A Hybrid Scheme for Distributed Control of Autonomous Swarms

Wei Xi, Xiaobo Tan and John S. Baras

Abstract— **In this paper a hybrid scheme for distributed control of autonomous vehicles is presented by combining the deterministic gradient-flow method and the stochastic method based on the Gibbs sampler. The scheme has the advantages of both methods and can potentially provide fast, distributed maneuvers while avoiding getting trapped at local minima of the potential function. Preliminary analysis is performed for the optimal design of the parameters controlling the switching between the two methods. The performance of the hybrid scheme is further enhanced by the introduction of vehicle memory. Simulation results are provided to confirm the analysis and show the effectiveness of the proposed algorithm.**

I. Introduction

With the rapid advances in sensing, communication, computation, and actuation capabilities, autonomous unmanned vehicles (AUVs) are expected to cooperatively perform dangerous or explorative tasks in various hazardous, unknown or remote environments [1]. Distributed methods for control and coordination of vehicles are especially appealing due to the large scale of vehicle networks and bandwidth constraints on communication [2], [3], [4], [5]. A popular approach is based on artificial potential functions, which encode desired vehicle behaviors such as inter-vehicle interactions, obstacle avoidance, and target approaching [6], [7], [8], [9], [10]. Vehicles then follow the negative gradients of potentials mimicking the emergent behaviors (e.g. foraging) demonstrated by swarms of bacteria, insects, and animals [11].

The potential function-based approach has been explored for path planning and control of robotic manipulators and mobile robots over the past two decades [12], [13], [14]. 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 [15]. Researchers attempted to address this problem by designing potential functions that have no other local minima [16], [17], or escaping from local minima using ad hoc techniques, e.g., random walk [18], virtual obstacles [19], and virtual local targets [20].

An alternative approach to dealing with the local minima problem was explored using the concept of Markov Random Fields (MRFs) by Baras and Tan [21]. Traditionally used in statistical mechanics and in image processing [22], MRFs were proposed to model swarms of vehicles. Similar to the artificial potential approach, global objectives and constraints (e.g., obstacles) are reflected through the design of potential functions. The movement of vehicles is then decided using simulated annealing based on the Gibbs sampler. Simulation and analysis have shown that simulated annealing with the Gibbs sampler can lead to the achievement of global objectives with limited moving capabilities and communication ranges [21], [23]. However, the high traveling cost arising from the stochastic nature of the algorithm presents a barrier to its application in practice.

In this paper a hybrid approach is proposed, which combines the advantages of the deterministic gradient-flow method and the stochastic simulated annealing method. The algorithm works as follows. Each vehicle in a swarm makes its own decision to switch between the two methods: with initial choice of the gradient-flow scheme, a vehicle switches to simulated annealing when it determines that it gets trapped by some obstacles. After a certain number N of annealing steps, it switches back to the gradient-flow scheme to save traveling time and cost. A notion of memory is introduced to further improve the performance. Each vehicle records the “dangerous” locations where it has been trapped before, and adaptively takes this information into account when making moving decisions.

The paper is organized as follows. Some basic concepts on MRFs and the Gibbs sampler are reviewed in Section II. In Section III the path planning problem for an autonomous swarm is formulated. The hybrid scheme is proposed and its parameters' design is discussed in Section IV. Some preliminary analytical results are provided together with simulation results. The concept of memory is introduced and its impact studied in Section V. Concluding remarks are provided in Section VI.

II. Review of MRFs and Gibbs Sampler

A. MRFs and Gibbs Random Fields

Let S be a finite set of cardinality σ, with elements indexed by s and called *sites*. For $s \in S$, let Λ_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 Λ_s. A *configuration* of the system is $x = \{x_s, s \in S\}$ where $x_s \in \Lambda_s$, $\forall s$. The product space $\Lambda_1 \times \cdots \times \Lambda_\sigma$ is called the *configuration space*. A *neighborhood system* on S is a family $\mathscr{N} = \{\mathscr{N}_s\}_{s \in S}$, where $\forall s, r \in S$, $\mathscr{N}_s \subset S$, $s \notin \mathscr{N}_s$, and $r \in \mathscr{N}_s$ if and only if $s \in \mathscr{N}_r$. $\mathscr{N}_s$ is called the *neighborhood* of site s. The random field X is called a *Markov random field* (MRF) with respect to the neighborhood system $\mathscr{N}$

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.

W. Xi and J. S. Baras are with the Institute for Systems Research and the Department of Electrical & Computer Engineering, University of Maryland, College Park, MD 20742, USA. {wxi,baras}@isr.umd.edu

X. Tan is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA. xbtan@msu.edu

0-7803-9098-9/05/$25.00 ©2005 AACC

if, $\forall s \in S$, $P(X_s = x_s | X_r = x_r, r \neq s) = P(X_s = x_s | X_r = x_r, r \in \mathcal{N}_s)$.

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

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

where T is the temperature variable (widely used in simulated annealing algorithms), $U(x)$ is the potential (or energy) of the configuration x, and Z is the normalizing constant, called the *partition* function: $Z = \sum_x e^{-\frac{U(x)}{T}}$. One then considers the following useful class of potential functions $U(x) = \sum_{s \in \Lambda} \Phi_s(x)$, which is a sum of individual contributions Φ_s evaluated at each site. The Hammersley-Clifford theorem [24] establishes the equivalence of a Gibbs random field and an MRF.

