

On collective motion in sensor networks: sample problems and distributed algorithms

Anurag Ganguli Sara Susca Sonia Martínez Francesco Bullo Jorge Cortés

Abstract—Adopting a tutorial approach, this paper surveys some control and systems theory problems that have recently gained interest in the context of multi-vehicle and sensor networks. By means of illustrative examples, we discuss some challenges in modeling of robotic networks, motion coordination algorithms, sensing and estimation tasks, and complexity of distributed algorithms.

I. INTRODUCTION

Motion coordination is an extraordinary phenomenon in biological systems such as schools of fish, see Fig. 1, and serves as a remarkable tool for man-made groups of robotic vehicles and active sensors. Although each individual agent has no global knowledge about the group as a whole or about the surrounding environment, complex coordinated behaviors emerge from local interactions.

The objective of this paper is to present, in a tutorial spirit, some sample problems and solutions in the emerging discipline of motion coordination for robotic sensor networks. The key idea is that spatially-distributed sensing tasks, such as surveillance, search and monitoring, can be performed efficiently by robotic networks of sensors.

We begin by discussing models of robotic networks, i.e., groups of agents that can sense, communicate and take local control actions. We present basic notions of coordination tasks and time complexity in an attempt to provide a unifying modeling language for robotic networks. Next, we survey the state of the art on motion coordination by presenting some results on the design of coordination primitives, i.e., basic coordination skills for specific tasks such as deployment or pursuit. Remarkably, the problem of deploying a group of agents to form an arbitrary pattern using distributed decision-making and limited communication is in general an open problem. We emphasize here that the scope of the tools described in this paper is not limited to the problems mentioned within but can be applied to other tasks of similar nature. Indeed such an approach will be useful in the case where the same network of simple mobile agents is required to perform a variety of different motion coordination tasks.

A third focus of this paper is the use of controlled mobility in target and boundary tracking problems. Indeed, interesting

and well-motivated coordination problems arise from practical applications where the robotic network is required to track mobile targets or to estimate environmental boundaries, as those exhibited by gas diffusion, heat radiation, or fluid spills. (A third problem is that of estimating environmental fields such as deterministic functions of the environment, e.g., concentration of a pollutant in a lake, and probabilistic maps representing likelihood of events taking place in the environment. We will not talk much about this subject.) By means of some example scenarios, we illustrate how to characterize optimal sensor placement or motion patterns, design distributed sensing schemes, and integrate them with motion coordination algorithms. The examples only begin to shed light onto a large set of challenging control problems in which node mobility, communication, computation, and sensing aspects are jointly considered.

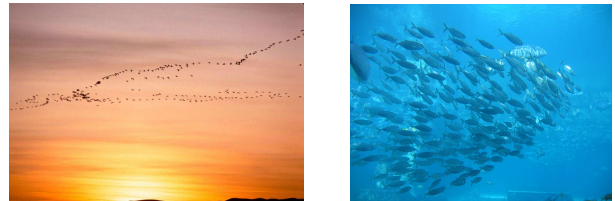


Fig. 1. Geese flying in formation (source: U.S. Fish & Wildlife Service <http://www.fws.gov/midwest/swanlake>) and a school of fish (location: IEEE CDC 2004 at Paradise Island, Bahamas).

The paper is organized as follows. In Section 2, we discuss models of multi-agent networks. In Section 3, we illustrate interesting algorithms for basic tasks such as deployment and rendezvous. In Section 4, we focus on the use of mobility in tracking moving targets and boundaries. In all sections we present sample networks, communication graphs, coordination tasks and algorithms and we discuss the information flow between agents. Additionally, in each section we highlight some potentially interesting open problems.

II. ROBOTIC NETWORKS AND COMPLEXITY

The global behavior of a robotic network can be seen as the sum of the local actions taken by its members. Each robot in the network can sense its immediate environment, communicate with its neighbors, process the information gathered and move according to it. The integrated capabilities together determine the behavior of each agent, which in turn impacts the overall collective response. This makes a robotic network a very versatile system and also a very complex one due to the confluence of processing, communication and sensing aspects. Therefore, to understand the trade-offs

Anurag Ganguli is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 and with the Department of Mechanical and Environmental Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106.

Sara Susca, Sonia Martínez and Francesco Bullo are with the Department of Mechanical and Environmental Engineering and with the Center for Control, Dynamical Systems and Computation, University of California at Santa Barbara, Santa Barbara, CA 93106.

