

Gibbs sampler-based coordination of autonomous swarms[☆]

Wei Xi^{a,b}, Xiaobo Tan^{c,*}, John S. Baras^{a,b}

^a*Institute for Systems Research, University of Maryland, College Park, MD 20742, USA*

^b*Department of Electrical & Computer Engineering, University of Maryland, College Park, MD 20742, USA*

^c*Department of Electrical & Computer Engineering, Michigan State University, East Lansing, MI 48824, USA*

Received 16 April 2005; received in revised form 4 November 2005; accepted 5 March 2006

Available online 4 May 2006

Abstract

In this paper a novel, Gibbs sampler-based algorithm is proposed for coordination of autonomous swarms. The swarm is modeled as a Markov random field (MRF) on a graph with a time-varying neighborhood system determined by local interaction links. The Gibbs potential is designed to reflect global objectives and constraints. It is established that, with primarily local sensing/communications, the swarm configuration converges to the global minimizer(s) of the potential function. The impact of the Gibbs potential on the convergence speed is investigated. Finally a hybrid algorithm is developed to improve the efficiency of the stochastic scheme by integrating the Gibbs sampler-based method with the deterministic gradient-flow method. Simulation results are presented to illustrate the proposed approach and verify the analyses.

© 2006 Elsevier Ltd. All rights reserved.

Keywords: Gibbs sampler; Simulated annealing; Autonomous swarms; Multi-vehicle coordination; Potential functions

1. Introduction

With rapid advances in sensing, actuation, communication, and computation capabilities, autonomous unmanned vehicles (AUVs) are expected to cooperatively perform tasks in various hazardous, unknown or remote environments. In this paper, by an *autonomous swarm*, we mean a group of AUVs with potentially huge number of members. Distributed methods for control and coordination of autonomous swarms are especially appealing due to large scales of vehicle networks and communication constraints. Such methods have often been inspired by the swarming behaviors demonstrated by bacteria, insects, and animals (Leonard & Fiorelli, 2001; Passino, 2002), or by the swarming models proposed in physics, biology, and computer science (Levine, Rappel, & Cohen, 2000; Mogilner & Edelstein-Keshet, 1999; Reynolds, 1987; Vicsek, Czirook, Ben-Jacob, Cohen, & Shochet, 1995). Recent work on multi-agent control includes Olfati-Saber and Murray (2002),

Lawton, Beard, and Young (2003), Jadbabaie, Lin, and Morse (2003), Tanner, Jadbabaie, and Pappas (2003), Gazi and Passino (2003), Justh and Krishnaprasad (2004), Olfati-Saber and Murray (2004), Olfati-Saber (2006), Moreau (2005), to name a few.

A popular approach in multi-vehicle coordination is based on artificial potential functions, which encode desired vehicle behaviors such as inter-vehicle interactions, obstacle avoidance, and target approaching (Baras, Tan, & Hovareshti, 2003; Leonard & Fiorelli, 2001; Ogren, Fiorelli, & Leonard, 2004). This approach has been explored for path planning and control of robotic manipulators and mobile robots over the past two decades (Khatib, 1986; Rimon & Koditschek, 1992). Despite its simple, local, and elegant nature, this approach suffers from the problem that the system dynamics could be trapped at the local minima of potential functions (Koren & Borenstein, 1991). Researchers attempted to address this problem by designing potential functions that have no other local minima (Kim & Khosla, 1992; Volpe & Khosla, 1990), or by escaping from local minima using ad hoc techniques, e.g., random walk (Barraquand, Langlois, & Latombe, 1992) and virtual obstacles (Liu, Krishna, & Yong, 2000).

In this paper a novel, systematic approach is proposed to coordinate an autonomous swarm based on the theory of Markov random fields (MRFs) and Gibbs sampling, a tool that has been

[☆] This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Ioannis Paschalidis under the direction of Editor Ian Petersen.

* Corresponding author. Tel.: +1 517 4325671; fax: +1 517 3531980.

E-mail addresses: wxi@glue.umd.edu (W. Xi), xbtan@msu.edu (X. Tan), baras@isr.umd.edu (J.S. Baras).

traditionally used in statistical mechanics, image processing, computer vision, and data analysis and clustering (Carlin & Louis, 2000; Chellappa & Jain, 1993; Geman & Geman, 1984; Medvedovic, Young, & Bumgarner, 2004; Winkler, 1995). The approach aims to achieve global objectives (without being trapped at local minima of potential functions) using primarily local interactions together with limited global interactions. A discrete-time path planning setting is considered, where vehicles are allowed to move on a discretized grid in a 2D space. We focus on the high-level coordination and planning problem in this paper, and thus the vehicle dynamics is not included in the analysis. However, such dynamics should be considered during the low-level vehicle control in executing the planned trajectory.

A swarm is modeled as an MRF on a graph, where the (mobile) vehicles and their communication/sensing links constitute the vertices and the edges of the graph, respectively. As in the artificial potential approach, global objectives and constraints are reflected in potential functions—in this case, Gibbs potentials. The movement of vehicles is decided using simulated annealing based on the Gibbs sampler. The dynamic graph associated with the evolution of vehicle networks, however, presents significant challenges in convergence analysis since classical MRF theory does not apply directly.

We first present analysis on the single-vehicle case before embarking on the multi-vehicle case. The aforementioned dynamic graph now degenerates to a single vertex and the analysis is much simplified. The purpose of studying the single-vehicle case is twofold. Firstly, this is of interest in its own right by showing the convergence to the optimal location despite the existence of local minima, a clear advantage over the traditional gradient-flow method. Secondly, some analytical techniques and results for the single-vehicle case, e.g., the convergence bound and the impact of potential function design on the convergence speed, also apply to the multi-vehicle case.

To deal with the coordination of multiple vehicles, a two-step sampling scheme is proposed: in the first sampling a vehicle is picked through a properly designed, *configuration-dependent* proposal distribution, and in the second sampling step the vehicle makes a move using the local characteristics of the Gibbs distribution. It is shown that simulated annealing with such a two-step scheme leads to convergence to the configuration(s) of minimal potential. This scheme requires mostly local interactions among vehicles except a global communication mechanism for notifying the newly selected vehicle. Such a mechanism could be provided, e.g., by a dedicated base station, or by multi-hop communication.

In practice the stochastic nature of the proposed algorithm might lead to high traveling cost and long maneuver time. To mitigate this problem, a hybrid algorithm is developed by combining the Gibbs sampler-based method with the deterministic gradient-flow method. A vehicle switches between the two schemes to achieve a sound tradeoff between efficiency and optimality. Analysis is performed on the choice of switching parameters. The impact of vehicle memory on performance enhancement is also investigated.

The remainder of the paper is organized as follows. The concept of MRFs and the problem setup are described in

Section 2. The single-vehicle case is treated in Section 3, while the multi-vehicle case studied in Section 4. In Section 5 the hybrid algorithm is presented. Section 6 concludes the paper.

2. Problem setup

2.1. MRFs and Gibbs sampler

One can refer to, e.g., Winkler (1995), Bremaud (1999), for a review of MRFs. Let S be a finite set of cardinality σ , with elements indexed by s and called *sites*. For $s \in S$, let A_s be a finite set called the *phase space* for site s . A *random field* on S is a collection $X = \{X_s\}_{s \in S}$ of random variables X_s taking values in A_s . A *configuration* of the system is $x = \{x_s, s \in S\}$, where $x_s \in A_s, \forall s$. The product space $\Lambda \triangleq A_1 \times \cdots \times A_\sigma$ is called the *configuration space*. A *neighborhood system* on S is a family $\Gamma = \{\Gamma_s\}_{s \in S}$, where $\forall s, r \in S$,

- $\Gamma_s \subset S$,
- $s \notin \Gamma_s$, and
- $r \in \Gamma_s$ if and only if $s \in \Gamma_r$.

Γ_s is called the *neighborhood* of site s . The random field X is called an MRF with respect to the neighborhood system Γ if, $\forall s \in S, P(X_s = x_s | X_r = x_r, r \neq s) = P(X_s = x_s | X_r = x_r, r \in \Gamma_s)$.

A random field X is a *Gibbs random field* if and only if it has the Gibbs distribution:

$$P(X = x) = \frac{e^{-\frac{U(x)}{T}}}{Z}, \quad \forall x \in \Lambda,$$

where T is the temperature variable, $U(x)$ is the potential for configuration x , and Z is the *partition function*: $Z = \sum_x e^{-\frac{U(x)}{T}}$. The Hammersley–Clifford theorem (Bremaud, 1999) establishes the equivalence of a Gibbs random field and an MRF.

The *Gibbs sampler* belongs to the class of *Markov Chain Monte Carlo* (MCMC) methods, which sample Markov chains leading to stationary distributions. The algorithm updates the configuration by visiting sites sequentially or randomly with certain *proposal* distribution (Winkler, 1995), and sampling from the local specifications of a Gibbs field. A *sweep* refers to one round of sequential visits to all sites, or σ random visits under the proposal distribution. The convergence of the Gibbs sampler was studied by Geman and Geman (1984) in the context of image processing. There it was shown that as the number of sweeps goes to infinity, the distribution of $X(n)$ converges to the Gibbs distribution. Furthermore, with an appropriate cooling schedule, simulated annealing using the Gibbs sampler yields a uniform distribution on the set of minimizers of $U(x)$. Thus the global objectives could be achieved through appropriate design of the Gibbs potential function.

2.2. Problem setup for multi-vehicle coordination

Consider a 2D mission space (the extension to 3D space is straightforward), which is discretized into a lattice of cells.

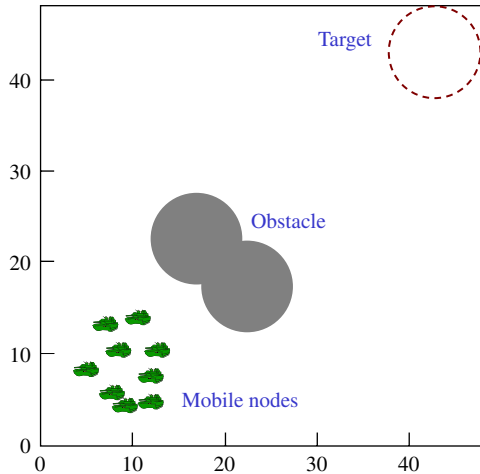


Fig. 1. An example mission scenario with a circular target and a nonconvex obstacle (formed by two overlapping circular obstacles). Since the mission space is a discretized grid, a cell is taken to be within a disk if its center is so.

