Chapter 3
Interacting with Networks of Mobile Agents

Magnus Egerstedt, Jean-Pierre de la Croix, Hiroaki Kawashima, and Peter Kingston

Abstract How should human operators interact with teams of mobile agents, whose movements are dictated by decentralized and localized interaction laws? This chapter connects the structure of the underlying information exchange network to how easy or hard it is for human operators to influence the behavior of the team. "Influence" is understood both in terms of controllability, which is a point-to-point property, and manipulability, which is an instantaneous influence notion. These two notions both rely on the assumption that the user can exert control over select leader agents, and we contrast this with another approach whereby the agents are modeled as particles suspended in a fluid, which can be "stirred" by the operator. The theoretical developments are coupled with multirobot experiments and human user-studies to support the practical viability and feasibility of the proposed methods.

Keywords Multi-agent robotics · Networked control · Human–robot interactions

3.1 Introduction

As networked dynamical systems appear around us at an increasing rate, questions concerning how to manage and control such systems are becoming increasingly important (e.g., [6]). Examples include multiagent robotics, distributed sensor networks, interconnected manufacturing chains, and data networks. In this chapter, we

M. Egerstedt (✉) · J.-P. de la Croix · P. Kingston
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
e-mail: magnus@gatech.edu

J.-P. de la Croix
e-mail: jdelacroix@gatech.edu

P. Kingston
e-mail: kingston@gatech.edu

H. Kawashima
Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
e-mail: kawashima@i.kyoto-u.ac.jp

P. Benner et al. (eds.), *Large-Scale Networks in Engineering and Life Sciences*,
Modeling and Simulation in Science, Engineering and Technology,
DOI 10.1007/978-3-319-08437-4_3

investigate how to interact with teams of distributed, mobile agents, and we propose two different ways of making the team amenable to human control. These two different approaches can be though of as representing the *Lagrangian* and *Eulerian* paradigms. The Lagrangian approach corresponds to a focus on the movements of the individual agents, and control is exerted over select leader-nodes in the network. In contrast to this, the Eulerian vantage-point corresponds to viewing the agents as particles suspended in a fluid, and the description is given in terms of particle flows. The human operator can influence such systems by manipulating the flows directly, rather than the movements of individual agents.

The outline is as follows: In Sect. 3.2, the interaction models are defined through information exchange graphs (networks), and we discuss how to design controllers for achieving geometric objectives, such as rendezvous or formation control. Leader-based interactions are the main topic of Sect. 3.3, and we show how human control can be achieved through a direct interaction with leader agents. Notions such as controllability and manipulability are used to evaluate the effectiveness of these human–swarm interactions. These notions are further pursued in Sect. 3.4, where user studies are conducted that connect the theoretical developments with how easy or hard it is for human operators to actually control the multiagent team. In Sect. 3.5, a fluid-based approach to human–swarm interactions is introduced, and its interpretation within the Eulerian context is discussed and evaluated experimentally in Sect. 3.6.

3.2 Multiagent Networks

The main objective when designing control, communication, and coordination strategies for multiagent networks is to have a collection of agents achieve some global objective using only local rules [3, 17]. If we associate a state $x_i \in \mathbb{R}^d$, $i = 1, \ldots, N$, with each of the N agents in the team, the global objectives can typically be encoded in terms of costs or constraints on the states. Here d is the dimension of the state, and if the agents are planar, mobile robots, x_i could be the position of agent i, in which case $d = 2$.

Central to the notion of a distributed strategy is the fact that each agent only has access to a limited set of neighboring agent states, and the control decisions must be made solely based on this limited information. If we let N_i denote the set of agents that are available to agent i (this set may be time varying as the team evolves), and we assume that the evolution of the agent's state is directly under control in the sense that $\dot{x}_i = u_i$, then the design choice involves selecting appropriate interaction laws $f_{ij}(x_i, x_j)$ with

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

Note that more involved dynamics could be imagined, but they would inevitably make the analysis more involved.

3.2.1 The Graph Abstraction

As the set of neighboring agents is crucial when defining the interaction laws, it is natural to view the system as one defined over a graph $G = (V, E)$. Here $V = \{1, \ldots, N\}$ is the set of agents, and the edge set $E \subset V \times V$ encodes neighborhood information in the sense that $j \in N_i \Leftrightarrow (j, i) \in E$, that is, an edge points from agent j to agent i if information is flowing from agent j to agent i. We will assume that the edges are undirected, that is, $j \in N_i \Leftrightarrow i \in N_j$, which corresponds to agent i having access to agent j's state if and only if agent j has access to agent i's state.

This graph abstraction is useful in that one can ask questions pertaining to what information is needed to support various multiagent tasks, which translates into finding the appropriate, underlying network graphs. As an example, if the graph is disconnected, that is, there are nodes in-between which no paths exists (possibly over multiple nodes), then there is no way information can be made available that correlates the states of these two nodes. Disconnectedness is thus a topological obstruction to achieving certain multiagent objectives. Similarly, if the graph is complete, that is, all agents have immediate access to all other agents ($N_i \cup \{i\} = V \ \forall i = 1, \ldots, N$), then what we in essence have is a centralized rather than decentralized situation. As we will see in subsequent sections, there are tight couplings between the network topology and how easy it is to interact with the networks. However, these couplings only become meaningful in the context of particular interaction protocols and global task objectives. We will start with a canonical such objective, namely the consensus problem, whereby all agents should agree on a common state value.

3.2.2 Consensus

The consensus problem is arguably the most fundamental of the coordinated controls problems in that it asks the agents to agree, that is, make their state values converge to a common value. One way of achieving this is to let each agent move towards the centroid of its neighboring agents, that is, to let

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

which is known as the *consensus* equation [12, 17, 20, 25]. As long as the underlying graph remains connected (there is a path between any two agents in the network), this will indeed achieve consensus in the sense that $\|x_i - x_j\| \to 0$ for all i, j as $t \to \infty$. An example of this is shown in Fig. 1.

Now, if we assume that the agents' states are all scalars (without loss of generality), we can gather them together in the ensemble vector $x = [x_1, \ldots, x_N]^{\mathrm{T}}$ and write the ensemble-level dynamics associated with the consensus equation as

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

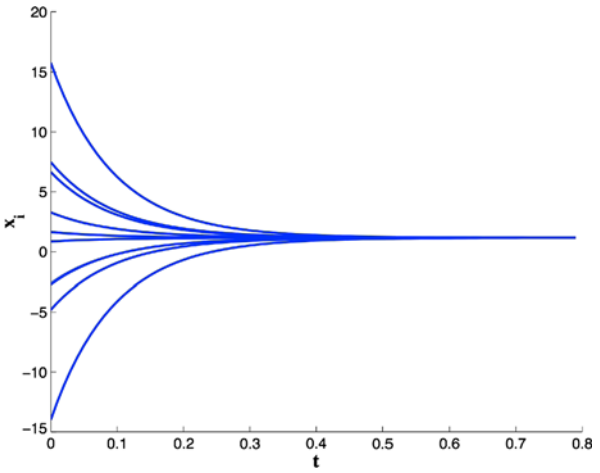


Fig. 1 Ten agents are executing the consensus equation. As a result, their state values converge to a common value

Here L is the graph *Laplacian* associated with the underlying network graph (e.g., [10]), and it is given by the difference between two other matrices associated with the graph,

$$L = D - A.$$

The matrix D is the *degree matrix*, which is a diagonal matrix

$$D = \text{diag}\big(\deg(1), \ldots, \deg(N)\big),$$

where the degree of node i ($\deg(i)$) is the cardinality of its neighborhood set N_i, that is, it captures how many neighbors that node has. The matrix A is the *adjacency matrix*, and it encodes the adjacency relationships in the graph in that $A = [a_{ij}]$, where

$$a_{ij} = \begin{cases} 1 & \text{if } j \in N_i, \\ 0 & \text{otherwise.} \end{cases}$$

The ensemble-level description of the node dynamics will prove instrumental for understanding how easy or hard it is to interact with such networks. However, before we can discuss this issue, some more should be said about how one can augment the consensus equation to solve more general problems, such as formation control problems.

3.2.3 Formations

The reason for the consensus equation's prominence is not necessarily in that it solves the consensus problem, but rather that it can be augmented to solve other types of problems. In fact, if we assume that agents i and j should end up at

a distance d_{ij} apart from each other, we can associate an edge tension energy $\mathscr{E}_{i,j}(\|x_i - x_j\|, d_{ij})$ to the edge between these two nodes, where this energy has been designed in such a way that $\mathscr{E}_{i,j} > 0$ as long as $\|x_i - x_j\| \neq d_{ij}$. If we do this for all edges in the network, we can then use the total energy $\mathscr{E}$ as a Lyapunov function to solve the "formation control problem" [13].