Jorge Cortés is with the Department of Applied Mathematics and Statistics, University of California at Santa Cruz, Santa Cruz, CA 95064.

between performance, reliability of algorithms and their costs (energy, time, communication, etc), it seems appropriate to propose a common modeling framework where the execution of different coordination algorithms can be appropriately formalized, analyzed and compared.

Since this is an important topic, we briefly present in this section some of the concepts that we think should be present in such a model. We do not attempt here to present this model in its full generality, but rather give a flavor of this research avenue. For a more detailed discussion, we refer the reader to [1] and to the related models proposed in [2], [3], [4].

We consider *uniform networks of robotic agents* (or *robotic networks*) defined by a tuple $\mathcal{S} = (I, \mathcal{A}, E_{\text{comm}}, X_0)$ consisting of

- (i) $I = \{1, \dots, N\}$; a set of unique identifiers (UIDs);
- (ii) $\mathcal{A} = \{A^{[i]}\}_{i \in I}$, with $A^{[i]} = (X, U, X_0, f)$ (Here, X is the *state space*, U is the *input space*, X_0 is the *set of allowable initial states* and f is a C^∞ map with domain $X \times U$), a set of identical control systems; this set is called the *set of physical agents*;
- (iii) E_{comm} , a map from $\prod_{i \in I} X$ to the subsets of $I \times I \setminus \text{diag}(I \times I)$ called the *communication edge map*.

An edge between two nodes in E_{comm} implies the ability of the corresponding two agents to exchange messages.

Next, a (*synchronous, dynamic*) *control and communication law* for \mathcal{S} consists of the sets:

- (i) $\mathbb{T} = \{t_\ell\}_{\ell \in \mathbb{N}_0} \subset \bar{\mathbb{R}}_+$, an increasing sequence of time instants, called *communication schedule*;
- (ii) L , a set containing the null element, called the *communication language*; elements of L are called *messages*;
- (iii) W , sets of values of some *logic variables* $w^{[i]} \in W$, $i \in I$. These sets correspond to the capability of agents to allocate additional variables and store sensor or communication data;
- (iv) $W_0 \subseteq W$, subsets of *allowable initial values*;

and the maps:

- (i) $\text{msg}: \mathbb{T} \times X \times W \times I \rightarrow L$, $i \in I$, called *message-generation function*;
- (ii) $\text{stf}: \mathbb{T} \times W \times L^N \rightarrow W$, called *state-transition function*;
- (iii) $\text{ctl}: \bar{\mathbb{R}}_+ \times X \times X \times W \times L^N \rightarrow U$, called *control function*.

To implement a control and communication law each agent performs the following sequence or cycle of actions. At each instant $t_\ell \in \mathbb{T}$, agent i sends to agent j a message computed by applying the message-generation function to the current values of t_ℓ , $x^{[i]}$ and $w^{[i]}$. After a negligible period of time (therefore, still at $t_\ell \in \mathbb{T}$), agent i resets the value of its logic variables $w^{[i]}$ by applying the state-transition function to the current value of $w^{[i]}$, and to the messages $y^{[i]}(t_\ell)$ received at t_ℓ . Between communication instants, i.e., for $t \in [t_\ell, t_{\ell+1})$, agent i applies a control action computed by applying the control function to $x^{[i]}(t_\ell)$, the current values of $x^{[i]}$ and $w^{[i]}$, and to the messages $y^{[i]}(t_\ell)$ received at t_ℓ .

Let us present some brief comments. In our present definition, all agents are identical and implement the same algorithm; in this sense the control and communication law is called *uniform* (or anonymous). If $W = W_0 = \emptyset$, then the control and communication law is *static* (or memoryless) and

no state-transition function is defined. It is also possible for a law to be *time-independent* if the three relevant maps do not depend on time. In most uniform control and communication laws, the messages interchanged among the network agents are (quantized representations of) the agents' states. In what follows we focus on the static time-independent case.

To analyze the performance of a motion coordination algorithm, we need to establish the notion of coordination task, and of task achievement by a robotic network. A (*static*) *coordination task* for a network \mathcal{S} is a map $\mathcal{T}: \prod_{i \in I} X^{[i]} \rightarrow \{\text{true}, \text{false}\}$. Additionally, let \mathcal{CC} be a control and communication law for \mathcal{S} . We say that \mathcal{CC} *achieves* the task \mathcal{T} if for all initial conditions $x_0^{[i]} \in X_0$, the corresponding network evolution $t \mapsto x(t)$ has the property that there exists $T \in \mathbb{R}_+$ such that $\mathcal{T}(x(t)) = \text{true}$ for $t \geq T$.

In some situations achieving a task efficiently means stabilizing the system. In other situations efficiency might be measured by required communication/control energy or by speed of completion. For the latter, we can establish the following notions of time complexity.