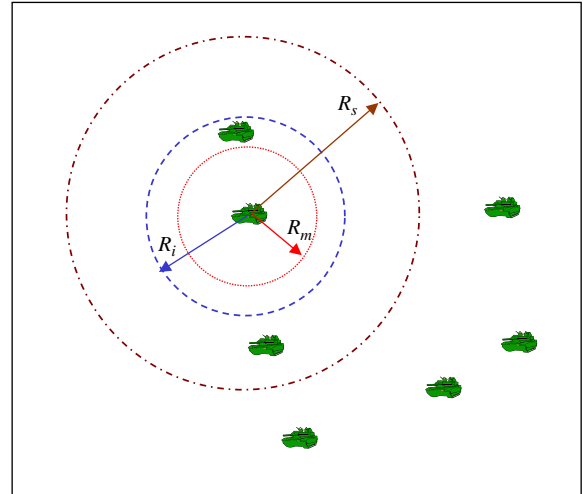


Fig. 2. Illustration of the sensing range R_s , the interaction range R_i , and the moving range R_m .

For ease of presentation, each cell is assumed to be square with unit dimensions. One could of course define cells of other geometries (e.g., hexagons) and of other dimensions (related to the coarseness of the grid) depending on the problems at hand. Label each cell with its coordinates (i, j) , where $1 \leq i \leq N_1$, $1 \leq j \leq N_2$, for $N_1, N_2 > 0$. There is a set S of vehicles (or *mobile nodes*) indexed by $s = 1, \dots, \sigma$ on the mission space. To be precise, each vehicle s is assumed to be a point located at the center of some cell (i_s, j_s) , and the position of vehicle s is taken to be $p_s = (i_s, j_s)$. It is further assumed that at most one vehicle is allowed to stay in each cell at any time instant. This assumption is motivated by several possible considerations, such as maintaining safe inter-vehicle distance or avoiding redundant vehicle coverage (for maximum coverage efficiency).

The distance R between two cells, (i_a, j_a) and (i_b, j_b) , is $R \triangleq \|(i_a, j_a) - (i_b, j_b)\| = \sqrt{(i_a - i_b)^2 + (j_a - j_b)^2}$. There might be multiple *obstacles* in the space, where an obstacle is defined to be a set of adjacent cells that are inaccessible to vehicles. For instance, a “circular” obstacle centered at (i^0, j^0) with radius R^0 can be defined as $O \triangleq \{(i, j) : \sqrt{(i - i^0)^2 + (j - j^0)^2} \leq R^0\}$. The *accessible area* is the set of cells in the mission space that are not occupied by obstacles. An *accessible-area graph* can then be defined by letting each cell in the accessible area be a vertex and connecting neighboring cells with edges. The mission space is *connected* if the associated accessible-area graph is connected, which will be assumed in this paper. There can be *target areas* in the space. A target area is a set of adjacent cells that represent desirable destinations of mobile nodes. A “circular” target area can be defined similarly as a “circular” obstacle. An example mission scenario is shown in Fig. 1.

In this paper all vehicles are assumed to be identical. Each vehicle has a *sensing range* R_s : it can “see” whether a cell within distance R_s is occupied by some node or obstacle through sensing or direct inter-vehicle communication. There is also an

interaction range R_i ($R_i \leq R_s$) for vehicles: for a vehicle s , only vehicles within a distance of R_i can impact its moving decision. These nodes form the set Γ_s of *neighbors* of node s . The distinction between R_s and R_i is made mainly for the convenience in later analysis, but it also reflects the scenarios where decision complexity forces each vehicle to consider only a subset of vehicles in its sensing range. We further assume that a node can travel at most R_m ($R_m \leq R_s$), called *moving range*, within one move. See Fig. 2 for illustration of the three range definitions.

The neighborhood system defined earlier naturally leads to a dynamic graph, where each vehicle represents a vertex of the graph and the neighborhood relation prescribes the edges between vehicles. An MRF can then be defined on the graph, where each vehicle s is a site and the associated phase space A_s is the set of all cells located within the moving range R_m from location p_s and not occupied by obstacles or other vehicles. The configuration space of the MRF is denoted as \mathcal{X} .

The Gibbs potential $U(x)$ takes the form:

$$U(x) = \sum_s \Phi_s(x), \tag{1}$$

where $\Phi_s(x)$ depends only on x_s and $\{x_r, r \in \Gamma_s\}$. Moreover, the individual potential $\Phi_s(x)$ consists of three terms with each reflecting one goal or one constraint:

$$\Phi_s(x) = \lambda_g J_g(x_s) + \lambda_o J_o(x_s) + \lambda_n J_n(x_s, x_{\Gamma_s}), \tag{2}$$

where x_{Γ_s} denotes the locations of nodes in Γ_s for configuration x . In (2), $J_g(x_s)$, $J_o(x_s)$, and $J_n(x_s, x_{\Gamma_s})$ account for the attraction from the target areas, the repelling from obstacles, and the interaction between neighbors, respectively, and $\lambda_g, \lambda_o, \lambda_n$ are the corresponding weighting coefficients for adjusting the potential surface. Note that the design of these constants is also a challenging and important issue as it may directly impact the behaviors of nodes and the convergence rates of algorithms.

J_n is assumed to consist of pairwise terms:

$$J_n(x_s, x_{T_s}) = \sum_{r \in \Gamma_s} \bar{J}_n(x_s, x_r). \quad (3)$$

There are important differences between a classical MRF introduced in Section 2.1 and the MRF defined for the vehicle networks. In a classical MRF, both the phase space A_s and the neighborhood Γ_s are time-invariant; however, for a vehicle network, both A_s and Γ_s depend on the dynamic graph and vary with time. This prevents the classical MRF theory from being adopted directly for convergence analysis in the context of this paper.

3. The single-vehicle case

In the single-vehicle case, $\sigma=1$ and the potential term involving vehicle interactions vanishes. The configuration x degenerates to the location of the vehicle, and the goal is to maneuver the vehicle to the target (or any other location with the minimal potential) in the presence of arbitrarily shaped obstacles. The algorithm is described and its convergence established in Section 3.1, and the impact of Gibbs potential design on the convergence speed is explored in Section 3.2.

3.1. The annealing algorithm and its convergence

Pick a cooling schedule $T(n)$ with $T(n) \rightarrow 0$ as $n \rightarrow \infty$, an integer $\tau \geq 1$, and a sufficiently large N_{\max} . The algorithm works as follows:

- *Step 1: Initialization.* Let $n = 1, k = 1$;
- *Step 2: Gibbs sampling.* Based on the current location x , determine the set \mathcal{N}_m^x of candidate locations for the next move: $\mathcal{N}_m^x \triangleq \{z : \|z - x\| \leq R_m, z \notin \text{any obstacle}\}$, where $\|\cdot\|$ denotes the Euclidean norm. Update x to $y \in \mathcal{N}_m^x$ with local specifications

$$P(y|x) = \frac{e^{-\frac{U(y)}{T(n)}}}{\sum_{z \in \mathcal{N}_m^x} e^{-\frac{U(z)}{T(n)}}}.$$

Let $k = k + 1$. If $k = \tau$, let $k = 0$, and $n = n + 1$;

- *Step 3:* If $n < N_{\max}$, go to Step 2; quit otherwise.

Some results on classical MRFs (Winkler, 1995) can be adapted to establish the convergence of the above algorithm. One can first show that there is a unique stationary distribution Π_T for a fixed temperature T :

Theorem 3.1. *Assume that the accessible-area graph is connected. For a fixed temperature T , let P_T be the kernel of the homogeneous Markov chain $X(n)$ defined by the Gibbs sampler. Then $X(n)$ has a unique stationary distribution Π_T :*

$$\Pi_T(x) = \frac{e^{-\frac{U(x)}{T}} \sum_{z \in \mathcal{N}_m^x} e^{-\frac{U(z)}{T}}}{Z_T}, \quad (4)$$

where $Z_T = \sum_{y \in \mathcal{X}} (e^{-\frac{U(y)}{T}} \sum_{z \in \mathcal{N}_m^y} e^{-\frac{U(z)}{T}})$. Furthermore, starting from any distribution ν ,

$$\lim_{n \rightarrow \infty} \nu P_T^n = \Pi_T. \quad (5)$$

Proof. Since the MRF has only one site, the Markov kernel P_T defined by the Gibbs sampler is

$$P_T(x, y) \triangleq P(X(n+1) = y | X(n) = x) = \begin{cases} \frac{e^{-\frac{U(y)}{T}}}{\sum_{z \in \mathcal{N}_m^x} e^{-\frac{U(z)}{T}}} & \text{if } y \in \mathcal{N}_m^x, \\ 0 & \text{if } y \notin \mathcal{N}_m^x. \end{cases} \quad (6)$$

Due to the connectivity of the accessible area, there exists at least one path between any two configurations x and y (i.e., a sequence of moves $\{x, x_1, \dots, y\}$), and the shortest path is bounded above by τ moves for some finite τ . This implies that P_T has a strictly positive power P_T^τ . Hence the Markov chain is ergodic (Horn & Johnson, 1985) and it has a unique invariant distribution Π_T , which implies (5). One can verify that (4) is a stationary distribution for P_T . \square

With a proper choice of $T(n)$ and τ , the simulated annealing algorithm yields the desired configuration:

Theorem 3.2. *Assume that the accessible area is connected. Pick τ as in the proof of Theorem 3.1. Let $T(n)$ be a cooling schedule decreasing to 0 such that eventually*

$$T(n) \geq \frac{\tau \Delta}{\ln n}, \quad (7)$$

where $\Delta = \max_{x,y} \{|U(x) - U(y)| : y \in \mathcal{N}_m^x\}$. Let $Q_n = P_{T(n)}^\tau$. Then from any initial distribution ν ,

$$\lim_{n \rightarrow \infty} \nu Q_1 \cdots Q_n = \Pi_\infty, \quad (8)$$

where Π_∞ is the distribution (4) at $T = 0$. Let \mathcal{M} be the set of minimizers of $U(x)$. Then Π_∞ satisfies

$$\sum_{x \in \mathcal{M}} \Pi_\infty(x) = 1. \quad (9)$$

Recall that for a Markov kernel P , its contraction coefficient $c(P)$ is defined as

$$c(P) = \left(\frac{1}{2}\right) \max_{x,y} \|P(x, \cdot) - P(y, \cdot)\|_1,$$

where $P(x, \cdot)$ denotes the vector of conditional distributions $p(\cdot|x)$, and $\|\cdot\|_1$ denotes the 1-norm. The following lemma will be useful in the proof of Theorem 3.2.

