

Distributed Formation Control with Permutation Symmetries

Michael M. Zavlanos and George J. Pappas

Abstract—In distributed control applications such as coverage or consensus by multiple mobile agents, a great new challenge is the development of motion algorithms that dynamically determine the positions of the agents in the formation using only local information. In this paper, we address this challenge using two novel ideas. First, we represent a formation as a rotational and translational invariant configuration in the free space and employ distributed consensus algorithms to guarantee that all agents *agree* on the rotation and translation of the final formation configuration. Second, local market-based coordination protocols dynamically determine a permutation of the agents in the formation, while artificial potential fields are used to drive the group of agents to the desired formation. Integration of the overall system results in a distributed, multi-agent, hybrid system which, under a connectivity assumption on the underlying communication network, is shown to always converge to the desired formation. Furthermore, the number of explored permutations is at most polynomial with the number of agents, while scalability of our approach is illustrated by nontrivial computer simulations.

I. INTRODUCTION

Coordinated motion and cooperative control has recently emerged as a fundamental problem in the control of multi-agent systems. Motivations come from the area of controlling formations of ground or aerial vehicles with applications in air traffic control, satellite clustering, automatic highways, mobile robotics and mobile sensor networks. One of the main goals is to achieve a coordinated objective while using only local information. The objective investigated in this paper is that of controlling the formation of a group of mobile agents.

In the area of multi-agent systems, a formation is roughly defined as a configuration of the agents in the free space characterized by given inter-agent distances. Depending explicitly on inter-agent distances, a formation can be realized anywhere in the free space. In general, an explicit assignment of the agents in a formation might be desired, however, for indistinguishable agents, this is not always the case. In particular, formations resulting from distinct permutations of indistinguishable agents are called identical *modulo permutation* and consist the main focus of this paper. In this context, the problem of formation control consists of controlling the relative positions of the agents with respect to formations that are identical up to a permutation.

Due to its frequent appearance in the context of multi-agent systems, formation control has already received considerable attention. Problems that have been investigated

include motion feasibility of formations [1], reaching a formation [2], [3], [4], preserving a formation pattern [5] and switching between different formations [6]. Approaches to these problems range from behavior-based [7] and leader-follower [8], [9], [10], to paradigms involving virtual structures [11] and rigid-body type formations [12]. Common in all these approaches is a given assignment of the agents in the desired formation. On the other hand, the assignment problem per se, has recently been studied in the context of destination or target allocation in distributed robotics. Approaches are either on-line [13], [14] or off-line [15], [16], [17], depending on whether the discrete assignment is addressed simultaneously with the continuous navigation strategies or is solved independently in advance.

As recent applications in formation control, such as coverage [18] or consensus [19], [20], [21], [22], [23] by mobile sensor networks, involve increasingly large numbers of indistinguishable mobile agents, the need for fully distributed feedback control frameworks also able to determine the positions of the agents in the formation becomes vital. In this paper, we address this challenge using a novel control decomposition. In particular, we represent a formation as a rotational and translational invariant configuration in the free space and employ distributed consensus algorithms to guarantee that all agents *agree* on the rotation and translation of the final configuration of the formation. On the other hand, the permutation of the agents in the formation is determined dynamically, by means of distributed market-based coordination protocols, where the agents bid for their position in the desired formation. Navigation of the agents to the desired formation is, then, due to artificial potential fields composed with the aforementioned coordination schemes. Integration of the overall system results in a distributed, multi-agent, hybrid system which, under certain connectivity assumptions on the underlying communication network, is shown to always reach the desired formation. Furthermore, the number of explored permutations is at most polynomial, dramatically reducing the combinatorial nature of purely discrete assignment problems. Our scalable approach is illustrated with nontrivial computer simulations.

The rest of this paper is organized as follows. In Section II we define the problem of distributed formation control with permutation symmetries and develop a necessary graph theoretic background. In Section III we elaborate on the consensus, navigation and market-based controller specifications, while in Section IV, we integrate the above controllers into a hybrid control scheme for every agent. The overall system is studied in Section V, while in Section VI, our approach is illustrated with nontrivial computer simulations.

This work is partially supported by ARO MURI SWARMS Grant W911NF-05-1-0219 and the NSF ITR Grant 0324977.

Michael M. Zavlanos and George J. Pappas are with GRASP Laboratory, Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA {zavlanos, pappasg}@grasp.upenn.edu

II. PROBLEM FORMULATION

Consider n identical fully actuated agents in \mathbb{R}^p , such that every agent i is described by,

$$\dot{x}_i(t) = u_i(t) \quad (1)$$

where $x_i(t) \in \mathbb{R}^p$ denotes the position vector of agent i at time t and $u_i(t) \in \mathbb{R}^p$ is a control input to be determined. Define, further, a *Euclidean Distance Matrix* (EDM) $C = (c_{ij}) \in \mathbb{R}^{n \times n}$, such that $c_{ii} = 0$ for all i and $c_{ij}^{1/2} \leq c_{ik}^{1/2} + c_{kj}^{1/2}$ for all $i, j, k = 1, \dots, n$, i.e., the entries of C satisfy the *triangle inequality*. Given any EDM $C = (c_{ij}) \in \mathbb{R}^{n \times n}$, we can define a *formation* in \mathbb{R}^p by,

$$\mathcal{F}_C = \{f_1, \dots, f_n \in \mathbb{R}^p \mid \|f_i - f_j\|_2^2 = c_{ij}, \forall i \neq j\} \quad (2)$$

Let $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ denote any *permutation* of integers $\{1, \dots, n\}$. Then, the problem addressed in this paper can be stated as follows.