- (i) The *time complexity to achieve \mathcal{T} with \mathcal{CC} from $x_0 \in \prod_{i \in I} X_0^{[i]}$* is

$$\text{TC}(\mathcal{T}, \mathcal{CC}, x_0) = \inf \{ \ell \mid \mathcal{T}(x(t_k)) = \text{true}, \forall k \geq \ell \},$$

where $t \mapsto (x(t))$ is the evolution of $(\mathcal{S}, \mathcal{CC})$ from x_0 .

- (ii) The *time complexity to achieve \mathcal{T} with \mathcal{CC}* is

$$\text{TC}(\mathcal{T}, \mathcal{CC}) = \sup \{ \text{TC}(\mathcal{T}, \mathcal{CC}, x_0) \mid x_0 \in \prod_{i \in I} X_0^{[i]} \}.$$

- (iii) The *time complexity of \mathcal{T}* is

$$\text{TC}(\mathcal{T}) = \inf \{ \text{TC}(\mathcal{T}, \mathcal{CC}) \mid \mathcal{CC} \text{ achieves } \mathcal{T} \}.$$

Another important notion is that of communication complexity. Roughly speaking, this represents the total number of messages exchanged to complete a task. In the following sections, we will describe certain coordination algorithms, some of which have been cast into this modeling framework and their complexity properties analyzed; see [1]. In the interest of space and to preserve the tutorial flavor of the paper, we will not model the algorithms in this framework but will only provide an informal description of them. We will, however, state their complexity properties whenever possible.

III. MOTION COORDINATION

Loosely speaking, by a motion coordination problem we mean a task where the network objective can be captured by the final spatial configuration of its agents and/or of their velocity vectors. Key problems include flocking [5], foraging [6], [7], rendezvous [8], [9], pursuit [10], coverage [11], [12], cooperative search [13], and formation control [14]. Behavior-based robotics, see [15], [16], [17], [18], has investigated heuristic approaches to the design of emerging behaviors. Lately there been a systematic effort to design scalable and efficient algorithms; see [5], [11], [9].

Our method of approaching motion coordination problems exploits their inherent geometric [19], [20], graph-theoretical [21], and optimization [22] structure. The sensing capabilities of the agents are captured through geometric

models; the information flow/neighborhood relationship of the agents is represented by appropriate graphs; and the network objective is characterized via appropriate utility functions. Algorithms are then designed via gradient/greedy methods. We illustrate our approach by discussing two basic types of problems: deployment and rendezvous.

A. Deployment problems

First, we consider the *area-coverage deployment* problem in a convex polygonal environment. The objective is to maximize the area within close range of the mobile nodes. This models a scenario in which the nodes take local measurements. Assume that certain regions in the environment are more important than others and describe this by a density function ϕ . Our recent work [11], [23] shows how this problem leads to the coverage performance metric $\mathcal{H}(p_1, \dots, p_N) = \sum_{i=1}^N \int_{V_i} f(\|q - p_i\|) \phi(q) dq$. Here p_i is the position of the i th node, f measures the performance of an individual sensor, and $\{V_1, \dots, V_N\}$ is the Voronoi partition of the nodes $\{p_1, \dots, p_N\}$. If we assume that each node obeys a first order dynamical behavior, then a simple gradient scheme can be easily implemented in a spatially-distributed manner. Because the closed-loop system is a gradient flow for the cost function \mathcal{H} , performance is locally, continuously optimized. Fig. 2 illustrates the performance of this coordination algorithm. As a special case, when the environment is a segment and $\phi = 1$, the time complexity of the algorithm can be shown to be $O(N^3 \log(N\epsilon^{-1}))$ where ϵ is a threshold value below which we consider the task accomplished; see [1].

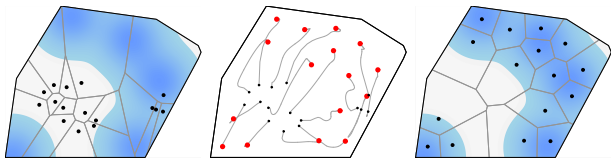


Fig. 2. Area-coverage deployment for 16 agents on a convex polygonal environment. The shaded region represents the density function ϕ . The left (respectively, right) figure illustrates the initial (respectively, final) locations and Voronoi partition. The central figure illustrates the gradient ascent flow.