In fact, if we let

$$\dot{x}_i = -\sum_{j \in N_i} \frac{\partial \mathscr{E}_{i,j}}{\partial x_i},$$

then this simplifies to a weighted consensus equation

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

where $w_{i,j}$ is a scalar weight function. Following this construction for all agents results in a gradient descent with regards to the total energy in the network,

$$\frac{d\mathscr{E}}{dt} = -\left\| \frac{\partial \mathscr{E}}{\partial x} \right\|^2,$$

that is, the energy is nonincreasing in the network, and, using LaSalle's invariance principle, this fact can be used to show convergence to the desired shape (under reasonable choices of edge tension energies); see, for example, [13, 17–19]. An example of this is shown in Fig. 2.

This way of adding weights to the consensus equation has been used not only to solve formation control problems, but other geometric problems involving coverage control in sensor networks, boundary protection, and self-assembly problems in multirobot networks. Is has also been used extensively in biologically defined problems, such as swarming (How make the agents form a tight spatial shape?), flocking (How make the agents move in such a way that their headings align?), and schooling (How make the agents move as a general shape without colliding with each other?). For a representative sample, see [8, 12, 22, 23].

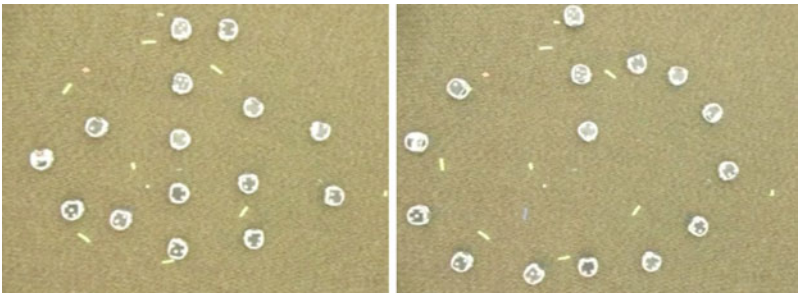
3.3 Leader-Based Interactions

Now that we have ways of describing the interagent interactions, we would like to insert human inputs into the network. In fact, we assume that a subset of the nodes $V_f \subset V$ (the so-called follower nodes) in the network evolve according to the consensus equation, whereas we inject control signals at the remaining nodes in $V_\ell \subset V$ (the leader nodes) through

$$\dot{x}_i = u_i, \quad i \in V_\ell,$$

or (which is equivalent from a controllability point of view)

$$x_i = u_i, \quad i \in V_\ell.$$



(a) Formation control ($t = 0$) (b) Formation control ($t = 5$)

Fig. 2 15 mobile robots are forming the letter "G" by executing a weighted version of the consensus equation

If we index the nodes in such a way that the last M nodes are the leader nodes and the first $N - M$ nodes are the followers, we can decompose L as

$$L = -\left[\begin{array}{c|c} A & B \\ \hline B^{\mathrm{T}} & \lambda \end{array}\right],$$

where $A = A^{\mathrm{T}}$ is $(N - M) \times (N - M)$, B is $(N - M) \times M$, and $\lambda = \lambda^{\mathrm{T}}$ is $M \times M$. The point behind this decomposition is that if we assume that the state values are scalars, that is, $x_i \in \mathbb{R}$, $i = 1, \ldots, N$, and gather the states from all follower nodes as $x = [x_1, \ldots, x_{N-M}]^{\mathrm{T}}$ and the leader nodes as $u = [x_{N-M+1}, \ldots, x_N]^{\mathrm{T}}$, then the dynamics of the controlled network can be written as

$$\dot{x} = Ax + Bu,$$

as was done in [21]. This is a linear-time invariant control system,[1] and the reason for this formulation is that we can now apply standard tools and techniques when trying to understand how easy or hard it is to interact with such systems.

3.3.1 Controllability

One interesting fact about this construction is that the followers tend to cluster together due to the cohesion provided by the consensus equation. This clustering effect can actually be exploited when analyzing the network's controllability properties. We thus start with a discussion of how such clusters emerge.

By a *partition* of the graph $G = (V, E)$ we understand a grouping of nodes into cells, that is, a map $\pi : V \to \{C_1, \ldots, C_K\}$, where we say that $\pi(i)$ denotes the *cell*

[1] Note that if the states were nonscalar, the analysis still holds even though one has to decompose the system dynamics along the different dimensions of the states.

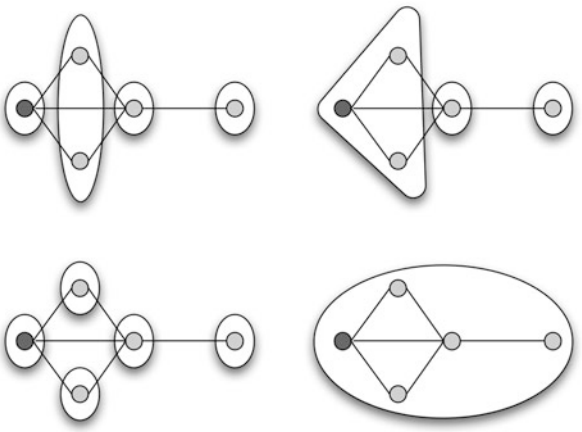


Fig. 3 A graph with four possible EEPs. The leader-node (*black node*) is in a singleton cell in the *two left-most figures*, and, as such, they correspond to leader-invariant EEPs. Of these two leader-invariant EEPs, the top-left partition has the fewest number of cells, and that partition is thus maximal. We note that this maximal partition is not trivial since one cell contains two nodes

that node i is mapped to, and we use $\text{range}(\pi)$ to denote the *codomain* to which π maps, that is, $\text{range}(\pi) = \{C_1, \ldots, C_K\}$. Similarly, the operation $\pi^{-1}(C_i) = \{j \in V \mid \pi(j) = C_i\}$ returns the set of nodes that are mapped to cell C_i.

But, we are not interested in arbitrary groupings. Instead, we partition the nodes into cells in such a way that all nodes inside a cell have the same number of neighbors in adjacent cells. To this end, the *node-to-cell degree* $\deg_\pi(i, C_j)$ characterizes the number of neighbors that node i has in cell C_j under the partition π,

$$\deg_\pi(i, C_j) = \left|\left\{k \in V \;\middle|\; \pi(k) = C_j \text{ and } (i,k) \in E\right\}\right|.$$

A partition π is said to be *equitable* if all nodes in a cell have the same node-to-cell degree to all cells, that is, if, for all $C_i, C_j \in \text{range}(\pi)$,

$$\deg_\pi(k, C_j) = \deg_\pi(\ell, C_j), \quad \text{for all } k, \ell \in \pi^{-1}(C_i).$$

This is almost the construction one needs in order to obtain a characterization of the controllability properties of the network. However, what we need to do is produce partitions that are equitable between cells in the sense that all agents in a given cell have the same number of neighbors in adjacent cells, but where we do not care about the structure *inside* the cells themselves. This leads to the notion of an *external equitable partition* (EEP) [6, 16], and we say that a partition π is an *EEP* if, for all $C_i, C_j \in \text{range}(\pi)$, where $i \neq j$,

$$\deg_\pi(k, C_j) = \deg_\pi(\ell, C_j), \quad \text{for all } k, \ell \in \pi^{-1}(C_i).$$

An example of this is given in Fig. 3.