Problem 1: Given any formation \mathcal{F}_C , determine distributed control laws $u_i(t)$ for all agents i that guarantee $\lim_{t \rightarrow \infty} \|x_i - x_j\|_2^2 = c_{\pi(i)\pi(j)}$ for all $i \neq j$ and any permutation π .

In other words, Problem 1 implies that we want the group of agents to converge to the formation \mathcal{F}_C . Note that the permutation π of the agents in the formation \mathcal{F}_C is not provided a priori hence, it needs to be determined *dynamically*. We achieve this goal using an equivalent representation of a formation. In particular, let $\bar{f}_1, \dots, \bar{f}_n \in \mathcal{F}_C$ denote any realization of the formation \mathcal{F}_C in \mathbb{R}^p and observe that,

$$\mathcal{F}_C = \{f_1, \dots, f_n \in \mathbb{R}^p \mid f_i = y + R\bar{f}_i, \forall i, y, R\} \quad (3)$$

where $y \in \mathbb{R}^p$ and $R \in SO(p)$ denote a translation vector and rotation matrix, respectively (Figure 1). Clearly, representations (2) and (3) of a formation are equivalent since,

$$\|f_i - f_j\|_2^2 = \|y + R\bar{f}_i - y - R\bar{f}_j\|_2^2 = \|R(\bar{f}_i - \bar{f}_j)\|_2^2 = c_{ij}$$

for all $i, j = 1, \dots, n$.¹ Observe that representation (3) explicitly describes a formation \mathcal{F}_C in terms of the points $f_1, \dots, f_n \in \mathbb{R}^p$ rather than the inter-agent distances c_{ij} . Hence, we can reformulate Problem 1 as follows.

Problem 2: Given any realization $\bar{f}_1, \dots, \bar{f}_n \in \mathcal{F}_C$ of a formation \mathcal{F}_C , determine distributed control laws $u_i(t)$ that guarantee $\lim_{t \rightarrow \infty} x_i = y + R\bar{f}_{\pi(i)}$ for all agents i , any permutation π , translation vector $y \in \mathbb{R}^p$ and rotation matrix $R \in SO(p)$.

Designing control inputs for the agents according to the specifications of Problem 2, critically relies on notions from graph theory. In particular, given the system of agents in system (1), we can define a *dynamic graph* $\mathcal{G}(t)$ as follows.

Definition 2.1 (Dynamic Graph): We call $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ a dynamic graph consisting of a set of vertices $\mathcal{V} = \{1, \dots, n\}$ indexed by the set of agents and a time varying set of links $\mathcal{E}(t) = \{(i, j) \in \mathcal{V} \times \mathcal{V} \mid \|x_i(t) - x_j(t)\|_2 < r\}$.

¹Note that $\|Rf\|_2 = \|f\|_2$ for any vector $f \in \mathbb{R}^p$ and any rotation matrix $R \in SO(p)$.

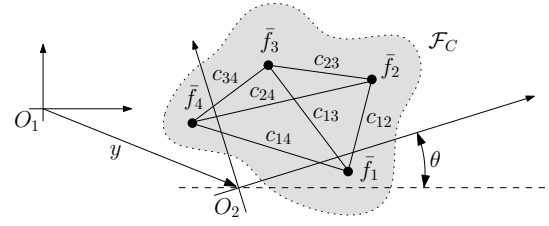


Fig. 1. A formation \mathcal{F}_C of 4 agents translated by a vector y and rotated by an angle θ . The realization $f_1, \dots, f_4 \in \mathcal{F}_C$ is with respect to the coordinate frame O_2 .

Dynamic graphs $\mathcal{G}(t)$ such that $(i, j) \in \mathcal{E}(t)$ if and only if $(j, i) \in \mathcal{E}(t)$ are called *undirected* and consist the main focus of this paper.² Furthermore, for any pair of vertices i and j such that $(i, j) \in \mathcal{E}(t)$ we write $i \sim j$ and say that vertices i and j are adjacent, or neighbors, at time t . A topological invariant of graphs that is of particular interest for the purposes of this work is graph connectivity.

Definition 2.2 (Graph Connectivity): We say that a dynamic graph $\mathcal{G}(t)$ is connected at time t if there exists a path, i.e., a sequence of distinct vertices such that consecutive vertices are adjacent, between any two vertices in $\mathcal{G}(t)$.

Given any collection of m distinct instances of $\mathcal{G}(t)$, i.e., $\{\mathcal{G}(t_1), \dots, \mathcal{G}(t_m)\}$, we say that the collection $\{\mathcal{G}(t_1), \dots, \mathcal{G}(t_m)\}$ is *jointly connected* if the union of its members is a connected graph [22].