Second, we consider the problem of deploying *to maximize the likelihood of detecting a source*. For example, consider devices equipped with acoustic sensors attempting to detect a sound-source (or similarly, antennas detecting RF signals, or chemical sensors localizing a pollutant source). For a variety of criteria, when the source emits a known signal and the noise is Gaussian, we know that (1) the optimal detection algorithm involves a matched filter, (2) detection performance is a function of signal-to-noise-ratio, and, in turn, (3) signal-to-noise ratio is inversely proportional to the sensor-source distance. How do we deploy the nodes and maximize the detection probability? We design a motion coordination algorithm to maximize detection likelihood as follows: each node moves toward the circumcenter of its Voronoi cell. Our work [24] shows that (1) the detection likelihood is inversely proportional to the circumradius of each node’s Voronoi cell, and (2) if the nodes follow this algorithm, then the detection likelihood increases monotonically as a function of time; see

Fig. 3. (This algorithm is designed for the detection problem; source localization/tracking is discussed in the next section.)

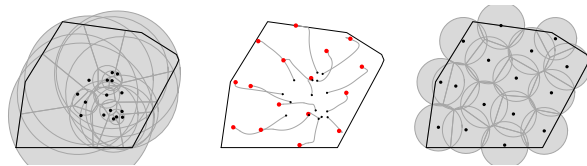


Fig. 3. The nodes solve a maximum-likelihood-detection deployment; the figure depicts the nodes final position and the circumcircles for each node.

Third, we consider a *visibility-based deployment* of nodes in a planar non-convex polygonal environment. Here, the coverage objective is to deploy the ad hoc network in such a way as to obtain complete visibility of the environment. This coverage problem is a distributed feedback version of the well-known “art gallery problem” from computational geometry [25]. Let us now describe an algorithm for this type of deployment. At every time instant, each node p_i computes a dominance region as the set of points for which p_i is either the only visible node or the closest visible node; p_i then moves toward the furthest vertex in its dominance region. The performance of this algorithm is, at this time, known only via simulations on a class of floor plan environments; e.g., see Fig. 4.

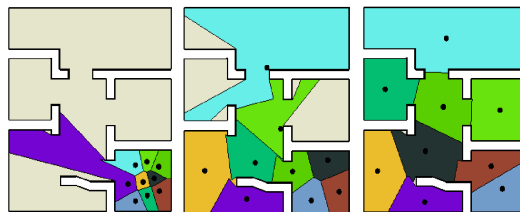


Fig. 4. The nodes solve the visibility deployment problem for a nonconvex polygonal environment shaped as a typical floor-plan.

B. Rendezvous problems

In the context of motion coordination, the rendezvous objective is to achieve agreement over the location of the agents, that is, to steer each agent to a common location. We consider two scenarios which differ in the agents’ sensing/communication capabilities and the environment to which the agents belong. Let $P = \{p_1, \dots, p_N\}$ represent the set of locations of the agents.

Let us first consider the problem of rendezvous for agents equipped with *range-limited sensors*. In this case, each agent is capable of sensing in a closed disk of bounded radius and belongs to the unbounded space \mathbb{R}^d of arbitrary dimension d . This is described by the r -disk graph, $\mathcal{G}_{\text{disk}}(r)$, in which two agents are neighbors if and only if the Euclidean distance between them is less than or equal to r . For a complete discussion of this problem, see [26].

Second, we consider *visually-guided agents*. Here the agents are assumed to belong to a nonconvex simple polygonal environment Q . Each agent can sense within line-of sight any other agent as well as sense the distance to the

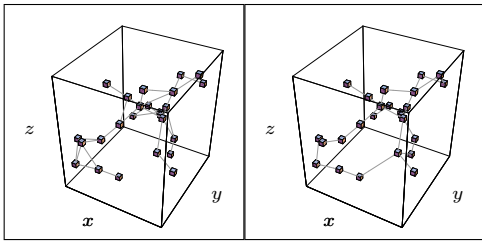


Fig. 5. The r -disk and Relative Neighborhood ($\mathcal{G}_{RN}(r)$) graphs in \mathbb{R}^3 for a configuration of 25 agents with coordinates uniformly randomly generated within the cube $[-7, 7] \times [-7, 7] \times [-7, 7]$. The parameter r is taken equal to 4. See [26] for a discussion on proximity graphs.

boundary of the environment. The relationship between the agents can be characterized by the visibility graph, $\mathcal{G}_{vis,Q}$. Two agents are neighbors if they are mutually visible to each other; see [27] for a complete discussion.