3.3.1.1 A Necessary Controllability Condition for Single-Leader Networks

Assume that there is a single leader acting as the leader node, and we are particularly interested in EEPs that place this leader node in a singleton cell,

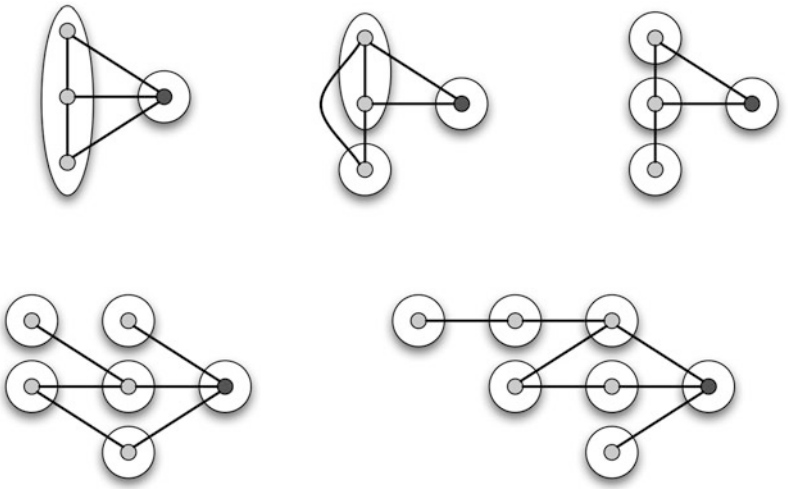


Fig. 4 *Clockwise from the top-left*: The first two networks are not completely controllable since their partitions π^* are not trivial. The partitions π^* associated with the remaining three networks are indeed trivial, but we cannot directly conclude anything definitive about their controllability properties since the topological condition is only necessary. Indeed, the third network is completely controllable, whereas the last two are not completely controllable

that is, in partitions where $\pi^{-1}(\pi(N)) = \{N\}$, and we refer to such EEPs as *leader-invariant*. Moreover, we say that a leader-invariant EEP is *maximal* if its codomain has the smallest cardinality, that is, if it contains the fewest possible cells, and we let $\pi^\star$ denote this maximal, leader-invariant EEP. Examples of the construction of $\pi^\star$ are shown in Fig. 3, and in [16] it was shown that the network is completely controllable only if G is connected and $\pi^\star$ is trivial, that is, $\pi^{\star -1}(\pi^\star(i)) = \{i\}$ for all $i \in V$, and examples of this topological condition for controllability are given in Fig. 4. What complete controllability means is that it is possible to drive the system from any configuration to any other configuration.

But, we can do even better than this in that we can characterize an upper bound on what the dimension of the controllable subspace is, as shown in [5]. In fact, let Γ be the controllability matrix associated with the controlled consensus equation. Then

$$\operatorname{rank}(\Gamma) \leq \left|\operatorname{range}\left(\pi^\star\right)\right| - 1.$$

We note that since this result is given in terms of an inequality instead of an equality, we have only necessary conditions for controllability rather than a, as of yet elusive, necessary and sufficient condition. One instantiation where this inequality is indeed an equality is when $\pi^\star$ is also a distance partition, as shown in [27]. What this means is that when all nodes that are at the same distance from the leader (counting hops through the graph) also occupy the same cell under $\pi^\star$, $\operatorname{rank}(\Gamma) = |\operatorname{range}(\pi^\star)| - 1$.

3.3.2 Manipulability

Controllability is ultimately a point-to-point property in that it dictates in-between what states it is possible to move the system. This is a rather strong condition, and one can also investigate a more localized notion of interactions, that is, one that describes what instantaneous changes to the system the control signal can achieve. To address the instantaneous effects that the inputs have on the team, we here discuss the notion of *manipulability* of leader-follower networks.

3.3.2.1 Manipulability of Leader–Follower Networks

In robotics, *manipulability* indices have been proposed as means for analyzing the singularity and efficiency of particular configurations and controls of robot-arm manipulators [1, 2, 26]. Let θ be the joint angles, and $r = f(\theta)$ be the state of the end-effector, where the function f represents the kinematic relation of the robot-arm manipulator. Then, a typical index of manipulability is defined in terms of the ratio of a measure of performance (end-effector response) $\dot{r}$ and a measure of effort (joint-angular velocity) $\dot{\theta}$ as

$$m_r = \frac{\dot{r}^{\mathrm{T}} W_r \dot{r}}{\dot{\theta}^{\mathrm{T}} W_\theta \dot{\theta}},$$

where $W_r = W_r^{\mathrm{T}}$ and $W_\theta = W_\theta^{\mathrm{T}} \succ 0$ are positive definite weight matrices. If f is differentiable, then we have the relation $\dot{r} = J_r(\theta)\dot{\theta}$ with $J_r(\theta)$ being the Jacobian matrix of the manipulator. Hence, the manipulability is given by the form of the Rayleigh performance-to-effort quotient [2, 26],

$$m_r = \frac{\dot{\theta}^{\mathrm{T}} J_r(\theta)^{\mathrm{T}} W_r J_r(\theta) \dot{\theta}}{\dot{\theta}^{\mathrm{T}} W_\theta \dot{\theta}}.$$

To establish a similar notion for leader-follower networks consisting of N_ℓ leaders and N_f followers with states $x_\ell = [x_{N_f+1}^{\mathrm{T}}, \ldots, x_N^{\mathrm{T}}]^{\mathrm{T}}$ and $x_f = [x_1^{\mathrm{T}}, \ldots, x_{N_f}^{\mathrm{T}}]^{\mathrm{T}}$, respectively (where we have assumed that the indexing is done such that the leader indices are last), one can simply define the manipulability index based on the ratio between the norm of the follower velocities and those of the leader velocities:

$$m(x, E, \dot{x}_\ell) = \frac{\dot{x}_f^{\mathrm{T}} Q_f \dot{x}_f}{\dot{x}_\ell^{\mathrm{T}} Q_e \dot{x}_\ell},$$

where $Q_f = Q_f^{\mathrm{T}} \succ 0$ and $Q_\ell = Q_\ell^{\mathrm{T}} \succ 0$ are positive definite weight matrices. Once this kind of indices is successfully defined under given agent configurations x and network topologies E, it can be used for estimating the most effective inputs to the

network by maximizing the manipulability m with respect to the input $\dot{x}_\ell$:

$$\dot{x}_{\ell,\max}(x, E) = \operatorname*{argmax}_{\dot{x}_\ell} m(x, E, \dot{x}_\ell),$$
$$m_{\max}(x, E) = \max_{\dot{x}_\ell} m(x, E, \dot{x}_\ell).$$

Another possible application is to use the manipulability index to find effective network topologies, given agent configuration x and leader inputs, $\dot{x}_\ell$, as

$$E_{\max}(x, \dot{x}_\ell) = \operatorname*{argmax}_{E} m(x, E, \dot{x}_\ell),$$

possibly under constraints on E (e.g., on the number of edges $|E|$).

For manipulability to be useful as a design tool, it needs to be connected to the underlying agent dynamics in a meaningful way, which presents some difficulty. Let us here, for example, consider the previously discussed agent dynamics for formation control. Specifically, the followers are trying to maintain given desired distances, whereas the leader agents are driven by exogenous inputs. As before, using the energy function $\mathscr{E}$, we let the control law of the followers be given by the weighted consensus equation

$$\dot{x}_f(t) = -\frac{\partial \mathscr{E}}{\partial x_f}^{\mathrm{T}}.$$

Under this dynamics, the followers try to "locally" decrease the total energy $\mathscr{E}$ through

$$\dot{\mathscr{E}} = \frac{\partial \mathscr{E}}{\partial x_f}\dot{x}_f + \frac{\partial \mathscr{E}}{\partial x_\ell}\dot{x}_\ell = -\left\| \frac{\partial \mathscr{E}}{\partial x_f} \right\|^2 + \frac{\partial \mathscr{E}}{\partial x_\ell}\dot{x}_\ell,$$

which ensures the desired behavior of the follower agents. (Note that $\mathscr{E}$ itself may increase because of the leaders' movement.)

In contrast to the manipulability of robot-arm manipulators, which can be analyzed through the kinematic relation, leader-follower network "links" are not rigid in the same way, and indeed we need to introduce an integral action to see the influence of $\dot{x}_\ell$. However, the input velocity $\dot{x}_\ell$ can vary over the time interval of integration. Thus, it is not possible to calculate an instantaneous performance-to-effort measure given by the definition of the manipulability m. For this reason, an approximate version of manipulability was introduced in [15] as a practically relevant manipulability proxy.

3.3.2.2 Approximate Manipulability

Let us consider the *rigid-link approximation* of the agent dynamics as an ideal situation, where all the given desired distances $\{d_{ij}\}_{(i,j)\in E}$ are perfectly maintained. Note that this approximation is reasonable if the scale of the edge-tension energy

$\mathscr{E}$ is large enough compared to that of the leader velocities $\dot{x}_\ell(t)$. Note also that, in real situations, $\mathscr{E}(t)$ needs to be greater than zero in order for the followers to move, whereas this approximation implies that $\mathscr{E}(t) = 0$ for all $t \geq 0$. Nevertheless, this approximation gives us a good estimate of the actual response of the network to inputs injected through the leader agents, unless the leaders move much faster than the followers.

