preliminaries as well as an analysis of how the hierarchial structure can balance the information flow. An extension of the hybrid Stop-Go policy to multi-layered structure is presented in Section III, and the tradeoffs between complexity and performance of the hierarchial structure is discussed in Section IV.

II. Hierarchical Network Topologies

In order to formalize matters, we first introduce some basic, mathematical preliminaries from graph theory (see for example [2]) and decentralized multi-agent control.

An undirected graph G is defined by a set $\mathcal{N}_G = \{1,\dots N\}$ of *nodes* and a set $\mathcal{E}_G \subset \mathcal{N}_G \times \mathcal{N}_G$ of *edges*. We will use $|\mathcal{N}_G|$ for denoting the cardinality of $\mathcal{N}_G$, and we say that two nodes x and y are *neighbors* if $(x,y) \in \mathcal{E}_G$. A *path* $x_0 x_1 \dots x_L$ is a finite sequence of nodes such that $x_{i-1} \sim x_i$, $i = 1,\dots,L$. A graph G is *connected* if there is a path connecting every pair of distinct nodes. G is *complete* if $\mathcal{E}_G = \mathcal{N}_G \times \mathcal{N}_G$.

Now, let $S = (\mathcal{N}_S, \mathcal{E}_S)$ be an undirected graph and let $\mathcal{N}_{S'} \subset \mathcal{N}_S$. The subgraph S' associated with $\mathcal{N}_{S'}$ is the pair $(\mathcal{N}_{S'}, \mathcal{E}_{S'})$ where $\mathcal{E}_{S'} = \{(x,y) \in \mathcal{E}_S : x \in \mathcal{N}_{S'}, y \in \mathcal{N}_{S'}\}$. This definition allows basic operations in set theory to be extended to graphs in the following manner: Let S_1 and S_2 be subgraphs of the graph S. Then, $S_1 \cup S_2$, $S_1 \cap S_2$, $S_1 \backslash S_2$ are the graphs associated with $\mathcal{N}_{S_1} \cup \mathcal{N}_{S_2}$, $\mathcal{N}_{S_1} \cap \mathcal{N}_{S_2}$, and $\mathcal{N}_{S_1} \backslash \mathcal{N}_{S_2}$, respectively.

For our purposes, we will often use graphs with a boundary. In particular, let S be a subgraph of G. The boundary of S is the subgraph $\partial S \subset G$ associated with $\mathcal{N}_{\partial S} \doteq \{y \in \mathcal{N}_G \setminus \mathcal{N}_S \ : \ \exists x \in \mathcal{N}_S \ : \ x \sim y\}$. The closure of S is $\bar{S} = \partial S \cup S$. Note that the definition of the boundary of a graph depends on the host graph G. This implies that if one considers three graphs $S' \subset S \subset G$, the boundaries of S' in S and in G may differ.

Now, as already mentioned, we will impose a hierarchical structure on the network topology. In particular, we let the agents be organized into M layers (encoded through subgraphs), where the inner-most layer is layer 1 and the outermost layer is layer M. We moreover assume that the

[†]Georgia Institute of Technology, School of Electrical and Computer Engineering, Atlanta, GA 30332,USA. {magnus, mengji}@ece.gatech.edu

[‡]INRIA, Domaine de Voluceau, Rocquencourt - B.P.105, 78153, Le Chesnay Cedex, France. Giancarlo.Ferrari-Trecate@inria.fr

[⋆]Dipartimento di Informatica e Sistemistica, Universitá degli Studi di Pavia Via Ferrata 1,27100 Pavia, Italy.

[∘]Istituto di Matematica Applicata e Tecnologie Informatiche, C.N.R., Via Ferrata, 1, 27100 Pavia, Italy. annalisa@imati.cnr.it

network topology is such that

$$\bigcup_{i=1}^{M} \mathcal{N}_{S_i} = \mathcal{N}_G, \ \mathcal{N}_{S_i} \cap \mathcal{N}_{S_j} = \emptyset, \ i \neq j$$

and

$$\begin{array}{l} \mathcal{N}_{\partial S_{i-1}} \subseteq \mathcal{N}_{S_i} \cup \mathcal{N}_{S_{i-2}}, \ \forall \ i = 3, \cdots, M \\ \mathcal{N}_{\partial S_1} \subseteq \mathcal{N}_{S_2} \\ \mathcal{N}_{\partial S_M} \subseteq \mathcal{N}_{S_{M-1}}, \end{array}$$

where S_i denotes the subgraph corresponding to layer i and ∂S_{i-1} is the (non-empty) boundary of S_{i-1} in the host graph G. Moreover, we assume that each subgraph is connected, and an example of such a topology is given in Figure 1.