III. DISTRIBUTED COORDINATION

Consider n agents in \mathbb{R}^p and a formation \mathcal{F}_C expressed as in equation (3) with realization $\bar{f}_1, \dots, \bar{f}_n \in \mathcal{F}_C$ that is known to all agents. The goal in this section is to develop a distributed control framework able to drive the agents to the desired formation \mathcal{F}_C for any translation vector $y \in \mathbb{R}^p$, rotation matrix $R \in SO(p)$ and permutation π , that are to be determined *dynamically*, using only local information. In particular, under certain connectivity assumptions on the communication network $\mathcal{G}(t)$, we employ *consensus* protocols from the literature [22] to get $y \in \mathbb{R}^p$ and $R \in SO(p)$ that are common for all agents, while the permutation π is determined dynamically by means of distributed *market-based* coordination.

A. Consensus on the Rotation and Translation

Let $y_i \in \mathbb{R}^p$ and $R_i \in SO(p)$ denote the translation vector and rotation matrix of agent i , respectively, and note that in \mathbb{R}^3 (similarly in \mathbb{R}^2), the rotation R_i can be written in terms of the angles ϕ_i , ψ_i and θ_i as,

$$R_i = R_z(\theta_i)R_y(\psi_i)R_x(\phi_i) \quad (4)$$

where $R_x(\phi_i)$, $R_y(\psi_i)$ and $R_z(\theta_i)$ denote the *roll*, *pitch* and *yaw* rotations along the x , y and z directions, respectively. Furthermore, given any communication network $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, let $\mathcal{N}_i(t) = \{j \mid (i, j) \in \mathcal{E}(t)\}$ denote the neighbors of agent i at time t . Then, the following theorem

²Dynamic graphs $\mathcal{G}(t)$ as in Definition 2.1, are sometimes also called *proximity graphs*.

guarantees consensus of all agents regarding their rotation and translation variables.

Theorem 3.1 ([22]): Let $y_i(t_0)$, $\phi_i(t_0)$, $\psi_i(t_0)$ and $\theta_i(t_0)$ be fixed and assume that every collection of communication graphs $\{\mathcal{G}_1, \dots, \mathcal{G}_m\}$ in every bounded time interval is *jointly connected*. Then, the system,

$$[\dot{y}_i^T \ \dot{\phi}_i \ \dot{\psi}_i \ \dot{\theta}_i] = - \sum_{j \in \mathcal{N}_i(t)} ([y_i^T \ \phi_i \ \psi_i \ \theta_i] - [y_j^T \ \phi_j \ \psi_j \ \theta_j]) \quad (5)$$

guarantees that $\lim_{t \rightarrow \infty} [y_i^T \ \phi_i \ \psi_i \ \theta_i] = [y^T \ \phi \ \psi \ \theta]$ for all i .

Note that the asymptotic values for y , ϕ , ψ and θ depend on the initial values $y_i(t_0)$, $\phi_i(t_0)$, $\psi_i(t_0)$ and $\theta_i(t_0)$ of every agent i as well as on the sequence of communication graphs $\mathcal{G}(t)$. Using Theorem 3.1, we can show the following result.

Proposition 3.2: Given any realization $\bar{f}_1, \dots, \bar{f}_n \in \mathcal{F}_C$ of a formation \mathcal{F}_C , assume that the conditions of Theorem 3.1 hold. Then, for any given permutation π and any constant $K > 0$, the control laws,

$$u_i(t) = -K \nabla_{x_i} \|x_i - y_i - R_i \bar{f}_{\pi(i)}\|_2^2, \quad i = 1, \dots, n \quad (6)$$

together with the consensus algorithm (5), guarantee that $\lim_{t \rightarrow \infty} x_i = y + R \bar{f}_{\pi(i)}$ for all agents i .

Proof: Observe first that the consensus algorithm (5) does not depend on the agents' states $x_i \in \mathbb{R}^p$ and so Theorem 3.1 guarantees convergence of the translation vectors $y_i \in \mathbb{R}^p$ and rotation matrices $R_i \in SO(p)$ to common values $y \in \mathbb{R}^p$ and $R = R_z(\theta)R_y(\psi)R_x(\phi) \in SO(p)$, respectively, for all agents.³ Hence, the control laws (6) asymptotically become $u_i(t) = -K \nabla_{x_i} \|x_i - y - R \bar{f}_{\pi(i)}\|_2^2$ and clearly guarantee that $\lim_{t \rightarrow \infty} x_i = y + R \bar{f}_{\pi(i)}$ for all agents i . To see this, just note that $V_i(x_i) = \frac{1}{2} \|x_i - y - R \bar{f}_{\pi(i)}\|_2^2$ is a Lyapunov function associated with agent i . ■

In other words, Proposition 3.2 implies that, given any permutation π , control laws (6) and the proposed consensus algorithm (5) guarantee that the group of agents eventually converges to formation \mathcal{F}_C .⁴ Hence, the major challenge is to determine a permutation π in a distributed way. To achieve this goal, we employ a *market-based* coordination framework, where every agent *bids* for a position $\pi(i)$ in the permutation π . The resulting auction is distributed, since it takes place in the absence of a centralizing auctioneer.