Lemma 3.1 (Winkler, 1995). *Let μ and ν be probability distributions, and P and Q be Markov kernels. Then*

$$\begin{aligned} \|\mu P - \nu P\|_1 &\leq c(P) \|\mu - \nu\|_1, \\ c(PQ) &\leq c(P)c(Q), \\ c(P) &\leq 1 - |\mathcal{X}| \min\{P(x, y) : x, y \in \mathcal{X}\}, \end{aligned}$$

and for a primitive P (i.e., all entries of P^τ are positive for sufficiently large τ),

$$c(P^n) \rightarrow 0 \quad \text{as } n \rightarrow \infty,$$

where \mathcal{X} is the state space of the Markov chain and $|\mathcal{X}|$ denotes its cardinality.

Proof of Theorem 3.2. Let $m_x \triangleq \min\{U(z) : z \in \mathcal{N}_m^x\}$. For $y \in \mathcal{N}_m^x$,

$$P_T(x, y) = \frac{e^{-\frac{U(y)-mx}{T}}}{\sum_{z \in \mathcal{N}_m^x} e^{-\frac{U(z)-mx}{T}}} \geq |\bar{A}|^{-1} e^{-\frac{\Delta}{T}}, \quad (10)$$

where $|\bar{A}|$ is the maximal cardinality of the phase space. For $Q = P_T^\tau$, from Lemma 3.1 and (10),

$$\begin{aligned} c(Q) &\leq 1 - |\mathcal{X}| \min_{x,y} Q(x, y) \leq 1 - |\mathcal{X}| \left(\min_{x \in \mathcal{X}, y \in \mathcal{N}_m^x} P_T(x, y) \right)^\tau \\ &\leq 1 - \lambda e^{-\frac{\tau \Delta}{T}}, \end{aligned} \quad (11)$$

where $\lambda = |\mathcal{X}| \cdot |\bar{A}|^{-\tau} \leq 1$. This implies, for $Q_k = P_{T(k)}^\tau$,

$$c(Q_k) \leq 1 - \lambda e^{-\frac{\tau \Delta}{T(k)}}.$$

Then for $i \leq n$,

$$\prod_{k=i}^n c(Q_k) \leq \prod_{k=i}^n (1 - \lambda e^{-\frac{\tau \Delta}{T(k)}}) \leq \prod_{k=i}^n (1 - \lambda k^{-1}) \quad (12)$$

$$\leq e^{-\sum_{k=i}^n \lambda k^{-1}} \quad (13)$$

$$\leq \left(\frac{i}{n}\right)^\lambda, \quad (14)$$

where (12) is from (7), (13) is from

$$(1-r) \leq e^{-r}, \quad \forall r \in (0, 1),$$

and (14) is from

$$\begin{aligned} \ln(ni^{-1}) &< \ln(n+1) - \ln(i) = \sum_{k=i}^n (\ln(1+k) - \ln k) \\ &= \sum_{k=i}^n \ln(1+k^{-1}) \leq \sum_{k=i}^n k^{-1}. \end{aligned}$$

With some abuse of notation, $\Pi_{T(n)}$ will be written as Π_n to simplify the expressions. For $i \leq n$, one has

$$\begin{aligned} &\|vQ_1 \dots Q_n - \Pi_\infty\|_1 \\ &= \|(vQ_1 \dots Q_{i-1} - \Pi_\infty)Q_i \dots Q_n + \Pi_\infty Q_i \dots Q_n - \Pi_\infty\|_1 \\ &\leq \|(vQ_1 \dots Q_{i-1} - \Pi_\infty)\|_1 c(Q_i \dots Q_n) \\ &\quad + \|\Pi_\infty Q_i \dots Q_n - \Pi_\infty\|_1 \\ &\leq 2c(Q_i \dots Q_n) + \|\Pi_i - \Pi_\infty\|_1 + \|\Pi_n - \Pi_\infty\|_1 \\ &\quad + \sum_{j=i}^{j=n-1} \|\Pi_j - \Pi_{j+1}\|_1, \end{aligned} \quad (15)$$

where the last inequality is established by Dobrushin's theorem (Winkler, 1995). When $n, i \rightarrow \infty$ and $\frac{n}{i} \rightarrow \infty$, the first term in (15) approaches 0 by (14), and the second and third terms also approach 0 by the definition of Π_∞ . To show that the fourth term vanishes as $i \rightarrow \infty$, it suffices to prove each component sequence of $\{\Pi_n(x)\}_{n \geq 1}, \forall x \in \mathcal{X}$, decreases or increases eventually. By investigating the derivative of $\Pi_n(x)$ with respect to the temperature T , the ultimate monotonicity can be established, as was done in Winkler (1995). This completes the proof of (8).

Let $m \triangleq \min_{x \in \mathcal{X}}$. One has

$$\Pi_T(x) = \frac{e^{-\frac{U(x)-m}{T}} \sum_{z \in \mathcal{N}_m^x} e^{-\frac{U(z)-m}{T}}}{\sum_{y \in \mathcal{X}} (e^{-\frac{U(y)-m}{T}} \sum_{z \in \mathcal{N}_m^y} e^{-\frac{U(z)-m}{T}})}.$$

If $U(x) \notin \mathcal{M}$, $e^{-\frac{U(x)-m}{T}} \rightarrow 0$ and thus $\Pi_T(x) \rightarrow 0$ as $T \rightarrow 0$, which implies (9). \square

Assume that there is a unique target area with minimum potential. Theorem 3.2 then implies that under the proposed algorithm, a vehicle can reach the target ultimately for *arbitrarily shaped* obstacles.

3.2. Impact of potential function on convergence speed

It is of interested to study how to improve the convergence speed by appropriately designing the potential function. The following result connects the potential function to the convergence speed.

Proposition 3.1. Let the cooling schedule $T(n) = \frac{\tau \Delta}{\ln n}$. The convergence speed in Theorem 3.2 is characterized by

$$\|vQ_1 \dots Q_n - \Pi_\infty\|_1 = O(n^{-\frac{\lambda \delta_m}{\delta_m + \lambda \tau \Delta}}), \quad (16)$$

where $\delta_m = \min_{y \notin \mathcal{M}} (U(y) - m)$, and m, λ, Δ , and τ are defined as in Theorem 3.2 or in its proof.

Proof. From (15), $\|vQ_1 \dots Q_n - \Pi_\infty\|_1$ is bounded by four terms. From (14), the first term is bounded by $(\frac{i}{n})^\lambda$. The other terms are each bounded by $\|\Pi_i - \Pi_\infty\|_1$, considering the ultimate monotonicity of the sequence $\{\Pi_n\}$.

If x is not a minimizer of U ,

$$e^{-\frac{U(x)-m}{T(i)}} = e^{-\frac{(U(x)-m) \ln i}{\tau \Delta}} \leq e^{-\frac{\delta_m \ln i}{\tau \Delta}} = i^{-\frac{\delta_m}{\tau \Delta}},$$

which implies

$$\begin{aligned} |\Pi_i(x) - \Pi_\infty(x)| &= \Pi_i(x) \\ &= \frac{e^{-\frac{U(x)-m}{T(i)}} \sum_{z \in \mathcal{N}_m^x} e^{-\frac{U(z)-m}{T(i)}}}{\sum_{y \in \mathcal{X}} (e^{-\frac{U(y)-m}{T(i)}} \sum_{z \in \mathcal{N}_m^y} e^{-\frac{U(z)-m}{T(i)}})} \\ &\leq \frac{i^{-\frac{\delta_m}{\tau \Delta}} \sum_{z \in \mathcal{N}_m^x} e^{-\frac{U(z)-m}{T(i)}}}{\sum_{y \in \mathcal{X}} (e^{-\frac{U(y)-m}{T(i)}} \sum_{z \in \mathcal{N}_m^y} e^{-\frac{U(z)-m}{T(i)}})}. \end{aligned}$$

Since $\sum_{z \in \mathcal{N}_m^x} e^{-\frac{U(z)-m}{T(i)}} \leq |\bar{A}|$, and

$$\begin{aligned} & \sum_{y \in \mathcal{X}} \left(e^{-\frac{U(y)-m}{T(i)}} \sum_{z \in \mathcal{N}_m^y} e^{-\frac{U(z)-m}{T(i)}} \right) \\ &= |\mathcal{M}| + \text{terms involving } e^{-\frac{U(z)-m}{T(i)}} \text{ with } z \notin \mathcal{M} \geq |\mathcal{M}|, \end{aligned}$$

one has

$$|\Pi_i(x) - \Pi_\infty(x)| \leq \frac{|\bar{A}| i^{-\frac{\delta_m}{\tau \Delta}}}{|\mathcal{M}|} = O(i^{-\frac{\delta_m}{\tau \Delta}}).$$

Now if x is a minimizer of U (i.e., $x \in \mathcal{M}$), $\Pi_\infty(x) \neq 0$. Assuming first $z \notin \mathcal{N}_m^y \forall y, z \in \mathcal{M}$, one can show $\Pi_i(x) = \frac{1}{|\mathcal{M}|}$ using Theorem 3.2. This implies

$$\begin{aligned} & |\Pi_i(x) - \Pi_\infty(x)| \\ &= \left| \frac{\sum_{z \in \mathcal{N}_m^x} e^{-\frac{U(z)-m}{T(i)}}}{\sum_{y \in \mathcal{X}} \left(e^{-\frac{U(y)-m}{T(i)}} \sum_{z \in \mathcal{N}_m^y} e^{-\frac{U(z)-m}{T(i)}} \right)} - \frac{1}{|\mathcal{M}|} \right| \\ &= \left| \frac{1+b}{|\mathcal{M}|+a} - \frac{1}{|\mathcal{M}|} \right| = \frac{|b|\mathcal{M}| - a|}{|\mathcal{M}|(|\mathcal{M}|+a)} \\ &\leq \frac{|b|\mathcal{M}| - a|}{|\mathcal{M}|^2} \leq \frac{b|\mathcal{M}| + a}{|\mathcal{M}|^2}, \end{aligned}$$

where a and b are terms involving $e^{-\frac{U(z)-m}{T(i)}}$ with $z \notin \mathcal{M}$. Since a and b are both of order $O(i^{-\frac{\delta_m}{\tau \Delta}})$, so is $|\Pi_i(x) - \Pi_\infty(x)|$. Similarly, one can show for more general cases involving multiple minimizers within R_m , $|\Pi_i(x) - \Pi_\infty(x)| = O(i^{-\frac{\delta_m}{\tau \Delta}})$, $\forall x \in \mathcal{M}$.