To analyze the approximated dynamics, we need the notion of a rigidity matrix [7, 24]. If the connections between agent pairs associated with the edges can be viewed as rigid links, the distances between connected agents do not change over time. Assume that the trajectories of $x_i(t)$ are smooth and differentiable. Then

$$\frac{d}{dt}\|x_i - x_j\|^2 = 0 \quad \forall (i,j) \in E,$$

and therefore

$$(x_i - x_j)^{\mathrm{T}}(\dot{x}_i - \dot{x}_j) = 0 \quad \forall (i,j) \in E.$$

This set of constraints can be written in matrix form as

$$R(x)\begin{bmatrix}\dot{x}_f \\ \dot{x}_\ell\end{bmatrix} = \begin{bmatrix}R_f(x) \mid R_\ell(x)\end{bmatrix}\begin{bmatrix}\dot{x}_f \\ \dot{x}_\ell\end{bmatrix} = 0,$$

where $R(x) \in \mathbb{R}^{|E| \times Nd}$, $R_f(x) \in \mathbb{R}^{|E| \times N_f d}$, $R_\ell(x) \in \mathbb{R}^{|E| \times N_\ell d}$, and $|E|$ is the number of edges. The matrix $R(x)$ is known as the *rigidity matrix*. Specifically, considering that R consists of $|E| \times N$ blocks of $1 \times d$ row vectors, its (k, i_k) and (k, j_k) blocks are $(x_{i_k} - x_{j_k})^{\mathrm{T}}$ and $-(x_{i_k} - x_{j_k})^{\mathrm{T}}$, respectively (the signs can be swapped), and other blocks are zeros, where i_k and j_k are the agents connected by edge $k \in \{1, \ldots, |E|\}$.

Assume that the leaders move in a feasible manner so that the rigid-link approximation stays valid. Solving the constraint equation, the possible set of follower velocities $\dot{x}_f$ associated with $\dot{x}_\ell$ can be obtained as the following general solution:

$$\dot{x}_f = -R_f^\dagger R_\ell \dot{x}_\ell + \big[\mathrm{null}(R_f)\big]q,$$

where $R_f^\dagger$ is the Moore–Penrose pseudo inverse of R_f, q is an arbitrary vector whose dimensionality is nullity(R_f), and [null(R_f)] is a matrix whose columns span null(R_f). This means that there may exist infinite possibilities of $\dot{x}_f$ (i.e., rotational freedom and/or formation flexibility) for a given input $\dot{x}_\ell$. For instance, the rotational freedom around the leader always remains in a single-leader case. In such indeterminate cases, the manipulability index cannot be determined uniquely. And, one option is to modify the definition of manipulability, for example, by using the "worst-case approach" [1], namely, to analyze the impact of given inputs based on the least response (i.e., the smallest norm of the generated follower velocities, in our case). However, in [15] it was shown that $\dot{x}_f$ is uniquely determined as

$$\dot{x}_f = -R_f^\dagger R_\ell \dot{x}_\ell,$$

that is, $q = 0$ even in the indeterminate cases, once one considers the original agent dynamics $\dot{x}_f = \frac{\partial \mathscr{E}}{\partial x_f}^{\mathrm{T}}$ and then applies the rigid-link approximation. This is the key to the notion of approximate manipulability of formation-controlled leader-follower networks.

Using the fact that, under the rigid-link approximation, the followers' response is given by $\dot{x}_f = J\dot{x}_\ell$, where $J(x, E) = -R_f^\dagger R_\ell$, the approximate manipulability can be defined as the Rayleigh quotient

$$m(x, E, \dot{x}_\ell) = \frac{\dot{x}_\ell^{\mathrm{T}} J^{\mathrm{T}} Q_f J \dot{x}_\ell}{\dot{x}_\ell^{\mathrm{T}} Q_\ell \dot{x}_\ell},$$

which is similar to the robot-arm manipulability m_r. One can moreover see that J is analogous to the Jacobian matrix for robot-arm manipulators. Hence, in a manner similar to the robot-arm manipulability m_r, the maximum and minimum values of the manipulability index are determined by a spectral analysis. In other words, $m_{\max}$ is dictated by the maximum eigenvalue $\lambda_{\max}$ of the generalized eigenvalue problem $J^{\mathrm{T}} Q_f J v = \lambda Q_\ell v$, and $\dot{x}_{\ell,\max}$ is obtained from its corresponding eigenvector $v_{\max}$ as $\dot{x}_{\ell,\max} = \alpha v_{\max}$ ($\alpha \neq 0$). Similarly, the minimum value of the manipulability m and its corresponding inputs can be obtained from the minimum eigenvalue and its corresponding eigenvector, respectively.

As a final exercise, we use the notion of approximate manipulability of multiagent networks to describe effective input directions, in the case where Q_ℓ is the identity matrix. In fact, for the robot-arm manipulability with the identity weight matrices, that is, $\dot{r}^{\mathrm{T}}\dot{r}/(\dot{\theta}^{\mathrm{T}}\dot{\theta})$, the manipulability ellipsoid is defined as $\dot{r}^{\mathrm{T}}(J_r J_r^{\mathrm{T}})^\dagger \dot{r} = 1$; this ellipsoid depicts which direction the end-effector can be effectively moved by given inputs (joint-angular velocities) $\dot{\theta}$ with the same norm $\|\dot{\theta}\| = 1$. In contrast, since what we are interested in is the effective direction (axis) of inputs, the following *leader-side manipulability ellipsoid* can be used to characterize the effectiveness of injected inputs in the space of leader velocities:

$$\dot{x}_\ell^{\mathrm{T}} \left(J^{\mathrm{T}} Q_f J\right)^\dagger \dot{x}_\ell = \text{const.}$$

As such, the longest axis of the ellipsoid corresponds to the eigenvector that gives the maximum eigenvalue of $J^{\mathrm{T}} Q_f J$ and hence the most effective, instantaneous direction in which to interact with the network.

3.4 Leader–Follower User Studies

The discussions in the previous sections tell us what is possible in terms of network interactions. And, if the inputs are computationally generated, controllability and manipulability tell a rather comprehensive story. However, just because something is theoretically possible, it does not follow that it is easy to do. As such, user studies are needed to see if the developed human–swarm interaction theories line up with

Table 1 Network configuration, leader location, and target configuration for each task

Tasks	Network	Leader	Notation	Targets
1, 8	L_7	Head	$L_{7,h}$	Ellipse, Wedge
2, 9	L_7	Offset	$L_{7,o}$	Ellipse, Wedge
3, 10	L_7	Center	$L_{7,c}$	Ellipse, Wedge
4, 11	C_7	Any	C_7	Ellipse, Wedge
5, 12	K_7	Any	K_7	Ellipse, Wedge
6, 13	S_7	Center	$S_{7,c}$	Ellipse, Wedge
7, 14	S_7	Periphery	$S_{7,p}$	Ellipse, Wedge

user experiences when interacting with networks of mobile agents. In particular, we wish to understand what properties of a network make it easy or hard for a human to reasonably interact with it. To answer this question, participants were tasked with controlling different networks and to rate the difficulty of interacting with these networks (see [4]).

3.4.1 Experimental Results

The experiments were organized in such a way that 18 participants rated the difficulty of forming two different geometries with a network of seven agents organized according to one of four topologies. Table 1 provides a list of the 14 tasks performed in random order by each participant.

The leader-based interaction topology is defined by the second and third columns. We selected a representative set of canonical topologies: the line graph L_N, the cycle graph C_N, the complete graph K_N, and the star graph S_N. The agents in an L_N graph are organized like points on a line, where each agent is connected to two immediate neighboring agents. We appoint three different agents as a possible leader of an L_N graph: an agent at the head of line, an agent behind the head of the line, and an agent in the center of the line. The C_N graph can be formed from an L_N graph by forming an edge between the head and tail agents of the line. If all agents in the network share an edge with all other agents, then this topology is referred to as the K_N graph. If all agents in the network share a single edge with a common agent, then this topology is referred to as the S_N graph. We appoint two agents as a possible leader of an S_N graph: the central agent and a peripheral agent. The fourth table column defines the notation that we used to define a particular single-leader network topology.