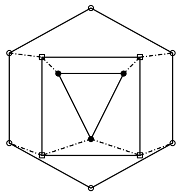


Fig. 1. A hierarchical layering of the network topology into three layers. The solid lines correspond to intra-layer edges while the dash-dotted lines correspond to inter-layer edges.

The control strategy that was developed in [1] is given by a hybrid Stop-Go control policy for the leaders, while the followers (that work only investigated the case with two layers) execute a simple, distributed control law. The basic idea behind this control policy was to ensure that the leaders come to a halt whenever a follower leaves the convex polytope Ω_L spanned by the leader, giving the followers the chance to return to Ω_L. In the hierarchical setting of this paper, this line of reasoning will be generalized through the convex polytopes $\Omega_i, \ i = 2, \ldots, M$, spanned by the agents in layer i. The main reason why such a layered approach is beneficial comes from the amount of information the different agents have to store. To see this, consider a generalization of the hybrid Stop-Go policy to affect agents in all layers except those at the inner-most layer (layer 1). In order to execute such a hybrid control policy, the individual agents in layer i must keep track of the agents in layer $i-1$ as well as its neighbors in layer i with whom they form faces in Ω_i.

In what follows, we assume, for notational simplicity, that we have a total of N agents, distributed equally across M layers. In other words, we have a total number of N/M agents in each layer. If the agents are planar, i.e. in 2D, each agent in $\mathcal{N}_{S_i}$ can at most form faces in Ω_i with two other agents in $\mathcal{N}_{S_i}$. However, each agent also has to keep track of all the agents in layer $i-1$, which implies that the total number of agents that each agent must keep track of (not considering agents in the inner-most layer) is

$$\mu_{2,M,N} = N/M + 3,$$

where the subscript 2 denotes the planar case. The first term corresponds to the agents in layer $i-1$, while the 3 corresponds to the two neighbors in Ω_i as well as the agent's own position. This number should be complemented with a situation where we only use two layers, and since $\mu_{2,M,N}$ is monotonously decreasing in M, we have that the total amount of information that must be stored by each agent (not in the inner-most layer) is strictly less in the hierarchical situation since $\mu_{2,M,N} < \mu_{2,2,N}, \ \forall M > 2$.

Now, if the agents are evolving in $\Re^n$, where $n > 2$, each agent may be forced to keep track of the positions of all but one of the agents in its layer, as shown in Figure 2. In other words, if we let $\bar{\mu}$ denote this worst-case scenario, we get the similar result

$$\bar{\mu}_{n,M,N} = 2N/M - 1 < \bar{\mu}_{n,2,N} = N - 1, \ \forall M > 2.$$

The conclusion to draw from this is that for each individual agent in the outer-most layer, a hierarchical structure is to prefer. However, this structure implies that agents not located in the outer-most layer will have to be more advanced as well in that they are executing a more complex control policy. This is both due to the fact that they need to store information about other agents, but they also must execute a more advanced strategy compared to the non-hierarchical case. As such, a hierarchical topology reduces the demands on the most advanced agents while it increases the demands on the less advanced agents, thus acting as an equalizer of the capabilities needed by the different agents.

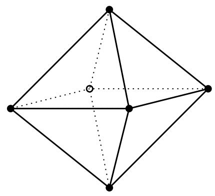


Fig. 2. Layer i in 3D, where each agent form a face in Ω_i with all but one agent.

III. Layered Containment Control

In [1] we developed a hybrid Stop-Go policy for the leader-follower (i.e. the two-layered) case. This policy can be easily extended to the hierarchical setting, and we here recall some of the basic ideas behind this control strategy.