Thus a bound for $\|vQ_1 \dots Q_n - \Pi_\infty\|_1$ is

$$\left(\frac{i}{n}\right)^\lambda + \text{const} \cdot i^{-\frac{\delta_m}{\tau \Delta}}. \quad (17)$$

This becomes minimal for

$$\begin{aligned} i^* &= \left(\text{const} \cdot \frac{\delta_m}{\tau \Delta} \right)^{\frac{1}{\lambda + \frac{\delta_m}{\tau \Delta}}} \cdot n^{\frac{\lambda}{\lambda + \frac{\delta_m}{\tau \Delta}}} \\ &= \text{const} \cdot n^{\frac{\lambda}{\lambda + \frac{\delta_m}{\tau \Delta}}}. \end{aligned}$$

Eq. (16) then follows by plugging i^* into (17). \square

Proposition 3.1 shows that the potential surface (in particular: $g \triangleq \frac{\lambda \delta_m}{\delta_m + \lambda \tau \Delta}$) determines the convergence speed of the algorithm. It is thus natural to use g as a design indicator. Simulation was conducted to verify the above analysis. A similar scenario as in Fig. 1 was used but with a 10 by 10 grid. A single vehicle starts from the lower left corner, and wants to reach the target area at the upper right corner. The potential function used was

$$U(x) = \lambda_g \|x - p^g\| + \lambda_0 \sum_{k=1}^2 \frac{1}{\|x - p^{ok}\|}, \quad (18)$$

where p^g and p^{ok} denote the centers of the circular target area and of the circular obstacles. In the simulation λ_g was varied from 0.05 to 100 while λ_0 was fixed to 1. For each pair of

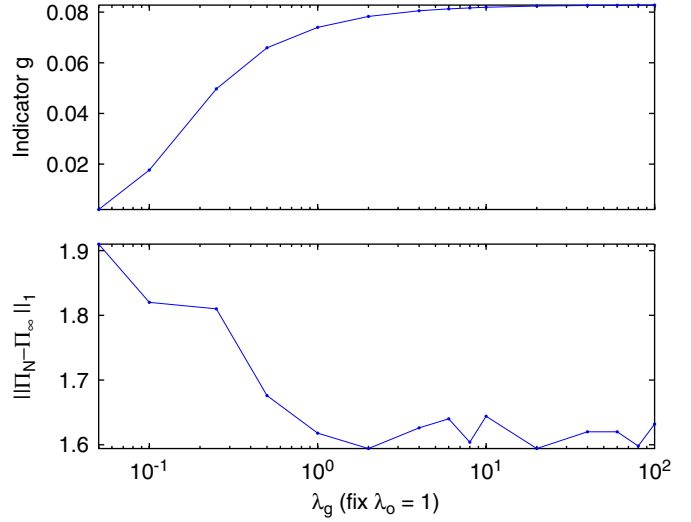


Fig. 3. The correlation of computed convergence speed with the design indicator g .

coefficients, the algorithm was run $N = 10^4$ steps, and the number of times w that the vehicle visited target during the last 100 steps was counted. The empirical distance was calculated by $\|\Pi_N - \Pi_\infty\|_1 = 2(1 - w/100)$. The variation of $\|\Pi_N - \Pi_\infty\|_1$ with respect to λ_g is consistent with that of the design indicator g (Fig. 3), which confirms the analysis (16).

4. The multi-vehicle case

4.1. The coordination algorithm

The coordination algorithm for multiple vehicles uses a randomized sequence for updating the nodes, and a key idea involved is the *configuration- and temperature-dependent* proposal distribution $G_T^x(s)$. Here s is the index of vehicles, T denotes the temperature, and x denotes a configuration, which specifies the locations of all vehicles. Given x and T , a vehicle s is selected with probability $G_T^x(s)$ for update, where

$$G_T^x(s) = \frac{\sum_{z \in \mathcal{N}_m^x(s)} e^{-\frac{U(z)}{T}}}{\sum_{s' \in \mathcal{S}} \sum_{z \in \mathcal{N}_m^x(s')} e^{-\frac{U(z)}{T}}} \quad (19)$$

$$= \frac{\sum_{z \in \mathcal{N}_m^x(s)} e^{-\frac{U(z)-U(x)}{T}}}{\sum_{s' \in \mathcal{S}} \sum_{z \in \mathcal{N}_m^x(s')} e^{-\frac{U(z)-U(x)}{T}}}. \quad (20)$$

In (19), $\mathcal{N}_m^x(s)$ denotes the set of s -neighbors of configuration x within one move:

$$\mathcal{N}_m^x(s) \triangleq \{z \in \mathcal{X} : z_{S \setminus s} = x_{S \setminus s}, \|z_s - x_s\| \leq R_m\}, \quad (21)$$

where $S \setminus s$ is the set of all nodes except s , and $z_{S \setminus s}$ is the locations of nodes in $S \setminus s$ for configuration z . Due to the form of Gibbs potential (see (1)–(3)), a node s will be able to evaluate, for $z \in \mathcal{N}_m^x(s)$, the potential difference $U(z) - U(x)$ and thus the sum $D_T^x(s) \triangleq \sum_{z \in \mathcal{N}_m^x(s)} e^{-\frac{U(z)-U(x)}{T}}$ when $R_s \geq R_i + R_m$. Roughly speaking, $D_T^x(s)$ measures the reduction of potential

function if node s is selected for update. The proposal function $G_T^x(s)$ in (19) implies that a node s gets a higher chance for update than a node r if $D_T^x(s) > D_T^x(r)$.

In a two-step sampling procedure, node s is first randomly selected with probability $G_T^x(s)$, and then x_s is updated according to its local characteristics while $x_{S \setminus s}$ is kept fixed: for $l \in C_m^s$,

$$P(x_s = l) = \frac{e^{-\frac{U(x_s=l, x_{S \setminus s}) - U(x)}{T}}}{\sum_{l' \in C_m^s} e^{-\frac{U(x_s=l', x_{S \setminus s}) - U(x)}{T}}}, \quad (22)$$

where C_m^s is the set of candidate locations node s can take, i.e., $l \in C_m^s$ is not occupied by any obstacle or other nodes, and $\|x_s - l\| \leq R_m$. Note that, again, the evaluation of (22) can be done locally by node s when $R_s \geq R_i + R_m$. One can verify that there exists a smallest integer τ , such that after τ steps of sampling, any configuration x has a positive probability of becoming any other configuration y .

The coordination algorithm is presented next. Pick an appropriate cooling schedule $T(n)$ with $T(n) \rightarrow 0$ as $n \rightarrow \infty$. Pick N_{\max} . For each $T(n)$, run τ steps of sampling as described above (called *one annealing step*).

- *Step 1: Initialization.* Start with an arbitrary configuration $x(0)$ and let $n = 1, k = 1$. Pick an arbitrary node $s(0)$. Have all nodes to evaluate and send $D_{T(1)}^{x(0)}(s)$ to $s(0)$. Node $s(0)$ calculates the proposal distribution $G_{T(1)}^{x(0)}(s)$ according to (19), and then selects a node $s^1(1)$ ¹ for update by sampling $G_{T(1)}^{x(0)}(s)$, and sends the vector $\{D_{T(1)}^{x(0)}(s), s \in S\}$ to $s^1(1)$;
- *Step 2: Updating the selected node.* Node $s^k(n)$ updates its location by sampling its local characteristics (see (22)). Denote the new configuration as $x^k(n)$;
- *Step 3: Selecting the next node.* Note that the neighborhood Γ_s of a node s changes *only* if node $s^k(n)$ was in Γ_s before its updating or is currently in Γ_s . For either case, the distance between such s (denoting the set of such nodes as $\bar{\Gamma}^k(n)$) and $s^k(n)$ is now no greater than $R_i + R_m \leq R_s$ and they can communicate *locally*. The node $s^k(n)$ thus collects and updates $D_{T(n)}^{x^k(n)}(s)$ for nodes in $\bar{\Gamma}^k(n)$. Let $k = k + 1$. If $k = \tau$, let $k = 0$ and $n = n + 1$. The current node evaluates and samples new proposal distribution, selects the next node to be updated, and communicates the updated $\{D_{T(n)}(s)\}$ to the next node (the superscript of D is omitted when it is clear from the context);
- *Step 4:* If $n < N_{\max}$, go to Step 2; otherwise quit.

Remark 4.1. Long-range (over a distance greater than R_s) communication is only required for initialization and for transferring $\{D_T^x(s)\}$ to the newly selected node. Since $\{D_T^x(s)\}$ is just a σ -dimensional vector, information exchange in the algorithm is primarily at the local level.