B. Market-Based Coordination

Let $\mathcal{I}_0 = \{1, \dots, n\}$ denote the index set of all available positions in a formation \mathcal{F}_C . We say that a position $k \in \mathcal{I}_0$ is being “sold” in the market if there is at least one agent i such that $\pi(i) = k$. Let $\mathcal{I}(t)$ denote the index set of all positions that are *not* “sold” in the market at time t and $\mathcal{I}^c(t) = \mathcal{I}_0 \setminus \mathcal{I}(t)$ denote the complement of $\mathcal{I}(t)$. We call $\mathcal{I}(t)$ and $\mathcal{I}^c(t)$ the sets of *available* and *taken* positions, respectively. Similarly, let $\mathcal{I}_i^a(t)$ and $\mathcal{I}_i^t(t)$ denote the index sets of available and taken positions from the perspective of

³Note that control of the angles is performed in \mathbb{R}^p and the mapping $\mathbb{R}^p \rightarrow SO(p)$ in equation (4) is injective.

⁴Note that for any formation formation \mathcal{F}_C and any permutation π , if $y + R \bar{f}_1, \dots, y + R \bar{f}_n \in \mathcal{F}_C$, then $y + R \bar{f}_{\pi(1)}, \dots, y + R \bar{f}_{\pi(n)} \in \mathcal{F}_C$.

Algorithm 1 Auction Mechanism for Agent i

Require: A position $\pi(i) \in \mathcal{I}_0$ and bid $b_i \geq 0$

- 1: Compute the set of agents bidding for the same position $\pi(i)$, i.e., $\mathcal{A}_i := \{j \in \mathcal{N}_i \mid \pi(j) = \pi(i)\}$
 - 2: **if** $\mathcal{A}_i = \emptyset$ **then**
 - 3: Exchange lists with neighbors, i.e.,
 $\mathcal{I}_i^a := \mathcal{I}_i^a \setminus (\cup_{j \in \mathcal{N}_i} \mathcal{I}_j^t)$ and $\mathcal{I}_i^t := \mathcal{I}_i^t \cup (\cup_{j \in \mathcal{N}_i} \mathcal{I}_j^a)$
 - 4: Update neighbors, $\mathcal{N}_i = \{j \mid x_j \in \mathcal{B}_r(x_i)\}$
 - 5: **else if** $\mathcal{A}_i \neq \emptyset$ **then**
 - 6: **if** $b_i > \max_{j \in \mathcal{A}_i} \{b_j\}$ **then**
 - 7: Exchange lists with neighbors, i.e.,
 $\mathcal{I}_i^a := \mathcal{I}_i^a \setminus (\cup_{j \in \mathcal{N}_i} \mathcal{I}_j^t)$ and $\mathcal{I}_i^t := \mathcal{I}_i^t \cup (\cup_{j \in \mathcal{N}_i} \mathcal{I}_j^a)$
 - 8: Update neighbors, $\mathcal{N}_i = \{j \mid x_j \in \mathcal{B}_r(x_i)\}$
 - 9: **else if** $b_i = \max_{j \in \mathcal{A}_i} \{b_j\}$ **then**
 - 10: Bid higher for the same position $\pi(i)$, i.e.,
 $b_i := g([b_i, \infty))$
 - 11: Update neighbors, $\mathcal{N}_i = \{j \mid x_j \in \mathcal{B}_r(x_i)\}$
 - 12: **else if** $b_i < \max_{j \in \mathcal{A}_i} \{b_j\}$ **then**
 - 13: Choose a new position among the available ones,
i.e., $\pi(i) := g(\mathcal{I}_i^a \setminus (\cup_{j \in \mathcal{N}_i} \mathcal{I}_j^t))$
 - 14: Bid higher for the new position $\pi(i)$, i.e.,
 $b_i := g([b_i, \infty))$
 - 15: Exchange lists with neighbors, i.e.,
 $\mathcal{I}_i^a := \mathcal{I}_i^a \setminus (\{\pi(i)\} \cup (\cup_{j \in \mathcal{N}_i} \mathcal{I}_j^t))$ and
 $\mathcal{I}_i^t := \mathcal{I}_i^t \cup \{\pi(i)\} \cup (\cup_{j \in \mathcal{N}_i} \mathcal{I}_j^a)$
 - 16: Update neighbors, $\mathcal{N}_i = \{j \mid x_j \in \mathcal{B}_r(x_i)\}$
 - 17: **end if**
 - 18: **end if**
-

agent i , respectively. The sets $\mathcal{I}_i^a(t)$ and $\mathcal{I}_i^t(t)$ are initialized such that every agent has knowledge of all available positions in \mathcal{I}_0 , i.e., $\mathcal{I}_i^a(t_0) = \mathcal{I}_0$ and $\mathcal{I}_i^t(t_0) = \emptyset$, while $\mathcal{I}_i^a(t) \cap \mathcal{I}_i^t(t) = \emptyset$ and $\mathcal{I}_i^a(t) \cup \mathcal{I}_i^t(t) = \mathcal{I}_0$ for all time $t \geq t_0$, i.e., no position can be considered both available and taken.