Each of the 14 tasks requires the participants to move the network from an initial geometry (sufficiently different from the geometry of the target formation) to one of two target geometries listed in the fifth table column. A participant is briefly shown the interaction topology of the network *before* starting the task. Once the task is started, the interaction topology, like wireless links, is not visually observable by the participant, and the participant has to infer the interactions over the network from the motion of the agents. The participant is able to directly control the motion

of the leader agent using a joystick to achieve the target geometry with the network. A translation, rotation, and assignment invariant least squares fit (see [14]) is used to measure a participant's performance. This score is not shown at any time to the participant to ensure that the participant is simply focused on completing the task and rating its difficult. The participant rates the difficulty of each task on a continuous numeric scale from 0.0 (very easy) to 20.0 (very hard). In addition, we asked each participant to complete the NASA Task Load Index (TLX) workload survey (see [11]), which consists of six questions that cover physical, mental, and temporal demands, as well as a self-evaluation of performance, effort, and frustration.

The ratings provided by participants, the LSQ fit errors, and the total raw TLX scores for each tasks were analyzed and visualized as histograms in Fig. 5. The mean is denoted by the height of the bar, and the standard error is denoted by the error bars.

However, to make any sort of comparisons between tasks from this data, we apply a series of one-way ANOVA statistical tests (see [9]). These tests reveal that the LSQ fit error ($p < 0.0000001$), ratings ($p = 0.00138$), and workload scores ($p = 0.0256$) are all statistically significant at a 95 % confidence level, meaning that one can distinguish between the different tasks given the three measures. Second, we use the one-way ANOVA statistical test again to compare tasks within the three measures. If, for example, this test revealed that there is a statistically significant difference between tasks 1 and 2 with respect to the rating score, then we are justified in claiming that the topology in task 1 is rated as easier or harder than the topology in task 2.

Each of the three measures—LSQ fit error, rating, and workload scores—demonstrates a similar trend. First, the task of forming an ellipse is generally easier than forming a wedge independent of network topology. Second, line graphs are mostly the easiest to control regardless of the target geometry. We have to be careful and use the modifier *mostly* here because not all pairwise comparisons yield statistically significant differences. Specifically, for those measures with a higher p-value, the difference between any two tasks is going to be less significant. However, almost without exception L_7 networks have a statistically significant lower (better) score than C_7, K_7, and S_7 topologies regardless of target formation. Similarly, S_7 topologies have in almost all cases a statistically significant higher (worse) score than all other topologies. It is not surprising that some network topologies were significantly more difficult to control than others. However, to make these types of observations stand on a more firm mathematical footing, we need to tie the results of the user study to controllability and other system and graph theoretic properties of networks with multiple agents.

3.4.2 Connecting Back to the Network

After the results of the user study are gathered, it is interesting to connect these back to interaction notions previously defined, such as network controllability. The reason for this is that we would like to know whether or not these theoretical properties also correspond to practically useful notions human operators are to interact with networks of mobile agents.

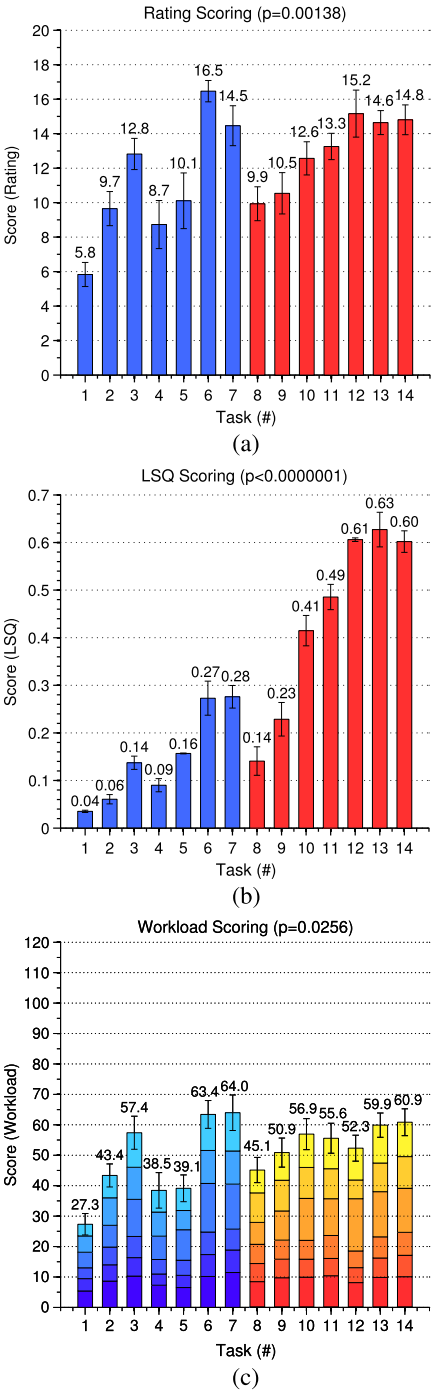


Fig. 5 Mean (**a**) LSQ, (**b**) rating, and (**c**) workload scores for each task

3.4.2.1 Controllability

A rank-deficient controllability matrix associated with the controlled consensus equation implies that there are certain things that the human operator simply cannot do. Therefore, the rank of the controllability matrix ought to be a good indicator of whether a network is easy or hard to control.

Since we are not only interested in whether a network is controllable or not, but also *how* controllable it is, we need to look at properties of the network beyond the rank of the controllability matrix. Therefore, we will use degree centrality, closeness, betweenness, and eigenvector centrality to try to quantify the importance of the leader v_ℓ. Degree centrality is defined by

$$C_D(v_\ell) = \deg(v_\ell), \quad \text{where } v_\ell \in V,$$

which only measures the importance of the leader by the size of its neighborhood set. Closeness on the other hand is defined by the length of the shortest paths from the leader to all other nodes on the network:

$$C_C(v_\ell) = \sum_{v \in V \setminus v_\ell} 2^{-\text{dist}(v_\ell, v)}, \quad \text{where } v, v_\ell \in V.$$

Betweenness measures the ratio of the number of shortest paths between any two agents that passes through the leader agent:

$$C_B(v_\ell) = \sum_{v \neq w \in V \setminus v_\ell} \frac{\sigma_{v,w}(v_\ell)}{\sigma_{v,w}},$$

where $\sigma_{v,w}(v_\ell)$ is the total number of shortest paths between v and w that intersect the leader, and $\sigma_{v,w}$ is the total number of shortest paths between v and w. Last, eigenvector centrality measures the influence of a node on the network, which can be computed by solving the eigenvalue problem $A\mathbf{y} = \lambda_{\max}\mathbf{y}$, where A is the adjacency matrix, and $\lambda_{\max}$ is its largest eigenvalue. Assuming that the leader is node N, the Nth entry of the vector $\mathbf{y}$ is the centrality score given to the leader:

$$C_E(v_\ell) = y_N, \quad \text{where } y_N \text{ is the } N\text{th entry of } \mathbf{y}.$$

Since the leader agent is the point of interaction for the human operator in these leader-based networks, we expect that the node centrality of the leader is an indicator of how easy or hard a network is to control.

3.4.3 Correlation to the User Study

Table 2 summarizes the results of connecting the candidate measure to the results of the user study.

What we want to know is how the rank of the controllability matrix and the node centrality measures correlate to the LSQ error fit, ratings, and workload scores

Table 2 Mean LSQ, rating, and workload scores with controllability matrix rank, ρ, and node centrality measures for each task

Task	Network	Target	ρ	C_D	C_C	C_B	C_E	LSQ	Rating	Workload
1	$L_{7,h}$	Ellipse	6	1	0.984	0	0.191	0.035	5.83	27.33
2	$L_{7,o}$	Ellipse	6	2	1.469	10	0.354	0.061	9.65	43.37
3	$L_{7,c}$	Ellipse	3	2	1.750	18	0.500	0.137	12.82	57.40
4	C_7	Ellipse	3	2	1.750	6	0.378	0.090	8.72	38.46
5	K_7	Ellipse	1	6	3.000	0	0.378	0.157	10.11	39.14
6	$S_{7,c}$	Ellipse	1	6	3.000	30	0.707	0.273	16.47	63.42
7	$S_{7,p}$	Ellipse	2	1	1.750	0	0.289	0.276	14.46	63.98
8	$L_{7,h}$	Wedge	6	1	0.984	0	0.191	0.141	9.93	45.14
9	$L_{7,o}$	Wedge	6	2	1.469	10	0.354	0.229	10.54	50.88
10	$L_{7,c}$	Wedge	3	2	1.750	18	0.500	0.415	12.57	56.94
11	C_7	Wedge	3	2	1.750	6	0.378	0.486	13.26	55.59
12	K_7	Wedge	1	6	3.000	0	0.378	0.606	15.16	52.32
13	$S_{7,c}$	Wedge	1	6	3.000	30	0.707	0.627	14.64	59.90
14	$S_{7,p}$	Wedge	2	1	1.750	0	0.289	0.602	14.81	60.86

collected from the user study. First, we observe that the rank of the controllability matrix is *negatively* correlated ($r^2_{\text{LSQ}} = -0.60,\ r^2_{\text{Rating}} = -0.73,\ r^2_{\text{Workload}} = -0.54$) to the scores. This correlation implies that *a configuration with a higher rank was almost without exceptions given a better score than a configuration with a lower rank*. We conclude that the rank of the controllability matrix is a strong predictor of how easy it is to control a network of multiple agents. As a corollary, it is not surprising that networks with a rank-deficient controllability matrix are more difficult to control because the human operator is likely to move the network into an uncontrollable subspace from which the task cannot be completed.