¹ In the notation $x^k(n)$ or $s^k(n)$, n indexes the annealing temperature, while k (from 1 to τ) indexes the sampling step within a fixed temperature.

4.2. Convergence analysis

Let P_T be the Markov kernel defined by the update (19) and (22), i.e.,

$$\begin{aligned} P_T(x, y) &= \sum_{s \in S} G_T^x(s) \cdot \mathbf{1}(y \in \mathcal{N}_m^x(s)) \frac{e^{-\frac{U(y)}{T}}}{\sum_{z \in \mathcal{N}_m^x(s)} e^{-\frac{U(z)}{T}}} \\ &= \frac{\sum_{s \in S} \sum_{z \in \mathcal{N}_m^x(s)} e^{-\frac{U(z)}{T}} \cdot e^{-\frac{U(y)}{T}} \cdot \mathbf{1}(y \in \mathcal{N}_m^x(s))}{\sum_{s' \in S} \sum_{z \in \mathcal{N}_m^{x'}(s')} e^{-\frac{U(z)}{T}} \cdot \sum_{z \in \mathcal{N}_m^{x'}(s')} e^{-\frac{U(z)}{T}}} \\ &= \sum_{s \in S} \frac{e^{-\frac{U(y)}{T}} \cdot \mathbf{1}(y \in \mathcal{N}_m^x(s))}{\sum_{s' \in S} \sum_{z \in \mathcal{N}_m^{x'}(s')} e^{-\frac{U(z)}{T}}}. \end{aligned} \quad (23)$$

Theorem 4.1. Let $Q_T = P_T^\tau$. Then Q_T has a unique stationary distribution Π_T with

$$\Pi_T(x) = \frac{e^{-\frac{U(x)}{T}} \sum_{s \in S} \sum_{z \in \mathcal{N}_m^x(s)} e^{-\frac{U(z)}{T}}}{Z_T}, \quad (24)$$

where $Z_T = \sum_y e^{-\frac{U(y)}{T}} \sum_{s \in S} \sum_{z \in \mathcal{N}_m^y(s)} e^{-\frac{U(z)}{T}}$.

Proof. First one can show that Π_T is a stationary distribution of P_T . From (23) and (24),

$$\begin{aligned} &\sum_y \Pi_T(y) P_T(y, x) \\ &= \sum_y \frac{e^{-\frac{U(y)}{T}} \sum_{s' \in S} \sum_{z \in \mathcal{N}_m^{y'}(s')} e^{-\frac{U(z)}{T}}}{Z_T} \\ &\quad \times \sum_{s \in S} \frac{e^{-\frac{U(x)}{T}} \cdot \mathbf{1}(x \in \mathcal{N}_m^y(s))}{\sum_{s' \in S} \sum_{z \in \mathcal{N}_m^{y'}(s')} e^{-\frac{U(z)}{T}}} \\ &= \frac{e^{-\frac{U(x)}{T}} \sum_y e^{-\frac{U(y)}{T}} \sum_{s \in S} \mathbf{1}(x \in \mathcal{N}_m^y(s))}{Z_T} \\ &= \frac{e^{-\frac{U(x)}{T}} \sum_{s \in S} \sum_{z \in \mathcal{N}_m^x(s)} e^{-\frac{U(z)}{T}}}{Z_T} = \Pi_T(x). \end{aligned}$$

Since $Q_T = P_T^\tau$, Π_T is also a stationary distribution for Q_T . Due to the choice of τ , $Q_T(x, y) > 0, \forall x, y$. Thus, from the Perron–Frobenius theorem, Q_T has a unique stationary distribution, which is Π_T . \square

Let Δ be the maximal local oscillation of the potential U :

$$\Delta \triangleq \max_x \max_{y \in \mathcal{N}_m^x} |U(x) - U(y)|,$$

where $\mathcal{N}_m^x = \bigcup_{s \in S} \mathcal{N}_m^x(s)$.

Theorem 4.2. Let $T(n)$ be a cooling schedule decreasing to 0 such that eventually, $T(n) \geq \frac{\tau \Delta}{\ln n}$. Let $Q_n = P_{T(n)}^\tau$, and let \mathcal{M} be the set of global minima of $U(\cdot)$. Then for any initial distribution ν ,

$$\lim_{n \rightarrow \infty} \nu Q_1 \cdots Q_n \rightarrow \Pi_\infty, \quad (25)$$

where Π_∞ is the distribution (24) evaluated at $T = 0$. In particular,

$$\sum_{x \in \mathcal{M}} \Pi_\infty(x) = 1. \tag{26}$$

Proof. Let $\alpha_x = \min_{z \in \mathcal{N}_m^x} U(z)$. For $y \in \mathcal{N}_m^x$, from (23),

$$P_T(x, y) = \frac{e^{-\frac{U(y)-\alpha_x}{T}}}{\sum_{s' \in \mathcal{S}} \sum_{z \in \mathcal{N}_m^x(s')} e^{-\frac{U(z)-\alpha_x}{T}}} \geq \frac{e^{-\frac{A}{T}}}{\sigma|\mathcal{X}|},$$

where $|\mathcal{X}|$ denotes the cardinality of the configuration space \mathcal{X} . Following analogous arguments to those in the proof of Theorem 3.2, one can show

$$c(Q_n) \leq 1 - \lambda e^{-\frac{\tau A}{T(n)}},$$

where $c(Q_n)$ denotes the contraction coefficient of Q_n , and $\lambda = \frac{|\mathcal{X}|}{(\sigma|\mathcal{X}|)^\tau}$. Similarly, one can prove claim (25). From (24), as $T(n) \rightarrow 0$, $\Pi_{T(n)}(x) \rightarrow 0$, for all $x \notin \mathcal{M}$. Eq. (26) thus follows. \square

From Theorem 4.2, the proposed algorithm can achieve global objectives provided that the global minimizers of $U(\cdot)$ correspond to the desired configurations. Let $\delta_m = \min_{x \notin \mathcal{M}} U(x) - m$, i.e., the minimal potential difference between other configurations and the global minimizers. The following result characterizes the convergence speed of the multi-vehicle coordination algorithm. The proof is omitted due to its similarity to that of Proposition 3.1:

Proposition 4.1. Consider the coordination algorithm with $T(n) = \frac{\tau A}{\ln n}$. Then for any initial distribution v ,

$$\|vQ_1 \cdots Q_n - \Pi_\infty\| = O(n^{-\frac{\lambda \delta_m}{\delta_m + \lambda \tau A}}) = O(n^{-g}), \tag{27}$$

where λ is as defined in Theorem 4.2, and $g = \frac{\lambda \delta_m}{\delta_m + \lambda \tau A}$.

4.3. Simulation results

Simulation was conducted to verify the analysis. The emphasis was on scenarios involving inter-vehicle interactions (e.g., formation control). Two examples are presented, one on clustering and the other on formation control. Other objectives or constraints, such as target-approaching and obstacle avoidance, can be easily incorporated, as was done in the single-vehicle case.

4.3.1. Clustering

The goal is to cluster all the nodes without specifying a specific target area. This is more challenging than the case of having an explicit target, as the latter provides persistent attraction from a fixed location. The potential

$$U(x) = \sum_{r \neq s, \|x_r - x_s\| \leq R_i} -\frac{c}{\|x_r - x_s\|},$$

where $c > 0$. Clearly, the more neighbors each node has and the closer they are, the lower U . Simulation was performed for 50

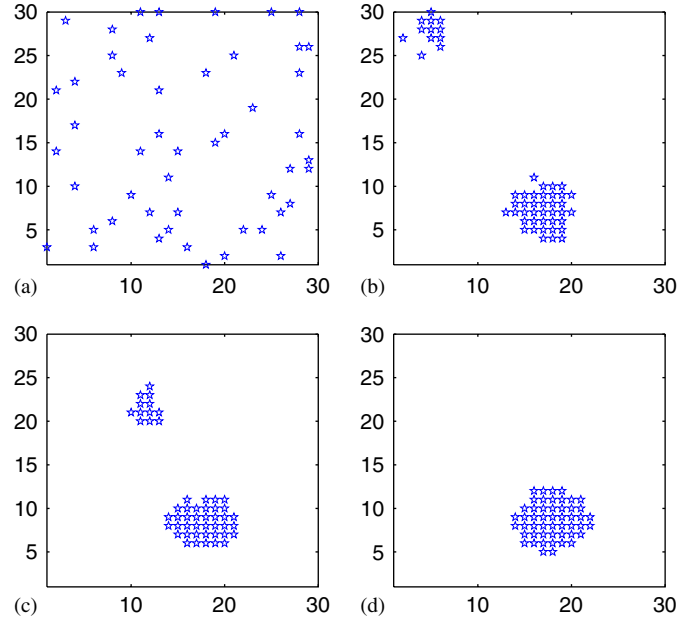


Fig. 4. Snapshots of clustering operation: (a) Initial configuration; (b) after 100 annealing steps; (c) after 400 annealing steps; (d) after 500 annealing steps.

nodes on a 30 by 30 grid with $R_I = 4\sqrt{2} + \varepsilon$ ($\varepsilon > 0$ and very small), $R_m = 2\sqrt{2} + \varepsilon$, $R_s = R_I + R_m$ (this was also true for other simulations in this section), $c = 2$, $T(n) = \frac{1}{0.08 \ln n}$, and $\tau = 50$.

Fig. 4 shows the snapshots of the network evolution. The algorithm’s ability to overcome local minima is evident from the figure: the nodes initially evolved into two separated (farther than R_s) sub-clusters, and yet they merged into one cluster after 500 annealing steps.

4.3.2. Formation

The goal is to have the nodes to form (square) lattice structures with a desired inter-vehicle distance R_{des} . The potential function used was

$$U(x) = \sum_{r \neq s, \|x_r - x_s\| \leq R_i} c_1 (\|x_r - x_s\| - R_{des})^\alpha - c_2,$$

where $c_1 > 0$, $c_2 > 0$, and $\alpha > 0$. A proper choice of c_2 encourages nodes to have more neighbors. The power α shapes the potential function. In particular, for $\|x_r - x_s\| - R_{des} < 1$, smaller α leads to larger potential difference from the global minimum.