Having characterized the positions in the formation \mathcal{F}_C as available or taken depending on whether they are being “sold” in the market or not, we now describe the auction mechanism itself. In the absence of a centralizing auctioneer, local communication among the agents is required to achieve the desired coordination. As before, let $\mathcal{N}_i(t) = \{j \mid (i, j) \in \mathcal{E}(t)\}$ denote the set of neighbors of agent i at time t and define a *selection* function $g: 2^{\mathbb{R}^+} \rightarrow \mathbb{R}_+$ by,

$$g(X) = x \in X$$

where $x \in X$ can be chosen according to any policy, deterministic or not. Then, the proposed auction mechanism is described in Algorithm 1 and the corresponding market-based coordination scheme consists of multiple such auctions taking place locally among groups of agents.

Remark 3.3: We assume that bidding is *sound* in the sense that every *tie* on the bids is eventually broken (line 9 of Algorithm 1). Moreover, we require that every *new* position that is being “sold” is chosen among the available ones (line 13 of Algorithm 1). This requirement guarantees progress towards reaching a final permutation since, it does not allow continuous bidding for any particular position in π .

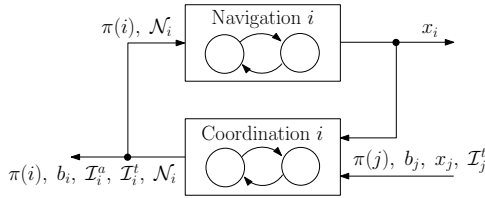


Fig. 2. Model for Agent i .

IV. MODELING THE AGENTS

Integration of the continuous time control laws in Subsection III-A with the discrete coordination protocols in Subsection III-B leads to a hybrid model for every agent [25] that consists of a *navigation* and a *coordination* automaton, as shown in Figure 2. *Inputs* to the navigation automaton of agent i consist the position $\pi(i)$ and neighbor set $\mathcal{N}_i(t)$, while *output* is the updated state $x_i(t)$. On the other hand, *inputs* to the coordination automaton of agent i consist the states $x_j(t)$, positions $\pi(j)$, bids $b_j(t)$ and sets of taken positions $\mathcal{I}_j^t(t)$ of all agents, while *outputs* are its own updated neighbor set $\mathcal{N}_i(t)$, position $\pi(i)$, bid $b_i(t)$ and sets $\mathcal{I}_i^a(t)$ and $\mathcal{I}_i^t(t)$.⁵ Updates in both navigation and coordination automata depend explicitly on nearest neighbors $\mathcal{N}_i(t)$, hence the distributed nature of the approach. The two automata are synchronized and together consist the model of agent i . The following notion of a *predicate* enables us to formally define the aforementioned automata.

Definition 4.1 (Predicate): Let $X = \{x_1, \dots, x_n\}$ be a finite set of variables. We define a predicate $\psi(X)$ over X to be a finite conjunction of strict or non-strict inequalities over X . We denote the set of all predicates over X by $\text{Pred}(X)$.

In other words, a predicate is a logical formula. For example, the predicate $\psi(X) = (\|x - x_0\| < r)$ over the set of variables $X \in \mathbb{R}^N$ returns 1 if x belongs in the open ball $\|x - x_0\| < r$ and 0 otherwise. Hence, the navigation automaton for agent i can be defined as follows.⁶

Definition 4.2 (Navigation Hybrid Automaton): We define the navigation hybrid automaton of agent i to be the tuple $N_i = (X_{N_i}, V_{N_i}, E_{N_i}, \Sigma_{N_i}, \text{sync}, \text{inv}, \text{init}, \text{guard}, \text{reset}, \text{flow})$, where,

- $X_{N_i} = \{x_i, y_i, \phi_i, \psi_i, \theta_i\}$ denotes the set of owned state variables with $x_i, y_i \in \mathbb{R}^p$ and $\phi_i, \psi_i, \theta_i \in [-\pi, \pi]$.
- $V_{N_i} = \{N\}$ denotes the finite set of control modes.⁷
- $E_{N_i} = \{(N, N)\}$ denotes the set of control switches.
- $\Sigma_{N_i} = \{\text{update}_i\}$ denotes the set of synchronization labels.
- $\text{sync} : E_{N_i} \rightarrow \Sigma_{N_i}$ with, $\text{sync}((N, N)) = \text{update}_i$, denotes the synchronization map mapping each control switch to a synchronization label.

⁵Technically, the model we propose for the agents does not correspond to an Input/Output hybrid automaton [26], but to a composition of synchronized automata. In our framework, the terms *input* and *output* are used for presentation purposes, exclusively.

⁶To simplify notation, we hereafter drop the dependence of the state variables on time t .

⁷The shorthand notation stands for $N := \text{Navigate}$.

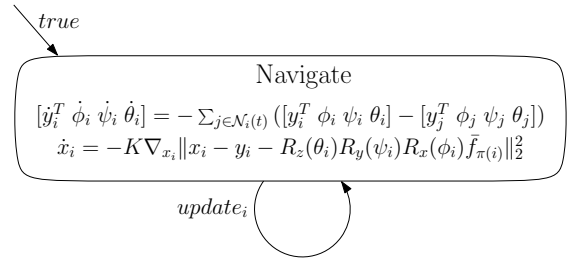


Fig. 3. Navigation Automaton for Agent i .