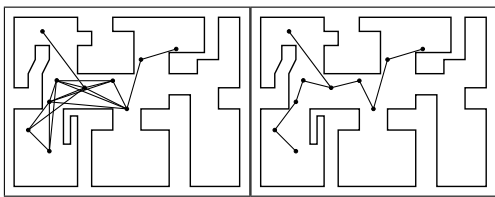


Fig. 6. From left to right, visibility graph and the locally-cliqueless visibility graph. The locally-cliqueless visibility graph is spatially distributed over the visibility graph (see [27]).

In both scenarios, the rendezvous problem cannot be solved with distributed information if the agents are placed in such a way that they do not form a connected sensing/communication graph. Arguably, a good property of any algorithm for rendezvous is that of maintaining some form of connectivity between agents, which in turn imposes constraints on the agents' motion. Motion constraints that maintain connectivity are designed in [8], [27] and exploit the geometric properties of proximity graphs. These constraints need not be imposed for all pair of neighbors. Instead, it is enough to impose them according to certain sparse proximity graphs. For the disk graph scenario, an appropriate graph is the so-called Relative Neighborhood graph depicted in Fig. 5.

We are now ready to outline an algorithm that solves the problems for both communication scenarios. The agents execute what we shall refer to as the *Circumcenter Algorithm*; here is an informal description. Each agent iteratively performs the following tasks:

- 1: detects its neighbors according to \mathcal{G}
- 2: computes the circumcenter of the point set comprised of its neighbors and of itself
- 3: moves toward this circumcenter while maintaining connectivity with its neighbors.

Fig. 7 and 8 illustrate the performance of the Circumcenter Algorithm for the first and second scenario, respectively. One can prove that, under technical conditions, the algorithm does achieve the rendezvous task in both scenarios. Additionally, when $d = 1$, it can be shown that the time complexity of the task using the Circumcenter Algorithm is $\Theta(N)$; see [1].

In this section, we have provided examples of certain motion coordination tasks and outlined approaches to solving

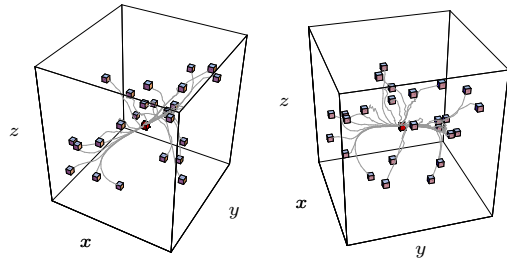


Fig. 7. Evolution (in light gray) of the Circumcenter Algorithm from two viewpoints. The initial configuration of the network is as in Fig. 5.

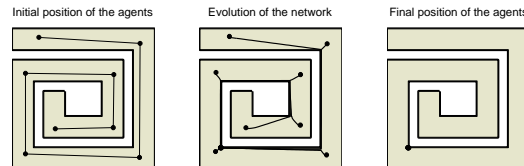


Fig. 8. Simulation results of the Circumcenter Algorithm on a network of agents distributed in a spiral polygon. The algorithm is run over $\mathcal{G}_{vis,Q}$.

the problems. However, many open research questions still remain unanswered. One such problem is that of achieving arbitrary patterns. The problem of deploying and controlling visually-guided agents is another problem where a deeper understanding is needed. Apart from motion coordination, another class of interesting problems is that of localizing and estimating moving targets and fields. We shall try to illustrate some of these problems in the following section.

IV. TARGET AND BOUNDARY TRACKING

The subject of this section is the design of algorithms that exploit controlled mobility to efficiently localize moving targets (or sources) and boundaries, and to efficiently estimate environmental fields (here we mean both functions of the environment, e.g., concentration of a pollutant in a lake, and probabilistic maps representing likelihood of events taking place in the environment, e.g., occupancy maps).

A possible approach to exploiting controlled mobility is based on a *next-best-view* paradigm. The key idea is to design greedy policies that move the network nodes in such a way as to maximize the information that the nodes will gather with subsequent measurements. Put into a broader perspective, an integrated algorithm entails an estimation filter and a motion coordination algorithm that takes the network agents to optimal sensor positions. Accordingly, a fundamental objective of this approach is to characterize optimal sensor placements or optimal sensor motion patterns for various estimation problems.