Simulation was first conducted for 9 nodes on an 8 by 8 grid. Parameter used were: $R_i = 2\sqrt{2} - \varepsilon$, $R_m = \sqrt{2} + \varepsilon$, $R_{des} = 2$, $c_1 = 10$, $c_2 = 1.05$, $\alpha = 0.02$, $T(n) = \frac{1}{0.01 \ln n}$, and $\tau = 20$. The desired configuration (global minimizer of U) is shown in Fig. 5 (modulo vehicle permutation and formation translation on the grid). Simulated annealing was performed for 10^4 steps. Empirical distributions with respect to configuration potentials were calculated based on the average of every 2500 steps (Fig. 6). The trend of convergence to the configuration of the lowest potential is clear from Fig. 6. One can further calculate the error $\|v_n - \Pi_\infty\|_1$, where v_n is the empirical distribution of configurations (again modulo vehicle permutation

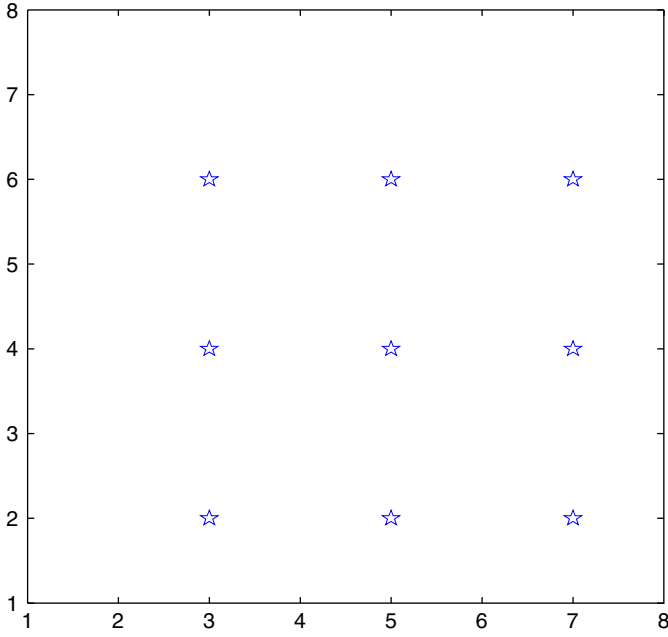


Fig. 5. The desired formation for 9 vehicles on an 8 by 8 grid.

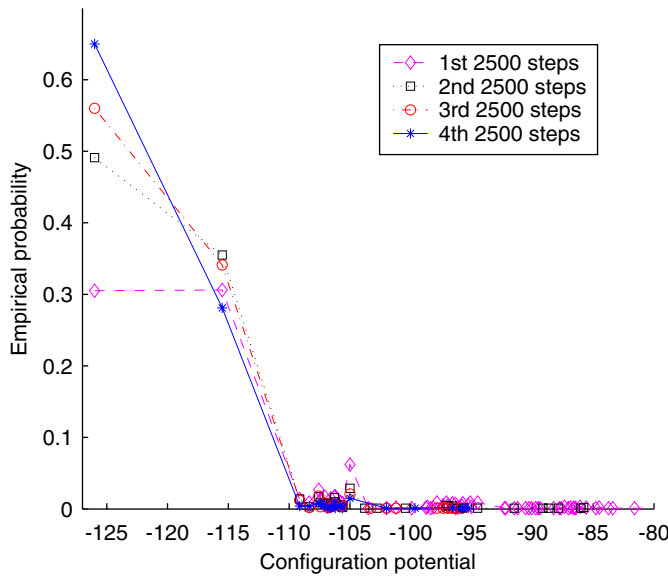


Fig. 6. Evolution of the empirical distribution of configuration potentials.

and network translation). In particular, v_n is only calculated for $n = 1000, 2000, 3000, \dots$, and for a configuration x , $v_n(x)$ equals the relative frequency of x between the $(n - 1000)$ th and the n th sampling steps. From Theorem 4.2,

$$\Pi_\infty(x) = \begin{cases} 1 & \text{if } x \text{ is desired,} \\ 0 & \text{otherwise,} \end{cases} \text{ implying}$$

$$\|v_n - \Pi_\infty\|_1 = 1 - v_n(x^*) + |0 - (1 - v_n(x^*))| = 2(1 - v_n(x^*)), \quad (28)$$

where x^* denotes the desired formation. The evolution of $\|v_n - \Pi_\infty\|_1$ is shown in Fig. 7.

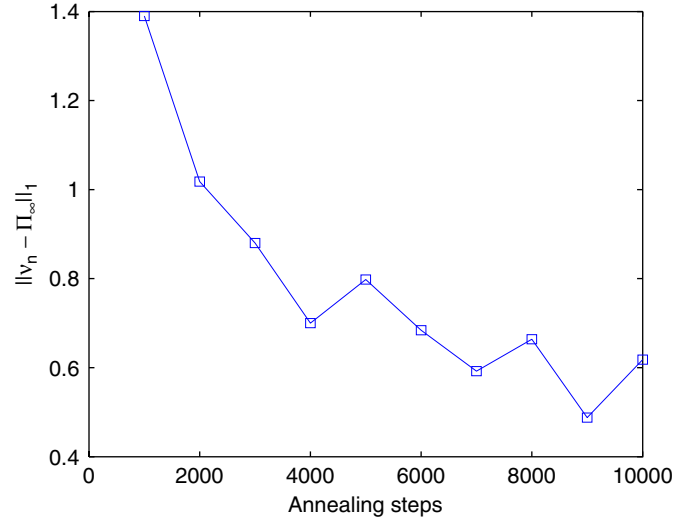


Fig. 7. Evolution of $\|v_n - \Pi_\infty\|_1$.

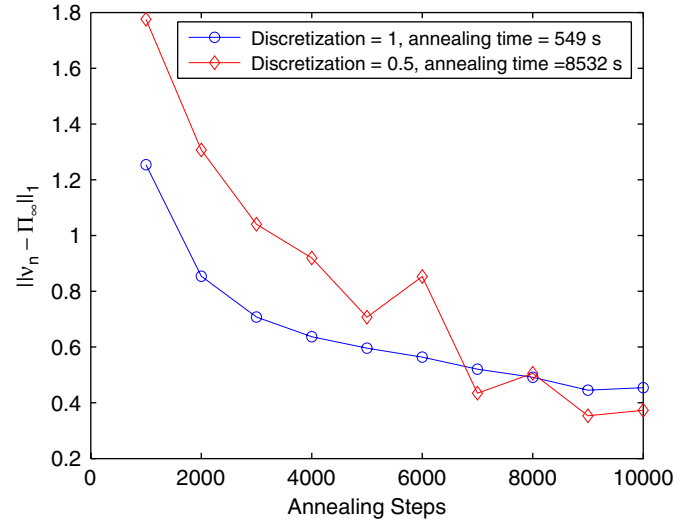


Fig. 8. Comparison of annealing schemes with different discretization levels.

It is of interest to investigate the robustness of the proposed scheme with respect to the discretization level of the grid. For this purpose, the same mission space was discretized into a 16 by 16 grid with cell size of 0.5, and simulation was performed based on the new lattice. All mission-related parameters were kept the same as in the 8 by 8 case. In particular, the exclusion condition (one cell cannot contain more than one vehicle) on the 8 by 8 grid was properly translated into a condition for the new grid. Five simulation runs were performed for both discretization schemes, and each was run for 10^4 annealing steps. For each simulation, the evolution of $\|v_n - \Pi_\infty\|_1$ was calculated according to (28) and the total annealing time was recorded. Fig. 8 compares the performance under the two discretization schemes, where $\|v_n - \Pi_\infty\|_1$ and the annealing time were averaged over the five runs for each case. The results indicate that the convergence to the desired formation can be achieved under both discretization

schemes, while it takes longer computational time for the finer discretization.

5. A hybrid coordination algorithm

The Gibbs sampler-based simulated annealing complements the deterministic gradient-flow method in that it could move vehicles out of otherwise trapping areas, but one has to pay the cost associated with probabilistic exploration—longer execution time and traveling distance. In addition, the algorithm is of sequential nature, i.e., vehicles move one at a time. The hybrid algorithm to be presented next aims to combine advantages of both the stochastic scheme and the gradient-flow scheme while avoiding their disadvantages.

5.1. The hybrid algorithm

To facilitate the discussion, a scenario illustrated in Fig. 1 is considered, and the mission is to maneuver the vehicles into the target while maintaining inter-vehicle interactions and avoiding obstacles. Hence all three times in (2) are present. The algorithm works as follows:

- *Step 1:* Each vehicle starts with the gradient-flow method and goes to Step 2;
- *Step 2:* If for d consecutive time steps a vehicle cannot move under the gradient method and its location is not within the target area, it is considered to be trapped. The vehicle then switches to the Gibbs sampler-based simulated annealing with a predetermined cooling schedule and goes to Step 3;
- *Step 3:* After N -step simulated annealing, the vehicle switches to the gradient method and goes to Step 2.

The same potential function $\Phi_s(x)$ will be used for both the deterministic scheme and the stochastic scheme. Let x be the current configuration, and $\mathcal{N}_m^x(s)$ be the set of s -neighbors of x within one move, as defined in (21). In the gradient-flow scheme, vehicle s takes a move to minimize $\Phi_s(z)$, $z \in \mathcal{N}_m^x(s)$. In the stochastic scheme, vehicle s makes a move by sampling a Gibbs distribution as in the single-vehicle case with potential $U(y_s) \triangleq \Phi_s(y_s, x_{S/s})$. Note that $x_{S/s}$ will vary as other vehicles make their own moves.

As vehicles move simultaneously, conflicts can occur, i.e., multiple vehicles contend for one spot. To resolve this issue, it is assumed that $R_s \geq 2R_m$ and thus potentially contending nodes are within the local communication range before they make the moves. A conflict can then be resolved *locally*. For instance, a uniform sampling could be performed among the contenders to pick a winner, and the winner takes the desired spot while the other contenders stay put for this round.