Second, the node centrality measures of the leader are *positively* correlated (e.g., for C_E, $r^2_{\text{Rating}} = 0.58,\ r^2_{\text{Workload}} = 0.54$) to the scores. This correlation implies that *given two configurations with the same ranks, C_D, C_B, C_C, and C_E all serve as reasonable tie breakers for which network is easiest to control*. In other words, given two networks with equally ranked controllability matrices, the network with the least central leader is likely to be the easiest to control by a human operator. It is important to note, however, that rank and node centrality are by no means *absolute* measures of the difficulty of controlling a given network, but good predictors of the difficulty for human operators to control these networks of multiple agents.

3.5 A Fluid-Based Approach

If the interactions are not based on influencing the behaviors of select agents, then one first has to understand by which means the interactions are physically supported.

For instance, one can envision scenarios where boundary control is exerted at some part of the swarm or where general flows (or other types of behavioral modifications) are imposed on the swarm as a whole. But, both of these types of interactions either require a broadcast to the entire swarm, which is not scalable as the swarm size scales up, or the information is injected at select nodes and then propagated through the network, which is inherently just a small variation to the leader-based interaction paradigm.

So what can one do about this? It is clear that the interactions have to have a physical manifestation, and one possible way forward is to take advantage of the fact that many mobile multiagent systems are in fact interacting over a fixed communications infrastructure. Examples include wireless LAN (802.11) routers, cellular networks (e.g., GSM), or air traffic control mechanisms (ATCT, TRACON, ARTCC). Common to these physical infrastructure networks is that they themselves are static, whereas the mobile agents are routed around in-between "cells." So one possible way of injecting information might be to interface directly with the nodes in the infrastructure network and have those nodes then interact with the agents that they are currently influencing.

3.5.1 The Infrastructure Network

Without committing to any particular interpretation of the state of an infrastructure node, assume that the state $p_i \in \mathbb{R}$ is associated with node i, $i = 1, \ldots, N$. These nodes will be interacting with the mobile agents. But they will also be interacting among themselves. Following the developments in previous sections, assume that the nodes are interacting through a controlled linear consensus equation

$$\dot{p}_i = -\sum_{j \in N_i} (p_i - p_j) + u_i,$$

where N_i is the set of nodes adjacent to node i. This can, as before, be written on ensemble form as

$$\dot{p} = -Lp + u,$$

where $p = (p_1, \ldots, p_N)^{\mathrm{T}}$ and $u = (u_1, \ldots, u_N)^{\mathrm{T}}$, and where L is the graph Laplacian associated with the infrastructure network. What we will do in subsequent sections is understand just what the correct interpretation of the node state p is as well as the corresponding interpretation of the control input u.

3.5.2 A Least-Squares Problem

If we associate an arbitrary orientation to the edges in the infrastructure network, we can factor the Laplacian as

$$L = DD^{\mathrm{T}},$$

where D is the incidence matrix, with $d_{ij} = 1$ if node i is the head to edge j, $d_{ij} = -1$ if it is the tail, and $d_{ij} = 0$ if node i is not incident to edge j. The important aspect of this factoring is that L is a Gramian. And Gramians have interpretations.

Consider for a moment the standard problem of finding a solution x to the problem $Ax = b$. If there is no such solution, the next best thing is the least-squares problem

$$\min_x \|Ax - b\|^2,$$

and the derivative of this cost is $1/2(A^{\mathrm{T}}Ax - A^{\mathrm{T}}b)$. Setting the derivative equal to zero yields the normal equation

$$AA^{\mathrm{T}}x = A^{\mathrm{T}}b,$$

where we have the Gramian AA^{T} play a central role.

In light of this discussion, we can reverse engineer a least-squares problem where the graph Laplacian takes on the role of AA^{T}. In other words, the corresponding least-squares problem is

$$\min_p \left\| D^{\mathrm{T}}p - f \right\|^2,$$

which in turn tries to find a solution p to $D^{\mathrm{T}}p = f$.

If we iteratively try to solve this problem, using a gradient descent strategy, we get

$$\dot{p} = -DD^{\mathrm{T}}p + Df$$

or, put another way,

$$\dot{p} = -Lp + Df.$$

This dynamical system is both decentralized and converges asymptotically to a solution to the normal equation $Lp = Df$.

But, the real benefit behind this detour to a least-square problem is that we now see what u really "is" in the controlled consensus equation, that is, we now know that

$$u = Df.$$

It remains to interpret this in a way that makes sense and use this interpretation as a basis for human–swarm interactions.

3.5.3 A Fluid-Based Interpretation

We directly note from the equation $D^{\mathrm{T}}p = f$ that p is simply assigning a number to each node in the network. Similarly, f assigns a number to each edge, whereas

D^{T} computes differences between nodes across edges. Using a continuous analog, p acts like a scalar field, f acts like a vector field, and D^{T} acts like a gradient. With this interpretation in mind, we see that the choice of letters p and f was not arbitrary. Instead, we can think of p as pressure and f as flow.

This interpretation gives us the means to interact with the infrastructure network directly. By specifying what we would like the flow to be in a particular cell around a given node, we in essence specify f. As we will see in subsequent sections, this specification will be done by moving a physical wand through cell boundaries, and the direction and magnitude of that movement will dictate the corresponding desired flow.

Once a flow vector has been established, the nodes update their individual pressure values using the decentralized controlled consensus equation, which on node-level form becomes

$$\dot{p}_i = \sum_{j \in N_i} \big(-(p_i - p_j) + \sigma_{ij} f_{ij}\big),$$

where σ_{ij} is the orientation of the edge between nodes i and j, and f_{ij} is the specified flow in-between those nodes.

3.6 Eulerian Swarms

In order to use the fluid-based interpretation of how one can interact with swarms of mobile agents, we first have to change the way we view said swarms. Since the leader-based interaction model is based on controlling individual agents, and the control design is done by focusing on the individual agents directly, we can call this the *Lagrangian* approach to swarm-interactions. The reason for this terminology is that Lagrangian fluid mechanics takes the point of view that the motions of individual particles in the fluid should be characterized. The alternative view, the *Eulerian* approach to fluid mechanics, instead focuses on particular spatial locations and models how the fluid passes through those locations. And, using the idea of a fixed infrastructure network, with spatial cells associated with the nodes in the infrastructure network through which the agents pass, we thus arrive at an Eulerian approach to multiagent swarms rather than the standard, Lagrangian approach.

3.6.1 From Lagrange to Euler

Given a static infrastructure network $G_I = (V_I, E_I)$, one way of thinking about the nodes is as zero-dimensional objects, or 0-simplexes. Similarly, an edge is a 1-simplex. This notion can of course be extended to surfaces, and we let a 2-simplex be given by "triangles" in the network $(i, j, k) \in V_I \times V_I \times V_I$ in the sense that $(i, j) \in E_I$, $(j, k) \in E_I$, and $(k, i) \in E_I$. These triangles (or rather, their spatial

footprint) constitute the spatial locations needed for the Eulerian view of multiagent swarms.

At any given time, inside each such triangle, we have a certain amount of agents. And, through the fluid-based equation $\dot{p} = -Lp + Df$, we also have a pressure associated with the triangles. By computing differences in pressure across boundaries in the triangles (through $D^{\mathrm{T}} p$), we thus get the desired flow of agents across those boundaries. So, if we somehow could turn those desired flows into control laws for the individual agents (back to a Lagrangian view again), then we would have come full circle and would be able to specify desired flows in the infrastructure network and then translate those flows into control laws for the individual agents, which is the topic of the next section.