First, we describe the basic Laplacian control law that will dictate the movements of the agents in the inner-most layer, i.e. for $x \in \mathcal{N}_{S_1}$. Let $r(x,t)$ be the position of the agent x at time $t \geq 0$. We now assume that the agents in layer 1 obey the simple dynamics $\dot{r}(x,t) = u(x,t)$, where

$$u(x,t) \doteq \Delta_1 r(x,t) \tag{1}$$

is the *Laplacian* [3], [4], [5], [6], [7], [8], [9] control law. It should be noted that this is a so-called Partial Difference

Equation, or PdE [8], [9], [10], where the *partial derivative* of r is defined as $\partial_y r(x,t) \doteq r(y,t) - r(x,t)$ and enjoys the following properties:

(1) $\partial_y r(x,t) = -\partial_x r(y,t)$,
(2) $\partial_x r(x,t) = 0$,
(3) $\partial_y^2 r(x,t) = -\partial_y r(x,t)$.

Moreover, the Laplacian of r is given by

$$\Delta r(x,t) \doteq - \sum_{y \in \mathcal{N}_G, y \sim x} \partial_y^2 r(x,t) = + \sum_{y \in \mathcal{N}_G, y \sim x} \partial_y r(x,t), \quad (2)$$

where the last identity follows from property (3). But, in Equation 1, the Laplacian is defined only with respect to the agents in layer 1 as well as the agents in layer 2 that define the convex polytope Ω_2. In a similar manner, we define the reduced Laplacian $\Delta_k,\ k = 1, \ldots, M-1$, as

$$\Delta_k r(x,t) \doteq - \sum_{y \in \mathcal{N}_{S_k} \cup \mathcal{N}_{S_{k+1}}, y \sim x} \partial_y^2 r(x,t). \quad (3)$$

Now, since $\mathcal{N}_{\partial S_1} \subseteq \mathcal{N}_{S_2}$, the only agents outside $\mathcal{N}_{S_1}$ that agents in $\mathcal{N}_{S_1}$ are connected to are those in $\mathcal{N}_{S_2}$. And, in [1] it was shown that if the agents in $\mathcal{N}_{S_2}$ are fixed and stationary, and $S_1 \cup S_2$ is connected, the agents in $\mathcal{N}_{S_1}$ will converge to Ω_2, i.e. to the convex polytope spanned by the agents in layer 2, under the Laplacian control law in Equation 2. Moreover, agents in layers $2, \ldots, M-1$ will execute a Laplacian Stop-Go policy in that agents in layer i will switch between executing a Laplacian control law when all agents in layer $i-1$ are in Ω_i (Go mode) to halting their evolution if an agent in layer $i-1$ leaves Ω_i (Stop mode). In order to avoid blocking executions, we relax the latter condition and let agents in the i-th layer enter the Stop mode if an agent in the layer $i-1$ is at distance ϵ outside Ω_i [1]. Since, in the halting mode, the agents in layer $i+1$ will return to Ω_i, as shown in [1], this policy will ensure that containment is ensured up to the safety margin ϵ around Ω_i, that can be made arbitrarily small.

The only remaining agents are the leaders, i.e. the agents in layer M. They will also halt their execution (Stop mode) if agents in layer $M-1$ are outside Ω_M. The only difference is that these agents will be given a goal-oriented motion in the Go mode. This is done in order to ensure that the team reaches its desired target location/formation. In other words, we have the two modes for the leaders

$$STOP: \quad \dot{r}(x,t) = 0, \quad x \in \mathcal{N}_{S_M} \quad (4)$$

And, the second control mode under consideration is the Go mode, in which the leaders move toward a given target location/formation/shape. A number of different control laws can be defined for this, but we, for the sake of conceptual unification, let the Go mode be given by a Laplacian-based control strategy as well:

$$GO: \quad \dot{r}(x,t) = \Delta_L (r(x,t) - r_T(x)), \quad x \in \mathcal{N}_{S_M}, \quad (5)$$

where $r_T(x),\ x \in \mathcal{N}_{S_M}$ denotes the desired target position of leader x, and where we use the "Leader Laplacian" Δ_L to denote the Laplacian operator defined solely over the subgraph S_M, i.e.

$$\Delta_L f(x,t) \doteq - \sum_{y \sim x,\ y \in \mathcal{N}_M} \partial_y^2 f(x,t),$$

for any function f defined over the graph [10]. Now, under the assumption that S_M is connected, and, by exactly the same reasoning as for the standard rendezvous problem [11], the leaders will converge to positions $r_L(x)$ such that $\partial_y r_L(x) = \partial_y r_T(x),\ \forall x, y \in \mathcal{N}_{S_M}$. In other words, no convergence to a predefined point is achieved. Rather, this control law ensures that the leaders arrive at a translationally invariant target formation.