5.2. Choice of switching parameters

In the hybrid algorithm there are two key parameters: d and N . d is the waiting time that triggers a vehicle to switch from

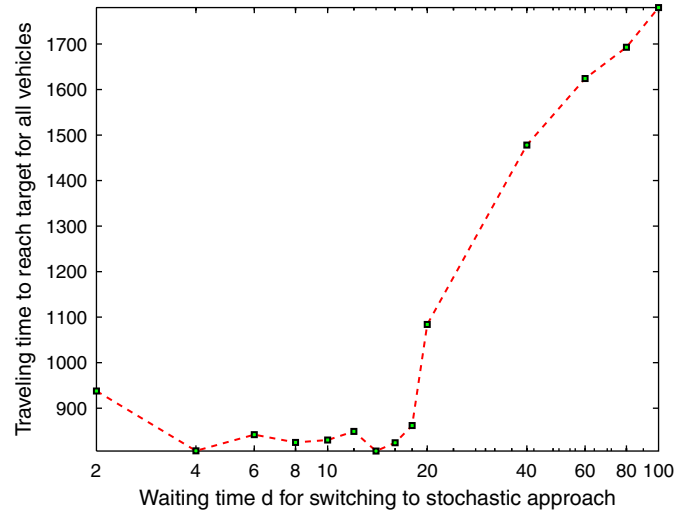


Fig. 9. Average traveling time versus switching parameter.

the deterministic approach to the stochastic one, and N is the duration of stochastic exploration.

5.2.1. Waiting time d

When d is smaller, it is more likely for a vehicle to make a wrong decision and switch to simulated annealing. In particular, a vehicle may be “trapped” temporarily due to the presence of its neighbors. Premature switching to stochastic exploration adds to the traveling cost. On the other hand, if d is too large, it will also be a waste of time if indeed the current cell is a trapping spot.

This tradeoff was verified through simulations. In the simulation there were 20 vehicles on a 48×48 grid (see Fig. 1). The target has a center $p^g = (43, 43)$ with radius $R^g = 5$, and the overlapping circular obstacles are centered at $(17, 23)$ and $(23, 17)$, respectively, with radii 5. Initially the vehicles were randomly distributed close to the other corner which is opposite to the target. The potential terms for target attraction and obstacle avoidance were as in (18), while a neighbor potential term was adopted to encourage having more neighbors. The following were used: $\lambda_g = 10$, $\lambda_o = 1$, $\lambda_n = 5$, $R_m = \sqrt{2} + \varepsilon$, $R_i = 6\sqrt{2} + \varepsilon$, $R_s = 7\sqrt{2} + \varepsilon$, and $T(n) = \frac{100}{\ln(n)}$.

A quantity u_g was defined, $u_g \triangleq \sum_{s \in S} \|x_s - p^g\|^2$, to measure how far the vehicles, as a whole, are away from the target. The simulation stops if $u_g \leq 200$, and the traveling time is recorded. So at the end of each simulation, the average distance between the target and vehicles is about $\sqrt{10}$, which is less than the target radius R^g . In the simulation d was varied from 2 to 100 while N was fixed to 100. For each d , 10 simulation runs were performed and the traveling times were averaged. Fig. 9 shows the average traveling time versus d . From the figure, a moderate d should be chosen for the best efficiency.

5.2.2. Duration N

Intuitively, very small N may not provide a trapped vehicle enough opportunities to get out; but very large N will kill the time-saving advantage offered by the gradient-flow algorithm.

To facilitate the analysis, a simplifying assumption is adopted. Considering that each vehicle makes its own moving decision, one might approximate the multi-vehicle system as a collection of independent single vehicles. Furthermore, it is assumed that the time spent on the gradient-flow method is much shorter than that spent on the stochastic approach, and can be neglected. The latter assumption was verified by simulation. Define the *reachable* area to be the set of cells from which a vehicle can reach the target area under the gradient-flow method, and the *unreachable* area to be its complement. Starting from the unreachable area with the gradient method, a vehicle will be trapped at some point and it will then switch to simulated annealing. For the duration N of stochastic exploration, let $P(N)$ be the probability that a vehicle will move to the reachable area after N -step simulated annealing. Then the expected time for the vehicle to reach the target is

$$T_{\text{total}} = \sum_{k=1}^{\infty} k \cdot N \cdot P(N)(1 - P(N))^{k-1} = N \cdot P(N)^{-1}, \quad (29)$$

where the vehicle is assumed to start from the unreachable area, otherwise $T_{\text{total}} = 0$.

The key question then becomes how to evaluate $P(N)$. Fortunately, a bound on $P(N)$ can be obtained based on Proposition 3.1. Let Π_{∞} has mass 1 in the target area. Since the target belongs to the reachable area,

$$P(N) \geq \text{Prob}(\text{vehicle reaches target}) \geq 1 - \frac{1}{2} \text{const} \cdot N^{-\frac{\lambda \delta_m}{\delta_m + \lambda \tau d}}. \quad (30)$$

Combining (29) and (30), one obtains

$$T_{\text{total}} \leq \frac{N}{1 - \frac{1}{2} \text{const} \cdot N^{-\frac{\lambda \delta_m}{\delta_m + \lambda \tau d}}}. \quad (31)$$

Eq. (31) clearly indicates that an optimal N exists to minimize the bound on T_{total} . This analysis was confirmed by simulation. The same scenario and parameters were used as earlier except that d was fixed to 6 and N was varied from 30 to 600. Fig. 10 shows the average traveling time versus N , and one can see that a choice of N between 50 and 150 would achieve good time-efficiency.

5.3. Effect of the memory

In this subsection the notion of memory is introduced to further improve the hybrid scheme. The idea is to record the trapping spots and reduce the probability of repeatedly being trapped at the same spots. Each vehicle keeps track of the *risk level* of dangerous cells, and accordingly lower the probability of accessing high-risk regions in simulated annealing.

- *Step 1:* Pick d , N , and $T(n)$, and let all vehicles initially choose the gradient-flow method;
- *Step 2:* When a vehicle s determines that it has been trapped at cell y , it increases the risk level R_y^s by 1 (the default risk

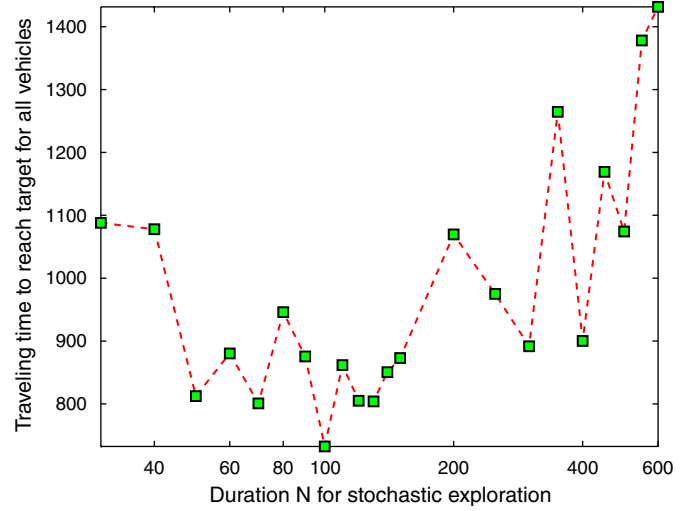


Fig. 10. Average traveling time versus duration for stochastic exploration.

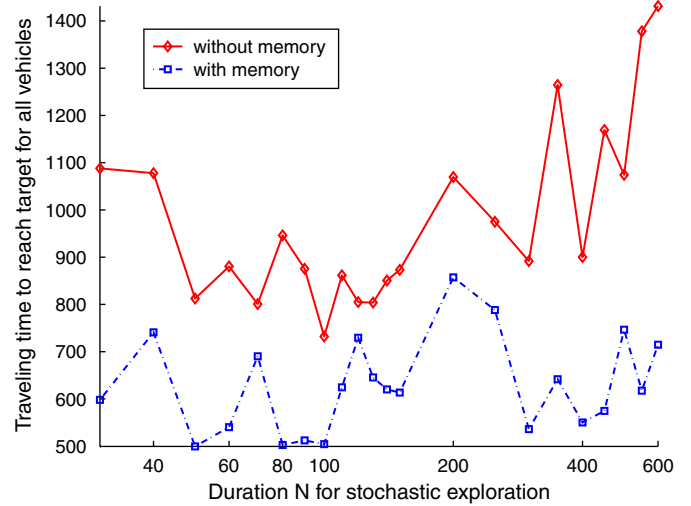


Fig. 11. Impact of memory on the average traveling time.

level for every location is 1). Then the vehicle switches to simulated annealing with $n = 1$;

- *Step 3:* At each annealing step, vehicle s determines the set C_m^s of candidate locations for its next move. For $l \in C_m^s$, it evaluates the Gibbs potential function $\Phi_s(x_s=l, x_{s/s})$, which is simply denoted as $\bar{\Phi}_s(l)$. Then vehicle s will take l with the probability

$$P(x_s = l) = \frac{e^{-\frac{\bar{\Phi}_s(l)}{T(n)} / R_l^s}}{\sum_{l' \in C_m^s} e^{-\frac{\bar{\Phi}_s(l')}{T(n)} / R_{l'}^s}};$$

- *Step 4:* Increase n by 1 and repeat Step 3 until n reaches N . The vehicle s then switches back to the gradient-flow algorithm and goes to Step 2;
- *Step 5:* The algorithm stops if the aggregate distance $u_g \leq \bar{\epsilon}$ for some specified tolerance $\bar{\epsilon}$.

To compare the performance with the original hybrid control scheme, simulation was performed with the same setup as in the previous section, where $d = 6$ and N was varied from 30 to 600. As seen in Fig. 11, the hybrid control scheme with memory always achieves better performance than the memoryless one.

6. Conclusions and discussions

In this paper a stochastic algorithm was presented for coordination of autonomous swarms. The algorithm was based on the Gibbs sampler with a random visiting scheme. The specific choice of the proposal distribution results in Gibbs-type distributions for vehicle configurations, leading to the convergence of the algorithm. The random visiting scheme entails long-range communication for notifying newly selected nodes although such information exchange is minimal. A deterministic sequential visiting scheme would eliminate this requirement; however, the convergence behavior would be unclear since the stationary distribution for each T is no longer of the Gibbs-type.