- $\text{inv} : V_{N_i} \rightarrow \text{Pred}(X_{N_i})$ with $\text{inv}(N) = \text{true}$, denotes the invariant conditions of the hybrid automaton.
- $\text{init} : V_{N_i} \rightarrow \text{Pred}(X_{N_i})$ with $\text{init}(N) = \text{true}$, denotes the set of initial conditions.
- $\text{guard} : E_{N_i} \rightarrow \text{Pred}(X_{N_i})$ with $\text{guard}((N, N)) = \text{true}$, denotes the set of guards of the hybrid automaton.
- $\text{reset} : E_{N_i} \rightarrow X_{N_i}$ with $[x_i^T y_i^T \phi_i \psi_i \theta_i] := \text{reset}((N, N)) = [x_i^T y_i^T \phi_i \psi_i \theta_i]$, denotes the set of resets associated with the guards of the hybrid automaton.
- $\text{flow} : V_{N_i} \rightarrow \dot{X}_{N_i}$ with $[\dot{x}_i^T \dot{y}_i^T \dot{\phi}_i \dot{\psi}_i \dot{\theta}_i] := \text{flow}(N)$ as in equations (4) – (6), denotes the flow conditions of the hybrid automaton that constrain the first time derivatives of the system variables in mode $v \in V_{N_i}$.

Figure 3 shows the graph representation of the hybrid automaton N_i . Observe that N_i consists of a single mode including the dynamics of agent i , while self transitions in N_i are synchronized with transitions of the coordination automaton of agent i , due to the synchronization labels $\text{sync}((N, N)) = \text{update}_i$. Such transitions are triggered whenever the position $\pi(i)$ or neighbor set \mathcal{N}_i of agent i are updated by the coordination automaton C_i , which consists an implementation of the auction mechanism in Algorithm 1 and can be defined in a similar way (Figure 4). Note that transitions from mode $v = A$ to modes $v' = N, U$ in automaton C_i are time triggered and take place after time $\tau_i > 0$ has elapsed. The reason for this concerns synchronization of the automata N_i and C_i . In particular, in the absence of the time $\tau_i > 0$, all transitions in C_i are instantaneous, which due to synchronization, implies that transitions in N_i are instantaneous too. But then, the flow in N_i is not well defined.⁸

V. INTEGRATION OF THE OVERALL SYSTEM

Composition of all elementary automata N_i and C_i leads to a *product system* S [25] that models the interconnection between them. Clearly, studying S we can identify the properties of the whole multi-agent system. The following proposition shows that every agent has always knowledge of at least all available destinations in $\mathcal{I}(t)$. This property of the product system S is necessary to show that eventually $\pi(i) \neq \pi(j)$ for all agents $i \neq j$, i.e., that π is eventually a

⁸One could also view $\tau_i > 0$ as a model for time delays due to communication or computation costs in the auction mechanism itself.

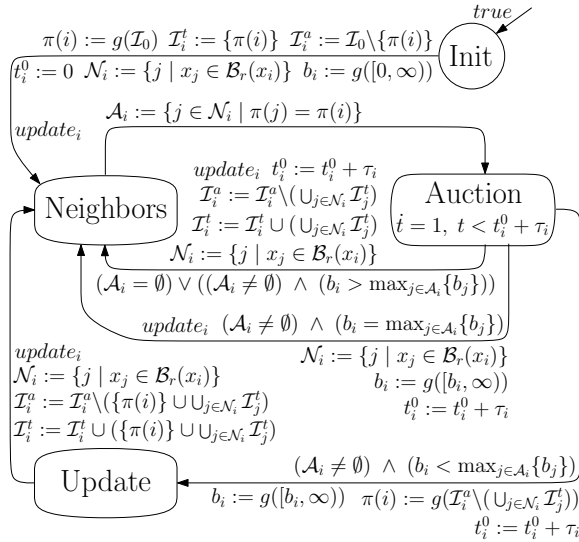


Fig. 4. Coordination Automaton for Agent i .

permutation.⁹

Proposition 5.1: The product system S guarantees that $\mathcal{I}(t) \subseteq \mathcal{I}_i^a(t)$ for all time t and all agents i .

Our next result concerns the running time of the hybrid system S . In particular, we show that the product system S in the worst case can only take a finite number of transitions $v_S \xrightarrow{e_S} v'_S$ such that $v_{C_i} = A$ and $v'_{C_i} = U$ for all agents i , which is polynomial with respect to the number of agents n . In other words, the permutation π can only be updated finitely many times. This result is important, given that the number of permutations π , and hence the space of control modes V_S of the product system S , grows exponentially with the number of agents.

Proposition 5.2: Let $v_S^* = (v_{N_1}^*, \dots, v_{N_n}^*, v_{C_1}^*, \dots, v_{C_n}^*)$ be such that $\pi(i) \neq \pi(j)$ for all $j \neq i$. Then, the product system S can reach v_S^* in at most $\frac{n(n-1)}{2}$ transitions $v_S \xrightarrow{e_S} v'_S$ such that $v_{C_i} = A$ and $v'_{C_i} = U$.

Having showed that the product system S has polynomial complexity, we now show that it also has the desired *liveness* property. In other words, we show that mode v_S^* is eventually reached for all initial conditions

Theorem 5.3: Assume that the conditions of Proposition 3.2 hold. Then, for all initial conditions $x_i(t_0)$, there exists a constant $T > 0$ such that for all time $t > t_0 + T$, the product system S is in mode $v_S^* = (v_{N_1}^*, \dots, v_{N_n}^*, v_{C_1}^*, \dots, v_{C_n}^*)$ with $\pi(i) \neq \pi(j)$ for all $j \neq i$. We call v_S^* the equilibrium mode of the system.