As an example, the evolution of a 9-agent 3-layer structure is shown in Fig. 3, where the outer layer is asked to rotate by $\frac{2}{3}\pi$. The switching sequences of the process is shown in Fig. 4.

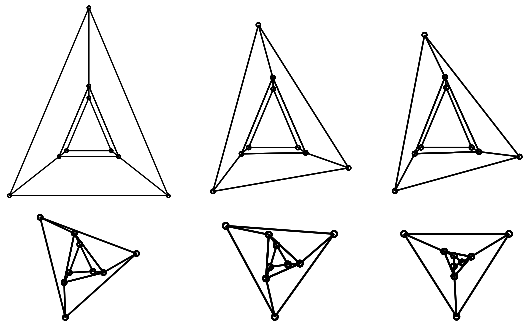


Fig. 3. Hierarchical, multi-layered containment control with 3 layers and 3 agents in each layer.

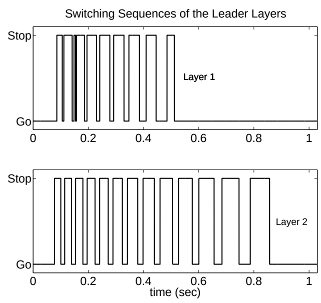


Fig. 4. The switching sequences associated with Layers 1 and 2 during the process depicted in Figure 3.

IV. Tradeoffs between Complexity and Performance

In this section, we will discuss how the multi-layered structure affects the performance of the multi-agent group. Two major issues will be considered, namely structural complexity and convergence speed.

The structural complexity measures of a network topology measure certain aspects of the information flow through the network. For instance, the *Kirchhoff complexity*, $C_K(G)$, of a graph G is given by

$$C_K(G) = \log(\tau(G)),$$

where $\tau(G)$ is the number of spanning trees in G. And, by Kirchhoff's theorem [12], $\tau(G)$ is related to the spectrum of the so-called graph Laplacian through

$$\tau(G) = \frac{1}{N}\prod_{i=2}^{N}\lambda_i(G).$$

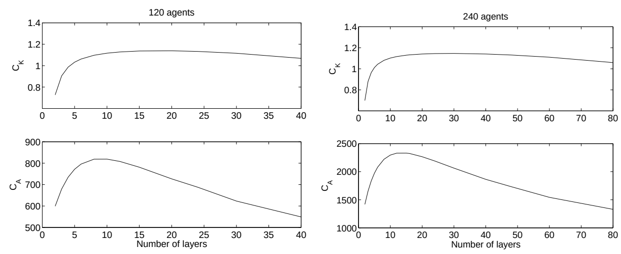
Here $\lambda_i(G)$ is the ith smallest eigenvalue of the *graph Laplacian* $\mathcal{L}(G)$ (see [2] for the definition of $\mathcal{L}(G)$).

In addition to the Kirchhoff complexity, we will also study a complexity measure based on the shortest pairwise path length between the agents [13], which is given by

$$C_A(G) = \sum_{x\in\mathcal{N}(G)}(d_x + \sum_{y\in\mathcal{N}(G)\setminus star(x)}\frac{d_y}{k_{xy}}),$$

where d_x is the degree of node x and k_{xy} is the length of the shortest path between x and y. Moreover, $star(x)$ denotes the star graph centered at x.

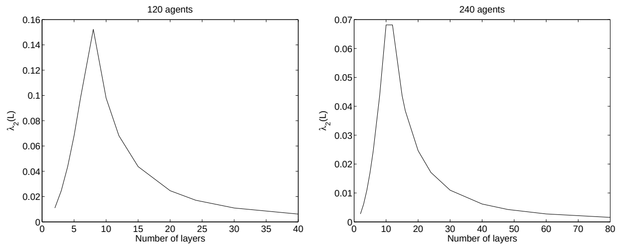
Fig 5 shows examples in which an equal number of agents belong to each layer. Shown are the Kirchhoff and distance complexity associates with 120 and 240 agents as functions of the number of layers. From the figure, we can see that as the number of the layers increases, both complexities increase first and then decrease after reaching a peak. By comparison, we see that C_A's peaks are much more apparent, while C_K is almost flat after reaching the peak. This difference shows that, in the multi-layered structure, the number of spanning trees does not change dramatically with respect to the number of layers, L, if L is not too small (in our case $L > 10$). On the other hand, the distance-related measure, C_A, is sensitive to L.