3.6.2 Local Stream Functions

Stream functions are used in fluid dynamics to define two-dimensional flows, which is exactly what we have in this situation. In particular, the difference between the stream function at different points gives the flow through a line connecting those points. As the infrastructure agents are really regions, we will endow these regions with a dynamics in the sense that the mobile agents in that region will move according to that dynamics. Assuming that the regions are triangular, on an individual triangle (or 2-simplex), we can let the nodes that define the vertices of the triangle be given by x_1, x_2, x_3. The local, so-called stream function on this 2-simplex is given by

$$\phi(x) = c^{\mathrm{T}}(B_1 x + B_2),$$

where $c \in \mathbb{R}^3$ for some choice of c (to be specified later), and $B_1 \in \mathbb{R}^{3\times 2}$ and $B_2 \in \mathbb{R}^{3\times 1}$ satisfy

$$\begin{bmatrix} X \\ \mathbf{1}^{\mathrm{T}} \end{bmatrix}^{-1} = [B_1, B_2],$$

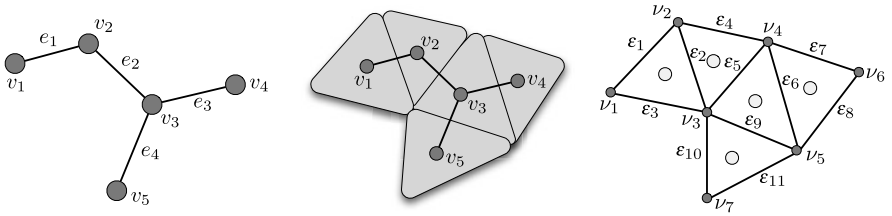
where $X = [x_1, x_2, x_3]$. The corresponding Hamiltonian, divergence-free dynamics, that is, the dynamics that an agent located at point x on the triangle should execute, is given by

$$\dot{x} = J \operatorname{grad} \phi(x) = J B_1^{\mathrm{T}} c,$$

with J being the $\pi/2$ rotation matrix

$$J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

What this means is that the flow inside a given triangle is constant, that is, it does not matter where inside the triangle an agent is. Moreover, all the agent needs to do is contact the infrastructure node inside the region to access that region's flow.



(a) (b) (c)

Fig. 6 An infrastructure network K (**a**), its triangular footprint (**b**), and the corresponding new network K

Philosophically speaking, the stream functions will be derived from the applied, user-specified flows, and they will be stitched together across the different triangles in order to obtain a global, piecewise linear stream function that will be used to dictate the motion of the individual agents. Since, for an individual region, $c \in \mathbb{R}^3$ is associated with the vertices in the region, we just need to map the input flow f associated with flows in-between regions to the nodes that make up the region. If we let G denote the infrastructure graph, then the new graph that we obtain by identifying edges in the triangles with edges in the new graph, and vertices with its vertices, we get a new graph K that has more edges than the original graph G since boundary edges are included as well. Letting L_K and D_K be the Laplacian and incidence matrices associated with the new graph, we (again) have to solve the least-squares problem

$$\dot{c} = -L_K c + D_K \hat{f},$$

where the old input flow f has been augmented to $\hat{f}$ to incorporate the new boundary edges that are present in K. For those edges, we set the flow equal to zero in order to not have agents leave the region.

As an example of this, consider the infrastructure network given in Fig. 6(a), with vertex set $\{v_1, \ldots, v_5\}$ and edge set $\{e_1, \ldots, e_4\}$. Given an arbitrary orientation of the edges, the corresponding matrices are

$$D = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}.$$

The association of triangular regions to the different infrastructure nodes are shown in Fig. 6(b), and the new graph K with vertex set $\{\nu_1, \ldots, \nu_7\}$ and edge set $\{\varepsilon_1, \ldots, \varepsilon_{11}\}$. We see that some of the edges in K are indeed corresponding to edges in G. In particular, we have the following correspondences: $e_1 \sim \varepsilon_2$, $e_2 \sim \varepsilon_5$, $e_3 \sim \varepsilon_6$, $e_4 \sim \varepsilon_9$. If the original input flow is specified through $f = [f_1, \ldots, f_4]^{\mathrm{T}}$, then we have the corresponding input flow $\hat{f}$ for the K graph given by $f_1 = \hat{f}_2$, $f_2 =$

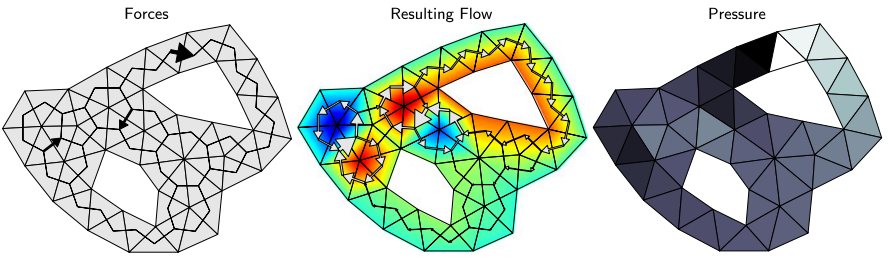


Fig. 7 Computational results are shown. Given a force field as input (*left*; arrow sizes indicate force magnitudes), a flow on the infrastructure graph, and a stream function over the environment are produced (*center*). The "pressure" computed as an intermediate step is also shown (*right*)

$\hat{f}_5$, $f_3 = \hat{f}_6$, $f_4 = \hat{f}_9$. The remaining edges in K $(\varepsilon_1, \varepsilon_3, \varepsilon_4, \varepsilon_7, \varepsilon_8, \varepsilon_{10}, \varepsilon_{11})$ are the boundary edges, and the corresponding $\hat{f}$-values are all 0, that is, $\hat{f}_1 = \hat{f}_3 = \hat{f}_4 = \hat{f}_7 = \hat{f}_8 = \hat{f}_{10} = \hat{f}_{11} = 0$.

Examples are helpful to demonstrate the qualitative characteristics of the flows obtained using the proposed interaction method. Figure 7 shows a typical solution. In that figure, a large force (desired flow) is exerted across a single face at the upper right of the complex, and this is propagated through the "jughandle" at the upper right. By contrast, the forces exerted lower in the complex, in less confined areas, result in pairs of vortices that have mostly local effects. Nevertheless, even in this case, small flows are produced throughout the complex. These qualitative characteristics are typical of the kinds of flows obtained; where necessary, flows propagate globally, but otherwise most effects are manifested locally.

It is the pressure field that propagates this information. Essentially, "shocks" are created across the faces where large forces are exerted, and elsewhere the pressure is smoothed throughout the environment by diffusion. The force exerted at the upper right demonstrates this well; it creates a "shock" in the pressure field (black triangle next to white triangle), which is spread by diffusion into linearly decreasing pressure around the upper right "jughandle." Where vortices are produced, the stream function exhibits a pair of local extrema, a maximum for a clockwise vortex and a minimum for a counterclockwise one, as can be observed in the left part of the complex. Vehicles then follow level sets of the stream function around the environment.

3.6.3 Conducting Swarms

A key goal of human–swarm interaction methods is to present human operators with high-level aggregate properties of swarms that they can manipulate, rather than requiring that they take on the cognitive workload of managing large numbers of agents individually. The fluid-based approach described in the previous sections gives an attractive way to do this by using "flows" of the agents as the aggregate

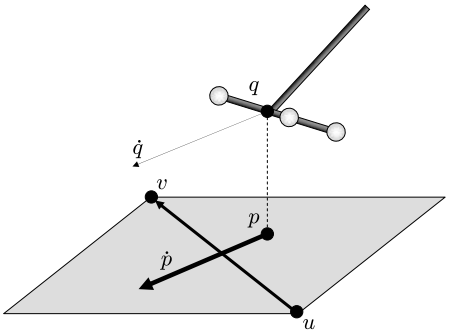


Fig. 8 Whenever the projection of the motion capture wand's center onto the floor plane crosses an edge between two triangles, a force in the direction of motion is superimposed across that edge. If the center of the motion-capture wand is q and its projection onto the ground plane is p, then a signed flow is superimposed across that edge of value $\tilde{f} = \dot{p}^{\mathrm{T}} J(v-u)/(\|v-u\|)$. Here, J is the $\pi/2$ rotation matrix used to define the stream function. Geometrically, $\tilde{f}$ is the component of the wand's projected velocity that is orthogonal to the edge

properties and by presenting humans with a physically inspired means of "pushing" and "pulling" on those flows.

In the context of the Eulerian approach to multiagent networks, what we are now concerned with is how to produce the vector f of "external forces" from human input that describes the "pushing" and "pulling." Our goal is to provide the human with a simple, intuitive interface that she can use to manipulate the swarm.