An intriguing idea would be to update the vehicles in parallel, each with a Gibbs sampler-based on local characteristics. This would lead to a fully decentralized scheme, which is desirable for large-scale systems. Although simulation has shown promising signs along this line (Baras & Tan, 2004), analytical results are currently lacking. Without further assumptions, it is difficult to argue that the overall system only settles at global minima of the potential function.

In the current paper a hybrid algorithm was also developed in an attempt to improve the efficiency of the stochastic scheme. Some analysis was provided to gain insight into the algorithm. The algorithm is heuristic in nature and treats each vehicle independently, and thus one cannot expect to get analytical convergence results for the network as a whole. In some sense the hybrid control scheme offers stochastic perturbation to the deterministic gradient-flow method; however, it has advantages over random walk-type perturbation schemes (Barraquand et al., 1992). Take the single-vehicle case as an example. One can show that the probability of a vehicle getting trapped again after N annealing steps goes to zero as N goes to infinity. For a random-walk perturbation scheme, however, as the number N of random walks goes to infinity, the configuration distribution approaches the stationary distribution of the Markov chain defined by the random walk, and there is no guarantee for the vehicle to reach the target as $N \rightarrow \infty$.

The coordination schemes presented are meant for high-level path planning. They should be combined with low-level planning and control modules in implementation. Since vehicles move on a discrete lattice, the choice of discretization level becomes an important problem. From the simulation results, a practical and efficient strategy is to use the coarsest discretization that is compatible with the problem setup (e.g., the minimum inter-vehicle separation). The cell geometry is another factor in discretization. In this paper the mission space was discretized into a lattice with square cells. One could use cells of other geometries without changing the algorithm (except the numbering scheme for cells) to implement, e.g., triangular or hexagonal formations. For future work, it would also be

interesting to investigate the robustness of the algorithm when uncertainties in sensing exist.

Acknowledgments

This research was supported by the Army Research Office under the ODDR&E MURI01 Program Grant No. DAAD19-01-1-0465 to the Center for Networked Communicating Control Systems (through Boston University), and under ARO Grant No. DAAD190210319.

References

- Baras, J. S., & Tan, X. (2004). Control of autonomous swarms using Gibbs sampling. In *Proceedings of the 43rd IEEE conference on decision and control* (pp. 4752–4757). Atlantis, Paradise Island, Bahamas.
- Baras, J. S., Tan, X., & Hovareshti, P. (2003). Decentralized control of autonomous vehicles. In *Proceedings of the 42nd IEEE conference on decision and control* (Vol. 2) (pp. 1532–1537), Maui, Hawaii.
- Barraquand, J., Langlois, B., & Latombe, J.-C. (1992). Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(2), 224–241.
- Bremaud, P. (1999). *Markov chains, Gibbs fields, Monte Carlo simulation and queues*. New York: Springer.
- Carlin, B. P., & Louis, T. A. (2000). *Bayes and empirical Bayes methods for data analysis*. Boca Raton, FL: Chapman & Hall/CRC.
- Chellappa, R., & Jain, A. (1993). *Markov random fields: Theory and applications*. Boston: Academic Press.
- Gazi, V., & Passino, K. M. (2003). Stability analysis of swarms. *IEEE Transactions on Automatic Control*, 48(4), 692–697.
- Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions and automation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721–741.
- Horn, R., & Johnson, C. R. (1985). *Matrix analysis*. New York: Cambridge University Press.
- Jadbabaie, A., Lin, J., & Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6), 988–1001.
- Justh, E., & Krishnaprasad, P. S. (2004). Equilibria and steering laws for planar formations. *Systems and Control Letters*, 52, 25–38.
- Khatib, O. (1986). Real time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotic Research*, 5(1), 90–98.
- Kim, J., & Khosla, P. (1992). Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, 8(3), 338–349.
- Koren, Y., & Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 1398–1404), Sacramento, CA.
- Lawton, J. R. T., Beard, R. W., & Young, B. J. (2003). A decentralized approach to formation maneuvers. *IEEE Transactions on Robotics and Automation*, 19(6), 933–941.
- Leonard, N. E., & Fiorelli, E. (2001). Virtual leaders, artificial potentials and coordinated control of groups. In *Proceedings of the 40th IEEE conference on decision and control* (pp. 2968–2973), Orlando, FL.
- Levine, H., Rappel, W., & Cohen, I. (2000). Self-organization in systems of self-propelled particles. *Physical Review E*, 63, 017101.
- Liu, C., Jr., M. H. A., Krishna, H., & Yong, L. S. (2000). Virtual obstacle concept for local-minimum-recovery in potential-field based navigation. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 983–988), San Francisco, CA.
- Medvedovic, M., Young, K. Y., & Bumgarner, R. E. (2004). Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics*, 20(8), 1222–1232.
- Mogilner, A., & Edelstein-Keshet, L. (1999). A non-local model for a swarm. *Journal of Mathematical Biology*, 38, 534–570.

- Moreau, L. (2005). Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2), 169–182.
- Ogren, P., Fiorelli, E., & Leonard, N. E. (2004). Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49(8), 1292–1302.
- Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3), 401–420.
- Olfati-Saber, R., & Murray, R. M. (2002). Distributed cooperative control of multiple vehicle formations using structural potential functions. In *Proceedings of the 15th IFAC world congress*, Barcelona, Spain.
- Olfati-Saber, R., & Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9), 1520–1533.
- Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22(3), 52–67.
- Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavior model. *Computer Graphics*, 21(4), 25–34.
- Rimon, E., & Koditschek, D. E. (1992). Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5), 501–518.
- Tanner, H. G., Jadbabaie, A., & Pappas, G. J. (2003). Stable flocking of mobile agents, Part I: Fixed topology. In *Proceedings of the 42nd IEEE conference on decision and control* (pp. 2010–2015). Maui, Hawaii.
- Vicsek, T., Czirook, A., Ben-Jacob, E., Cohen, O., & Shochet, I. (1995). Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6), 1226–1229.
- Volpe, R., & Khosla, P. (1990). Manipulator control with superquadric artificial potential functions: Theory and experiments. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(6), 1423–1436.
- Winkler, G. (1995). *Image analysis, random fields, and dynamic Monte Carlo methods: A mathematical introduction*. New York: Springer.



Wei Xi was born in Nanyang, China, in 1975. He received the Bachelor's degree in Mechanical Design and Automation and the Master's degree in System Engineering from Tsinghua University, Beijing, China, in 1998 and 2001, respectively. Since 2002 he has been working toward his Ph.D. degree in the Electrical and Computer Engineering Department, University of Maryland at College Park, USA. His Ph.D. thesis is focused on the distributed path planning and motion control of autonomous swarms.

His research interests include distributed control of networked swarms, and communication-constrained control of networked dynamic systems.



Xiaobo Tan received the Bachelor's and Master's degrees in Automatic Control from Tsinghua University, Beijing, China, in 1995, 1998, and his Ph.D. in Electrical and Computer Engineering from the University of Maryland at College Park, USA, in 2002.

From September 2002 to July 2004 he was a Research Associate with the Institute for Systems Research (ISR) at the University of Maryland. In August 2004 he joined the Department of Electrical and Computer Engineering at Michigan

State University as an Assistant Professor. His research interests include modeling and control of smart materials and micro-electromechanical systems, artificial muscle-based biomimetic robotics, micromanipulation, distributed control of networked systems, embedded control systems, and numerical integration of dynamical systems on manifolds.

Dr. Tan was an ISR Systems Fellow from 1998 to 2002. He was a finalist for the Best Student Paper Award at the 2002 IEEE Conference on Decision and Control, a co-recipient of the Best Poster Award at the Greater Washington-Baltimore Area MEMS Alliance Special Topics Symposium in April 2003, and a recipient of the NSF CAREER Award in 2006.



John S. Baras received the B.S. degree in Electrical Engineering from the National Technical University of Athens, Greece, in 1970, and the M.S. and Ph.D. degrees in Applied Mathematics from Harvard University, Cambridge, MA, in 1971 and 1973, respectively.

Since 1973 he has been with the Electrical and Computer Engineering Department and the Applied Mathematics Faculty, at the University of Maryland, College Park, where he is currently a Professor and the Lockheed Martin Chair in

Systems Engineering. He was the founding Director of the Institute for Systems Research (ISR, one of the first six NSF ERCs) from 1985 to 1991, and has been the Director of the Center for Hybrid and Satellite Communication Networks (a NASA Center for the Commercial Development of Space) since 1991. His research interests include integration of logic programming and nonlinear programming for trade-off analysis, object-oriented modeling, validation, and verification of complex systems models and engineering designs, hybrid, satellite and wireless communication networks, integrated network management systems, network security, stochastic systems, robust control of nonlinear systems, real-time parallel architectures for nonlinear signal processing, intelligent control systems, expert and symbolic systems for control and communication systems synthesis, biomimetic algorithms and systems for signal processing and sensor networks, intelligent manufacturing of smart materials. He has published over 350 technical articles on control, communication and computing systems, and was the editor of the book *Recent Advances in Stochastic Calculus* (New York: Springer, 1990).

Dr. Baras is a Fellow of the IEEE. He has served on the Board of Governors of the IEEE Control Systems Society, IEEE Engineering R&D Committee, Aerospace Industries Association advisory committee on advanced sensors, and IEEE Fellow evaluation committee. He is currently serving on the editorial boards of *Mathematics of Control, Signals, and Systems*, *of Systems and Control: Foundations and Applications*, *IMA Journal of Mathematical Control and Information*, and *Systems Automation-Research and Applications*. He has received the 1978, 1983 and 1993 Naval Research Laboratory Research (Alan Berman) Publication Awards, the 1980 Outstanding Paper Award of the IEEE Control Systems Society, the 1996 Outstanding Paper Award at Design SuperCon, and the 2002 Best paper Award at the 23rd Army Science Conference.