VI. SIMULATION RESULTS

We consider a navigation task in \mathbb{R}^2 , where $n = 231$ agents, starting from randomly chosen initial configurations, are required to reach a lattice formation with a given realization as in Figure 5. For all agents i , the translation vectors y_i are initialized with the agents' initial positions,

⁹Proofs for this and the other results in this section are omitted due to space limitations.

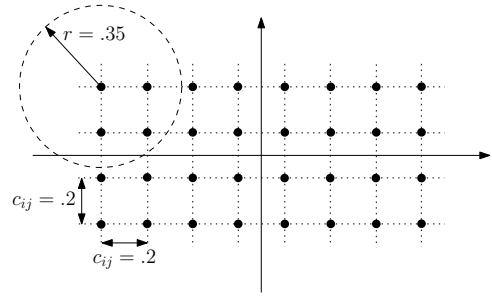


Fig. 5. Desired Lattice Formation.

i.e., $y_i(t_0) = x_i(t_0)$, while the angles ϕ_i , ψ_i and θ_i are randomly initialized in the interval $[0, 2\pi]$.¹⁰ Figure 6 shows the evolution of the system at 5 different time instants. The agents are denoted with dots and the communication range used is $r = .35$. Observe that the hybrid system S eventually drives the group of agents to the desired formation. Note also the scalability of our approach indicated by the number of agents that can be handled.

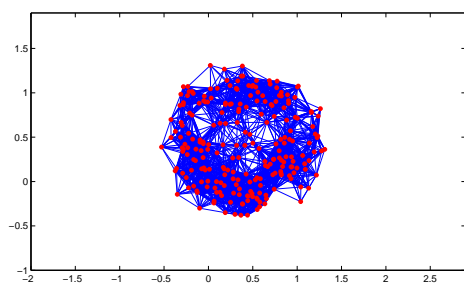
VII. CONCLUSIONS

In this paper, we considered a distributed hybrid approach to the formation control problem, that simultaneously addresses the discrete permutation of the agents in the formation as well as the continuous navigation strategies that drive the group to the desired formation. A formation was represented as a rotational and translational invariant configuration in the free space and distributed consensus algorithms guaranteed that all agents *agreed* on the rotation and translation of the final configuration of the formation. On the other hand, the permutation of the agents in the formation was determined dynamically, by means of distributed market-based coordination protocols, while navigation of the group to the desired formation was due to artificial potential fields. The overall hybrid system was shown to always converge to the desired formation as long as certain connectivity assumptions on the underlying communication network were satisfied and have at most polynomial complexity, despite the exponential growth of the number of permutations with respect to the number of agents. Our scalable approach was verified through non-trivial computer simulations.

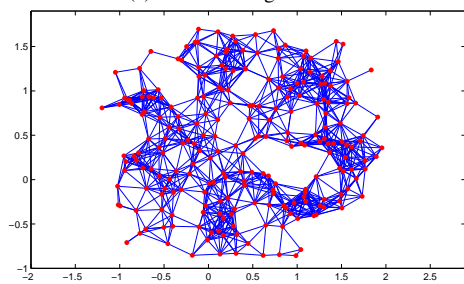
REFERENCES

- [1] P. Tabuada, G. J. Pappas and P. Lima. *Motion Feasibility of Multi-Agent Formations*, IEEE Transactions on Robotics, vol. 21(3), pp. 387-392, June 2003.
- [2] M. C. DeGennaro and A. Jadbabaie. *Formation Control for a Cooperative Multiagent System Using Decentralized Navigation Functions*, Proceedings of the American Control Conference, pp. 1346-1351, Minneapolis, MN, June 2006.
- [3] M. Ji and M. Egerstedt. *Distributed Formation Control While Preserving Connectedness*, Proceedings of the 45th IEEE Conference on Decision and Control, pp. 5962-5967, San Diego, CA, Dec. 2006.
- [4] J. A. Fax and R. M. Murray. *Information Flow and Cooperative Control of Vehicle Formations*, IEEE Transactions on Automatic Control, vol. 49(9), pp. 1465-1476, Sept. 2006.

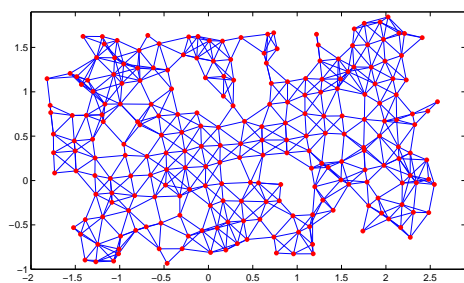
¹⁰Note that initialization of the translation vectors y_i aims at translating the formation “close” to the centroid of the initial configuration of the agents (see Theorem 3.1).



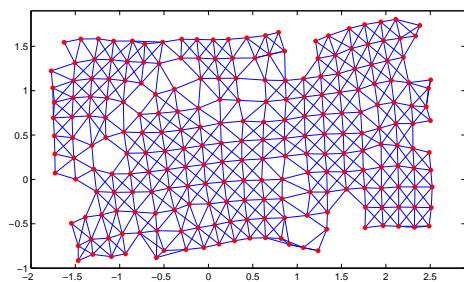
(a) Initial Configuration.