The literature on (static) sensor networks performing various estimation tasks is vast and we only mention the two surveys [28], [29] that are somehow related to our approach. From a robotic viewpoint, an incomplete list of works on active target tracking for controlled-mobility networks includes [30] and [31]. Related to our next-best-view and optimal sensor placement approach is the literature on optimum experimental design. Here the references [32], [33] show how to define appropriate "sensitivity performance measure" for optimal sensor placement; see also [34]. Boundary estimation

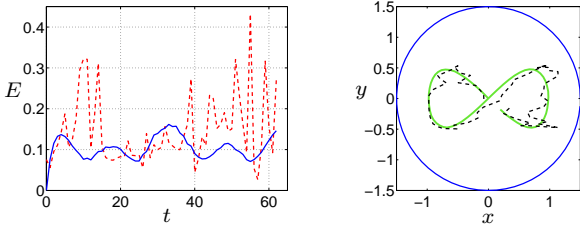


Fig. 9. Comparison of target localization algorithm for static and controlled mobile sensors. The left figure compares the estimation errors and the right figure shows the estimated target trajectories. The lower error and the more accurate “figure-eight” trajectory are achieved by the controlled mobile sensors. The mobile sensors move along the boundary of the environment (in this case, a circle).

has been recently studied in the context of static sensor fields; e.g., see [35] and references therein. Researchers in mobile robotics have explored alternative approaches for boundary estimation. A motion coordination mechanism is designed in [36] by relying on certain energy-minimizing curves from image processing. A related gradient climbing algorithm is proposed in [37].

A. Target tracking

In this section we present an example approach to target tracking. For this problem, an appropriate sensitivity performance measure in 2D and 3D environments is the determinant of the Fisher Information Matrix (FIM). The determinant measures [38] the information produced by a set of measurements in estimating a set of unknown parameters; its inverse, called the Cramer-Rao-Lower-Bound, characterizes the best achievable estimation error covariance. Under the assumptions of Gaussian independent noise, a 2D environment, and a stationary sound-source, the global maxima of the FIM determinant correspond to an optimal pattern in which the sensors are uniformly placed in circular fashion around the target. We use this information to improve the performance of a Kalman filter-based algorithm for target localization. In short, we implement a motion coordination algorithm that steers the mobile sensor network to an optimal deployment; we do not detail this algorithm here, but note that it is related to the ones presented in the previous section. A schematic description of the algorithm is as follows. Each agent iteratively performs the following:

- 1: measures target location and shares new measurement with neighbors
- 2: computes new estimate of target location
- 3: moves according to motion algorithm (based on target estimate and neighbors’ positions).

Fig. 9 (see [31]), illustrates how this integrated motion/sensing/estimation algorithms lead to improved performance of an extended Kalman filter in a target tracking scenario where the target moves along a “figure-eight” pattern.

B. Boundary estimation

Here we consider a boundary estimation problem. The aim is to select an interpolation technique and to deploy the sensors in such a way as to optimally estimate the boundary.

In other words, we define a cost function quantifying an estimation error and then design a motion coordination algorithm that minimizes it. The details are as follows. Assume that the unknown set Q is the planar subset where a certain environmental quantity, e.g., heat or chemical concentration, is above a given threshold. The objective is to estimate the boundary ∂Q by means of an array of sensors able to locally detect ∂Q and to move towards and along it. Let us consider the following basic task: how to place the robots along ∂Q in such a way that the polygon, whose vertices are the robots’ positions, is a good approximation of Q . To simplify the following discussion, we assume that Q is convex. Therefore, our optimal estimation problem is equivalent to finding the “best” N -vertices polytope inscribed inside Q that best approximates Q according to some metric. This setup is interesting also because polygonal approximations of planar convex bodies is a well-studied subject, e.g., see the survey [39] and references therein. It is known, for example, that the distance between the convex body Q and its best (as measured according to various metrics) inner polygonal approximation belongs to $O(\frac{1}{N^2})$.

Let us formalize one of these error formulations. Once the robots reach the boundary we order them in counterclockwise order $\{p_1, \dots, p_N\}$; for convenience, we set $p_0 = p_N$ and $p_{N+1} = p_1$. Among the possible choices of metric we consider $\mathcal{H}(g, g_I) := \int_a^b \|g(t) - g_I(t)\| dt$, where g and g_I are parametric representations of the boundary of Q and of the interpolating lines between any two nodes, respectively. It turns out that \mathcal{H} is the area of the convex set Q minus its inner approximating polygon. We regard \mathcal{H} as the cost that we minimize through a motion coordination algorithm. Thus

$$\min_{p_1, \dots, p_N \in \partial Q} \mathcal{H}(g, g_I) = A(Q) - \max_{p_1, \dots, p_N \in \partial Q} A(\text{co}(p_1, \dots, p_N)),$$

where A is area function, and co is the convex hull of its arguments. \mathcal{H} is always non-negative. The area of the polygon $\text{co}(p_1, \dots, p_N)$ is easily expressed as a function of position of the vertices, that is, $A(\text{co}(p_1, \dots, p_N)) = \frac{1}{2} \sum_{k=1}^N (x_k y_{k+1} - x_{k+1} y_k)$, where $p_k = (x_k, y_k)$. To maximize \mathcal{H} we consider the following gradient flow:

$$\dot{p}_i = \text{proj}_{T\partial Q} \left(\frac{\partial A(\text{co}(p_1, \dots, p_N))}{\partial p_i} \right) = \frac{1}{2} \text{proj}_{T\partial Q} \begin{pmatrix} y_{i+1} - y_{i-1} \\ x_{i-1} - x_{i+1} \end{pmatrix},$$

where $\text{proj}_{T\partial Q}$ is the orthogonal projection onto the tangent contour $T\partial Q$. (A nonsmooth gradient flow can be designed to handle nonsmooth contours.) Note that, in order to implement this gradient flow, every agent needs to know only the positions of its immediate clockwise and counterclockwise neighbors and of the gradient of the contour at its position; as for the target tracking problem, this information requirements can be formalized using proximity graph models as in the previous sections. By design, the gradient flow is guaranteed to lead the robots to the set of critical configurations of \mathcal{H} (see Fig. 10). It also turns out that \mathcal{H} is not strictly concave and it possesses multiple critical points.

In this section, we have discussed two types of problems related to target tracking and localization for mobile sensor networks. Numerous open questions remain. For example, there is a need for appropriate “sensitivity performance measures” for next-best-view algorithms in target, boundary and function estimation. Another interesting area is the study of different sensor models, e.g., sensors with limited footprint.

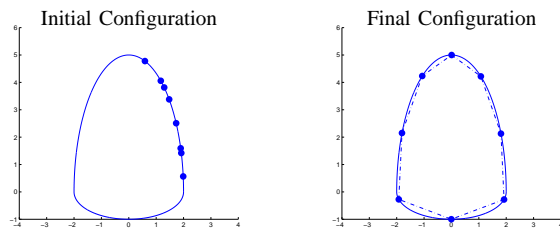


Fig. 10. Gradient flow maximizing inner approximating polygon ($N = 8$).

V. CONCLUSIONS

This paper is a partial survey of some control problems related to collective motion and estimation for sensor networks. Specifically, we have talked about deployment and rendezvous as examples of motion coordination tasks and target tracking and boundary estimation as examples of localization/estimation tasks. We have outlined possible approaches to these problems and mentioned some new research directions in this area.

Acknowledgments: This material is based upon work supported in part by ONR YIP Award N00014-03-1-0512, NSF SENSORS Award IIS-0330008, and DARPA/AFOSR MURI Award F49620-02-1-0325.