(a) 120 nodes (b) 240 nodes

Fig. 5. Complexity measures of the layered structure.

Now, the convergence rate of a multi-agent group has recently been shown [7] to be related to the second smallest eigenvalue of the graph Laplacian, $\lambda_2(G)$, which is also called the *algebraic connectivity*. The algebraic connectivity determines how fast the agent group can form a consensus or reach an agreement. The algebraic connectivity of different number of layers are plotted in Fig 6, where sharp peaks can be observed at 8 layers for 120 nodes and 10-12 layers for 240 nodes.

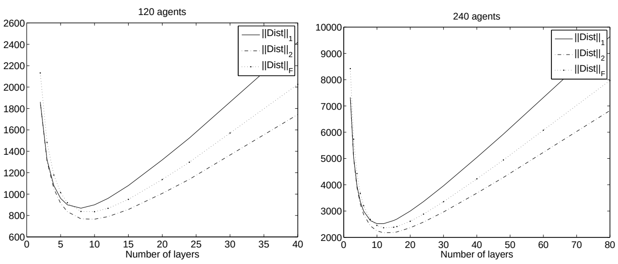


(a) 120 nodes (b) 240 nodes

Fig. 6. λ_2 of the graph Laplacian matrix.

The peaks' location can be interpreted through the information delay of multi-hops needed to propagate information through the graph. When the inter-layer delay and the intra-layer delay are balanced, i.e. when the maximum inter-layer delay is approximately equal to the maximum delay in the same layer, a peak can be observed. In other words, when the longest multi-hop between different layers, $M-1$, is equal or close to the longest multi-hop in each individual layer, $N/2M-1$, an maximal λ_2 can be observed. That is to say that the optimal number of layer is approximately equal to $\sqrt{N/2}$, which is verified by Fig 6.

In order to reveal the relationship between the algebraic connectivity and the pairwise distances, we define the distance matrix as $K = [k_{ij}],\ i,j = 1,2,\ldots,N$. Here i and j are indices running through the node set, and k_{ij} is, as defined before, the length of the shortest path between node i and node j. Different norms of the distance matrices are shown in Fig. 7, where we can see that the locations of minimum of the norms, especially the 1-norm, are consistent with the peaks of λ_2. The consistency is not surprising, since λ_2 has been shown to be related to distance invariants of the graph [12]. Further verification of our hypothesis is left to the future work.



(a) 120 nodes (b) 240 nodes

Fig. 7. Norms of the distance matrix.

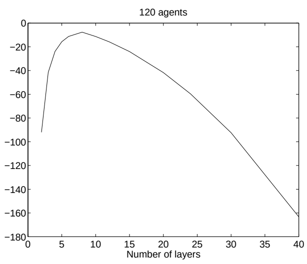
Now let us look at the tradeoff between complexity and the convergence rate. Generally speaking, one would prefer a structure with high convergence rate and low complexity. However, these two properties act in an opposite way in that the fastest convergence rate is associated with the complete graph, which has the highest complexity. On the other hand, the line graph has the lowest complexity [13], while it has the slowest convergence rate. As to our special topology, high complexities also appear in the same range where peaks of convergence rates occur, as shown in Fig. 5 and 6. We thus need to choose a structure that balances these two properties.

Different performance indices can be constructed for different situations, where complexity and convergence might have different weights. Here we propose two performance index functions, J_1 and J_2, which combine the consideration of complexity and convergence rate

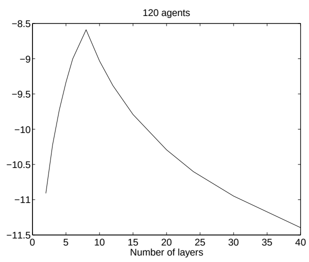
$$J_1 = -\frac{1}{\lambda_2} - C_K, \quad (6)$$

$$J_2 = \log(\lambda_2 / C_A). \quad (7)$$

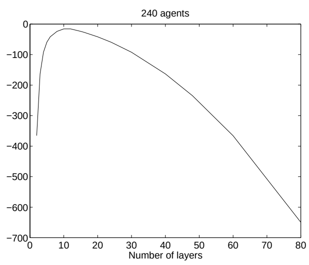
J_1 utilizes the Kirchhoff complexity, while J_2 uses the distance related complexity. They are depicted in Fig. 8, and the optimal structure for 120 nodes (8 layers) is shown in Fig. 9.