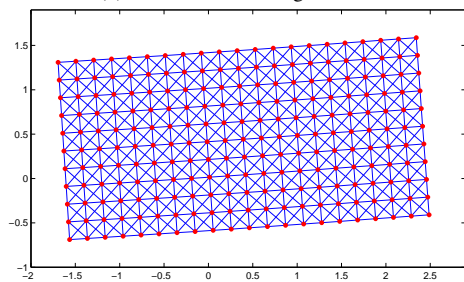
(b) Intermediate Configuration.



(c) Intermediate Configuration.



(d) Intermediate Configuration.



(e) Final Configuration.

Fig. 6. Distributed formation control for $n = 231$ agents.

- [5] J. P. Desai, J. P. Ostrowski and V. Kumar. *Modeling and Control of Formations of Nonholonomic Mobile Robots*, IEEE Transactions on Robotics and Automation, vol. 17(6), pp. 905-908, Dec. 2001.
- [6] J. P. Desai, V. Kumar and J. P. Ostrowski. *Control of Changes in Formation for a Team of Mobile Robots*, Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1556-1561, Detroit, MI, May 1999.
- [7] T. Balch and R. C. Arkin. *Behavior-based Formation Control for Multirobot Teams*, IEEE Transactions on Robotics and Automation, vol. 14(6), pp. 926-939, Dec. 1998.
- [8] N. E. Leonard and E. Fiorelli. *Virtual Leaders, Artificial Potentials and Coordinated Control of Groups*, Proceedings of the 40th IEEE Conference on Decision and Control, pp. 2968-2973, Orlando, FL, Dec. 2001.
- [9] M. Mesbahi and F. Y. Hadaegh. *Formation Flying Control of Multiple Spacecraft via Graphs, Matrix Inequalities and Switching*, AIAA Journal of Guidance, Control and Dynamics, vol. 24(2), pp. 369-377, 2001.
- [10] H. G. Tanner, G. J. Pappas and V. Kumar. *Leader-to-Formation Stability*, IEEE Transactions on Robotics and Automation, vol. 20(3), pp. 433-455, June 2004.
- [11] J. Lawton, B. Young and R. Beard. *A Decentralized Approach to Elementary Formation Maneuvers*, Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2728-2733, San Francisco, CA, April 2005.
- [12] C. Belta and V. Kumar. *Motion Generation for Formations of Robots: A Geometric Approach*, Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1245-1250, Seoul, Korea, May 2001.
- [13] M. M. Zavlanos and G. J. Pappas. *Dynamic Assignment in Distributed Motion Planning with Local Coordination*, IEEE Transactions on Robotics, November 2006. Accepted.
- [14] S. L. Smith and F. Bullo. *Target Assignment for Robotic Networks: Asymptotic Performance under Limited Communication*, Proceedings of the American Control Conference, pp. 1155-1160, New York, NY, July 2007.
- [15] M. M. Zavlanos and G. J. Pappas. *A Dynamical Systems Approach to Weighted Graph Matching*, Proceedings of the 45th IEEE Conference on Decision and Control, pp. 3492-3497, San Diego, CA, December 2006.
- [16] S. Kloder and S. Hutchinson. *Path Planning for Permutation-Invariant Multirobot Formations*, IEEE Transactions on Robotics, vol. 22(4), pp. 650 - 665, Aug. 2006.
- [17] M. Ji, S. Azuma, and M. Egerstedt. *Role-Assignment in Multi-Agent Coordination*, International Journal of Assistive Robotics and Mechatronics, vol. 7(1), pp. 32-40, March 2006.
- [18] S. Poduri and G. S. Sukhatme. *Constrained Coverage for Mobile Sensor Networks*, In IEEE International Conference on Robotics and Automation, pp. 165-172, New Orleans, LA, May 2004.
- [19] 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(8), pp. 1289-1298, Aug. 2006.
- [20] R. Olfati-Saber and R. M. Murray. *Consensus Problems in Networks of Agents with Switching Topology and Time-Delays*, IEEE Transactions on Automatic Control, vol. 49, pp. 1520-1533, Sep. 2004.
- [21] W. Ren and R. Beard. *Consensus of Information under Dynamically Changing Interaction Topologies*, Proceedings of the American Control Conference, pp. 4939-4944, Boston, MA, June 2004.
- [22] 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(6), pp. 988-1001, 2003.
- [23] H. Tanner, A. Jadbabaie and G. Pappas. *Flocking in Fixed and Switching Networks*, IEEE Transactions on Automatic Control, vol. 52(5), pp. 863-868, May 2007.
- [24] C. Godsil and G. Royle. *Algebraic Graph Theory*, Springer-Verlag, New York, 2001.
- [25] T. A. Henzinger. *The Theory of Hybrid Automata*, Proceedings of the 11th Annual Symposium on Logic in Computer Science (LICS), pp. 278-292, IEEE Computer Society Press, 1996.
- [26] N. A. Lynch, R. Segala and F. W. Vaandrager. *Hybrid I/O Automata*, Information and Computation, vol. 185(1), pp. 105-157, Aug. 2003.