REFERENCES

- [1] S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli, "Synchronous robotic networks and complexity of control and communication laws," Jan. 2005. <http://xxx.arxiv.org/math.OA/0501499>.
- [2] I. Suzuki and M. Yamashita, "Distributed anonymous mobile robots: Formation of geometric patterns," *SIAM J. on Computing*, vol. 28, no. 4, pp. 1347–1363, 1999.
- [3] N. A. Lynch, *Distributed Algorithms*. San Mateo, CA: Morgan Kaufmann Publishers, 1997.
- [4] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer, "Hard tasks for weak robots: The role of common knowledge in pattern formation by autonomous mobile robots," in *ISAAC 1999, 10th Int. Symp. Algorithm and Computation (Chennai, India)* (A. Aggarwal and C. P. Rangan, eds.), pp. 93–102, New York: Springer Verlag, 1999.
- [5] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.
- [6] K. M. Passino, *Biomimicry for Optimization, Control, and Automation*. New York: Springer Verlag, 2004.
- [7] Y. Liu and K. M. Passino, "Stable social foraging swarms in a noisy environment," *IEEE Trans. on Automatic Control*, vol. 49, no. 1, pp. 30–44, 2004.
- [8] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, "Distributed memoryless point convergence algorithm for mobile robots with limited visibility," *IEEE Trans. on Robotics and Automation*, vol. 15, no. 5, pp. 818–828, 1999.
- [9] J. Lin, A. S. Morse, and B. D. O. Anderson, "The multi-agent rendezvous problem," in *IEEE Conf. on Decision and Control*, (Maui, Hawaii), pp. 1508–1513, Dec. 2003.
- [10] J. A. Marshall, M. E. Broucke, and B. A. Francis, "Formations of vehicles in cyclic pursuit," *IEEE Trans. on Automatic Control*, vol. 49, no. 11, pp. 1963–1974, 2004.
- [11] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
- [12] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem," in *Distributed Autonomous Robotic Systems (DARS02)*, (Fukuoka, Japan), pp. 299–308, June 2002.
- [13] L. E. Parker and C. Touzet, "Multi-robot learning in a cooperative observation task," in *Distributed Autonomous Robotic Systems 4* (L. E. Parker, G. Bekey, and J. Barhen, eds.), pp. 3–12, Springer Verlag, 2000.
- [14] K. Sugihara and I. Suzuki, "Distributed algorithms for formation of geometric patterns with many mobile robots," *J. Robotic Systems*, vol. 13, no. 3, pp. 127–39, 1996.
- [15] R. A. Brooks, "A robust layered control-system for a mobile robot," *IEEE J. Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.
- [16] R. C. Arkin, *Behavior-Based Robotics*. Cambridge, UK: Cambridge University Press, 1998.
- [17] M. J. Mataric, "Behavior-based control: Examples from navigation, learning, and group behavior," *J. Experimental and Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 323–336, 1997. Special issue on Software Architectures for Physical Agents.
- [18] T. Balch and L. E. Parker, eds., *Robot Teams: From Diversity to Polymorphism*. Natick, MA: A K Peters Ltd., 2002.
- [19] J. W. Jaromczyk and G. T. Toussaint, "Relative neighborhood graphs and their relatives," *Proceedings of the IEEE*, vol. 80, no. 9, pp. 1502–1517, 1992.
- [20] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley Series in Probability and Statistics, New York, NY: John Wiley & Sons, second ed., 2000.
- [21] C. D. Godsil and G. F. Royle, *Algebraic Graph Theory*. GTM, New York: Springer Verlag, 2001.
- [22] Z. Drezner and H. W. Hamacher, eds., *Facility Location: Applications and Theory*. New York: Springer Verlag, 2001.
- [23] J. Cortés, S. Martínez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM. Control, Optimisation & Calculus of Variations*, Jan. 2004. To appear.
- [24] J. Cortés and F. Bullo, "Coordination and geometric optimization via distributed dynamical systems," *SIAM J. Control and Optimization*, June 2004. To appear.
- [25] J. O'Rourke, *Art Gallery Theorems and Algorithms*. Oxford, UK: Oxford University Press, 1987.
- [26] J. Cortés, S. Martínez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Trans. on Automatic Control*, July 2004. To appear.
- [27] A. Ganguli, J. Cortés, and F. Bullo, "On rendezvous for visually-guided agents in a nonconvex polygon," in *IEEE Conf. on Decision and Control*, (Seville, Spain), Dec. 2005. To appear.
- [28] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. S. Sastry, "Distributed control applications within sensor networks," *IEEE Proceedings*, vol. 91, no. 8, pp. 1235–1246, 2003. Special issue on Sensor Networks and Applications.
- [29] D. Li, K. Wong, Y. H. Hu, and A. Sayeed, "Detection, classification and tracking of targets in distributed sensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17–29, 2002.
- [30] B. S. Y. Rao, H. F. Durrant-Whyte, and J. S. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance," *Int. J. of Robotics Research*, vol. 12, no. 1, pp. 20–44, 1993.
- [31] S. Martínez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, Nov. 2004. Submitted.
- [32] B. Porat and A. Nehorai, "Localizing vapor-emitting sources by moving sensors," *IEEE Trans. on Signal Processing*, vol. 44, no. 4, pp. 1018–1021, 1996.
- [33] D. Uciński, *Optimal Measurement Methods for Distributed Parameter System Identification*. Boca Raton, FL: CRC Press, 2004.
- [34] T. H. Chung, V. Gupta, J. W. Burdick, and R. M. Murray, "On a decentralized active sensing strategy using mobile sensor platforms in a network," in *IEEE Conf. on Decision and Control*, (Paradise Island, Bahamas), pp. 1914–1919, 2004.
- [35] R. Nowak and U. Mitra, "Boundary estimation in sensor networks: Theory and methods," in *International Workshop on Information Processing in Sensor Networks (IPSN)*, (Palo Alto, CA), pp. 80–95, Apr. 2003.
- [36] D. Marthaler and A. L. Bertozzi, "Tracking environmental level sets with autonomous vehicles," in *Proc. of the Conference on Cooperative Control and Optimization*, (Gainesville, Florida), Dec. 2002.
- [37] P. Ögren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: adaptive gradient climbing in a distributed environment," *IEEE Trans. on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, 2004.
- [38] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York, NY: John Wiley & Sons, 2001.
- [39] P. M. Gruber, "Aspect of approximation of convex bodies," in *Handbook of Convex Geometry* (P. M. Gruber and J. M. Willis, eds.), vol. A, pp. 319–345, Oxford, UK: Elsevier, 1993.