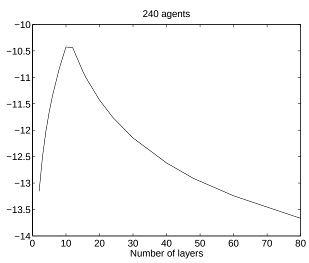
(a) J_1:120 nodes



(b) J_2:120 nodes



(c) J_1:240 nodes



(d) J_2:240 nodes

Fig. 8. Performance indices of the multi-layered multi-agent group.

ACKNOWLEDGMENT

The work by Magnus Egerstedt and Meng Ji was supported by the U.S. Army Research Office through Grant #99838. The work by Giancarlo Ferrari-Trecate was partially supported by the European Commission under the Network of Excellence HYCON, contract number FP6-IST-511368.

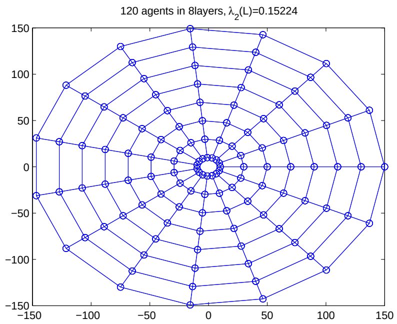


Fig. 9. The optimal layered structure for 120 agents: 8 layers.

REFERENCES

[1] Ferrari-Trecate, G., Egerstedt, M., Buffa, A., Ji, M.: Laplacian sheep: A hybrid, stop-go policy for leader-based containment control. (To appear in Hybrid Systems: Computation and Control, Santa Barbara, CA, March 2006)

[2] Bollobás, B.: Modern graph theory. Graduate texts in Mathematics. Springer-Verlag (1998)

[3] Tanner, H., Jadbabaie, A., Pappas, G.: Flocking in fixed and switching networks. (2005) `http://www.seas.upenn.edu/~pappasg/publications.htm`. Submitted.

[4] Saber, R.O.: A unified analytical look at Reynolds flocking rules. Technical Report CIT-CDS 03-014, California Institute of Technology (2003)

[5] Muhammad, A., Egerstedt, M.: Connectivity graphs as models of local interactions. (To appear in Journal of Applied Mathematics and Computation, 2005)

[6] Jadbabaie, A., Lin, J., Morse, A.S.: Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Trans. on Automatic Control **48** (2003) 988 – 1001

[7] Olfati-Saber, R., Murray, R.: Consensus problems in networks of agents with switching topology and time-delays. IEEE Trans on Autom. Control **49** (2004) 101–115

[8] Ferrari-Trecate, G., Buffa, A., Gati, M.: Analysis of coordination in multiple agents formations through Partial difference Equations. Technical Report 5-PV, IMATI-CNR (2004) `http://www-rocq.inria.fr/who/Giancarlo.Ferrari-Trecate/publications.html`.

[9] Ferrari-Trecate, G., Buffa, A., Gati, M.: Analysis of coordination in multi-agent systems through partial difference equations. Part I: The Laplacian control. 16th IFAC World Congress on Automatic Control (2005)

[10] Bensoussan, A., Menaldi, J.L.: Difference equations on weighted graphs. Journal of Convex Analysis (Special issue in honor of Claude Lemaréchal) **12** (2005) 13–44

[11] Ji, M., Muhammad, A., Egerstedt, M.: Leader-based multi-agent coordination: Controllability and optimal control. Accepted by the American Control Conference, Minneapolis, MN, (Jun. 2006)

[12] Mohar, B.: The laplacian spectrum of graphs. In: Sixth International Conference on the Theory and Applications of Graphs. (1988) 871–898

[13] Muhammad, A., Egerstedt, M.: On the structural complexity of multi-agent robot formations. In: American Control Conference 2004, Boston, MA (2004)