The implementation demonstrates how this can work, using motion capture as the user interface. The human makes physical motions that are tracked, and forces are generated on the fluid as she moves through it. Specifically, the human moves a wand with reflective markers that are tracked by cameras, and, as the wand crosses over edges between triangles, flows are created over them, as illustrated by Fig. 8.

There are a variety of options for how precisely to evolve the force vector f. In the implementation shown in Fig. 9, the force vector f is evolved by adding flows according to Fig. 8, and otherwise letting the forces decay according to first-order, linear dynamics. This means that if at times $t_1, t_2, \ldots$, edges indexed $i_1, i_2, \ldots$ are crossed, and flow increments $\tilde{f}_1, \tilde{f}_2, \ldots$ are calculated according to Fig. 8, then f is evolved as

$$\dot{f} = -\gamma f + \sum_{k=1}^{\infty} \tilde{f}_k \delta(t - t_k) e_{i_k},$$

where $\gamma \geq 0$ is a choice of decay rate; if there are m edges, e_i is the ith element of the $m \times m$ identity matrix; and δ is the Dirac delta distribution. This is one representative example of how motions can be mapped to (time-varying) force vectors and happens to be the one used in the implementation shown in Fig. 9.

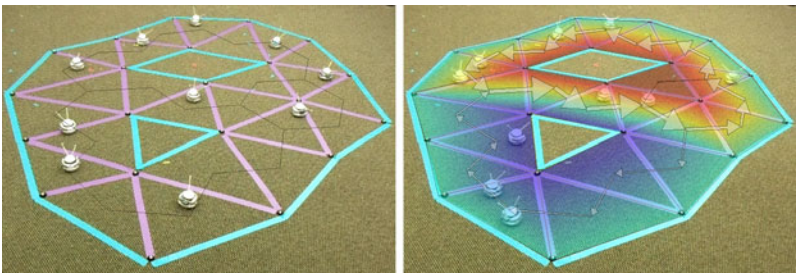


Fig. 9 Khepera III mobile robots in a simplicial complex (*left*) (internal edges are shown in purple and boundary edges in *blue*) and robots moving in the same complex according to a stream function, overlaid (*right*)

3.7 Conclusions

This chapter discusses a number of different ways in which human users can interact with networks of mobile agents. In particular, a Lagrangian approach is presented, where the user takes active control of a select number of leader nodes. Within this context, controllability and the instantaneous notion of manipulability are introduced. User studies where furthermore conducted that connected controllability and centrality notions to the ease by which human operators could interact with the network.

The other approach presented in this chapter is an Eulerian approach. This is characterized by the fact that the user no longer controls individual agents. Instead, the agents are assumed to be suspended in a fluid, and the user "stirs" this fluid by injecting desired flows across edges in the underlying infrastructure network. This second approach was experimentally tested, and a human operator could successfully move 10 mobile agents over the infrastructure network.

Despite the recent advances described in this chapter, the study of human–swarm interactions is still in its infancy. We still do not understand what the correct abstractions should be when interacting with complex networks, nor what the appropriate performance measures might be that ultimately determine the viability of the abstractions. As such, much work yet remains to be done in this increasingly relevant area of research.

References

1. Bicchi, A., Prattichizzo, D.: Manipulability of cooperating robots with unactuated joints and closed-chain mechanisms. IEEE Trans. Robot. Autom. **16**(4), 336–345 (2000)
2. Bicchi, A., Melchiorri, C., Balluchi, D.: On the mobility and manipulability of general multiple limb robots. IEEE Trans. Robot. Autom. **11**(2), 215–228 (1995)
3. Bullo, F., Cortés, J., Martínez, S.: Distributed Control of Robotic Networks. A Mathematical Approach to Motion Coordination Algorithms. Princeton University Press, Princeton (2009)

4. de la Croix, J.P., Egerstedt, M.: Controllability characterizations of leader-based swarm interactions. In: AAAI Symposium on Human Control of Bio-Inspired Swarms, Arlington, DC, Nov. 2012 (2012)
5. Egerstedt, M.: Controllability of networked systems. In: Mathematical Theory of Networks and Systems, Budapest, Hungary, 2010, pp. 57–61 (2010)
6. Egerstedt, M., Martini, S., Cao, M., Camlibel, K., Bicchi, A.: Interacting with networks: how does structure relate to controllability in single-leader consensus networks? IEEE Control Syst. Mag. **32**(4), 66–73 (2012)
7. Eren, T., Belhumeur, P.: A framework for maintaining formations based on rigidity. In: Proc. 15th IFAC World Congress, pp. 2752–2757 (2002)
8. Fax, J.A., Murray, R.M.: Graph Laplacian and stabilization of vehicle formations. In: Proc. 15th IFAC World Congress (2002)
9. Girden, E.R.: ANOVA: Repeated Measures. Sage, Thousand Oaks (1991)
10. Godsil, C., Royle, G.: Algebraic Graph Theory. Springer, New York (2001)
11. Hart, S.G., Staveland, L.E.: Development of NASA-TLX (Task Load Index): results of empirical and theoretical research. In: Hancock, A., Meshkati, N. (eds.) Human Mental Workload. North-Holland, Amsterdam (1988)
12. Jadbabaie, A., Lin, J., Morse, A.S.: Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Trans. Autom. Control **48**(6), 988–1001 (2003)
13. Ji, M., Egerstedt, M.: Distributed coordination control of multi-agent systems while preserving connectedness. IEEE Trans. Robot. **23**(4), 693–703 (2007)
14. Ji, M., Azuma, S., Egerstedt, M.: Role assignment in multi-agent coordination. Int. J. Assist. Robot. Mechatron. **7**(1), 32–40 (2006)
15. Kawashima, H., Egerstedt, M.: Approximate manipulability of leader-follower networks. In: IEEE Conference on Decision and Control and European Control Conference, pp. 6618–6623 (2011)
16. Martini, S., Egerstedt, M., Bicchi, A.: Controllability decompositions of networked systems through quotient graphs. In: IEEE Conference on Decision and Control, Cancun, Mexico, Dec. 2008, pp. 2717–2722 (2008)
17. Mesbahi, M., Egerstedt, M.: Graph Theoretic Methods in Multiagent Networks. Princeton University Press, Princeton (2010)
18. Ogren, P., Egerstedt, M., Hu, X.: A control Lyapunov function approach to multi-agent coordination. IEEE Trans. Robot. Autom. **18**(5), 847–851 (2002)
19. Olfati-Saber, R., Murray, R.M.: Distributed structural stabilization and tracking for formations of dynamic multi-agents. In: Proc. 41st IEEE Conf. Decision Control, Dec. 2002, vol. 1, pp. 209–215, (2002)
20. Olfati-Saber, R., Murray, R.M.: Consensus protocols for networks of dynamic agents. In: American Control Conference, Denver, CO, June 2003, pp. 951–956 (2003)
21. Rahmani, A., Ji, M., Mesbahi, M., Egerstedt, M.: Controllability of multi-agent systems from a graph-theoretic perspective. SIAM J. Control Optim. **48**(1), 162–186 (2009)
22. Ren, W., Beard, R.W.: Consensus seeking in multiagent systems under dynamically changing interaction topologies. IEEE Trans. Autom. Control **50**(5), 655–661 (2005)
23. Reynolds, C.: Flocks, herds and schools: a distributed behavioral model. Proc. ACM SIGGRAPH **21**(4), 25–34 (1987)
24. Roth, B.: Rigid and flexible frameworks. Am. Math. Mon. **88**(1), 6–21 (1981)
25. Tanner, H., Jadbabaie, A., Pappas, G.J.: Flocking in fixed and switching networks. IEEE Trans. Autom. Control **52**(5), 863–868 (2007)
26. Yoshikawa, T.: Manipulability of robotic mechanisms. Int. J. Robot. Res. **4**(2), 3–9 (1985)
27. Zhang, S., Camlibel, K., Cao, M.: Controllability of diffusively-coupled multi-agent systems with general and distance regular coupling topologies. In: IEEE Conference on Decision and Control, Orlando, FL, Dec. 2011 (2011)

http://www.springer.com/978-3-319-08436-7

Large-Scale Networks in Engineering and Life Sciences
Benner, P.; Findeisen, R.; Flockerzi, D.; Reichl, U.; Sundmacher, K. (Eds.)
2014, XIV, 388 p. 111 illus., 63 illus. in color., Hardcover
ISBN: 978-3-319-08436-7
A product of Birkhäuser Basel