B. Gibbs Sampler and Simulated Annealing

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 each site sequentially and sampling from the local specifications of a Gibbs field. One sequential visit to all sites is called a *sweep*. The convergence of the Gibbs sampler was studied by D. Geman and S. Geman in the context of image processing [25]. There it was shown that as the number of sweeps goes to infinity, the distribution of $X(n)$ converges to the Gibbs distribution Π. Furthermore, with an appropriate cooling schedule, simulated annealing using the Gibbs sampler yields a uniform distribution on the space of configurations corresponding to the minimum energy $U(x)$. Thus the global objectives could be achieved through appropriate design of the Gibbs potential function.

III. Problem Setup

Consider a 2D mission space (the extension to 3D space is straightforward), which is discretized into a lattice of cells. 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 of vehicles (or *mobile nodes*) S indexed by $s = 1, \cdots, \sigma$ on the mission space. To be precise, each vehicle (node) s is assumed to be a point mass located at the center of some cell (i_s, j_s), and the position of vehicle s is taken to be $p_s = (i_s, j_s)$. At most one vehicle is allowed to stay in each cell at any time instant.

The distance between two cells, (i_a, j_a) and (i_b, j_b), is defined to be $R \triangleq \sqrt{(i_a - i_b)^2 + (j_a - j_b)^2}$. 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^o, j^o) with radius R^o can be defined as $O \triangleq \{(i, j) : \sqrt{(i - i^o)^2 + (j - j^o)^2} \leq R^o\}$. 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 induced 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 is one *target area* in the space. A target area is a set of adjacent cells that represent desirable destinations of mobile nodes. A "circular" target area with center $p^g = (i^g, j^g)$ and radius R^g can be defined similarly as a "circular" obstacle: $G \triangleq \{(i, j) : \sqrt{(i - i^g)^2 + (j - j^g)^2} \leq R^g\}$.

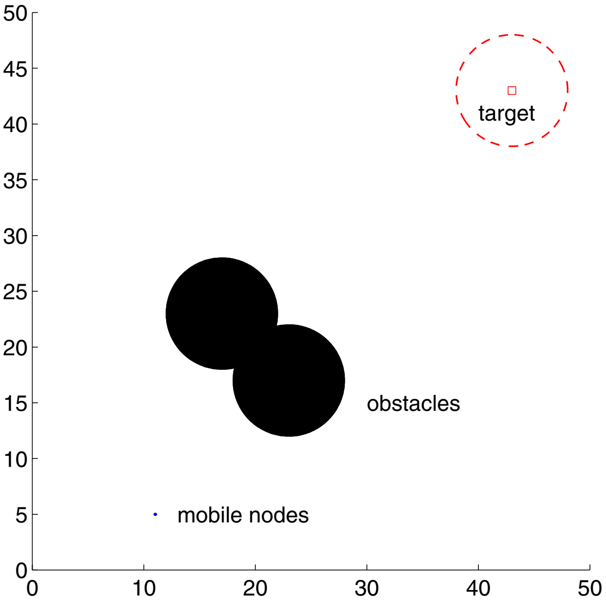

Fig. 1. An example mission scenario.

In this paper all nodes are assumed to be identical. Each node has a *sensing range* R_s: it can sense whether a cell within distance R_s is occupied by some node or obstacle. Communication between two nodes that are within a distance of R_s is regarded as local. The moving decision of each node s depends on other nodes located within distance R_i ($R_i \leq R_s$), called the *interaction range*. These nodes form the set $\mathcal{N}_s$ of *neighbors* of node s. A node can travel at most R_m ($R_m \leq R_s$), called *moving range*, within one move. We assume $R_s \geq R_i + R_m$, and $R_s \geq 2R_m$.

The mission goal is completed if all vehicles reach and gather in the target area. In addition, it is desired that vehicles have more neighbors. An example scenario, which will be used in the simulation, is shown in Fig. 1.

The neighborhood system defined earlier naturally leads to a dynamic graph where each vehicle stands for 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 Λ_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. An *admissible* configuration is a graph where vehicles stay in different cells.

The same potential function will be used for the gradient-flow method and the simulated annealing method. It is a sum of individual potentials $\Phi_s(x)$ for each site s

$$U(x) = \sum_{s \in S} \Phi_s(x), \tag{1}$$

where $x = \{x_s, 1 \leq s \leq \sigma\}$ is the configuration of vehicles. Moreover, the individual potential $\Phi_s(x)$ consists of three terms with each reflecting one goal or one constraint. To be specific,

$$\Phi_s(x) = \lambda_g J_s^g + \lambda_o J_s^o + \lambda_n J_s^n, \tag{2}$$

where J_s^g, J_s^o, and J_s^n account for the attraction from the target area, the repelling from obstacles, and the pulling force from 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 nodes behavior and the convergence rate of the algorithm [8], [23]. In this paper, it is assumed that there is one circular target area centered at p^g, and there are K (possibly overlapping) circular obstacles centered at p^{ok}, $1 \leq k \leq K$. The following potential functions are used:

$$\begin{aligned} J_s^g &= \|p_s - p^g\| \\ J_s^o &= \sum_{k=1}^{K} \frac{1}{\|p_s - p^{ok}\|} \\ J_s^n &= \begin{cases} \frac{1}{\sum\limits_{z \in \mathcal{N}_s} \frac{1}{\|p_s - p_z\|}}, & if \mathcal{N}_s \neq \emptyset \\ \Delta, & if \mathcal{N}_s = \emptyset \end{cases} \end{aligned} \tag{3}$$

where J_s^n tends to be smaller when site s has more neighbors. $\Delta > 0$ is a relative large constant and it represents the penalty for having no neighbors.

IV. The Hybrid Control Scheme

In this section the hybrid control scheme is first presented, and then the design of its algorithmic parameters is discussed.

A. A Hybrid Control Algorithm

The gradient-flow method alone provides fast march toward the target in the absence of obstacles but it may get vehicles trapped in non-target areas. On the other hand, the Gibbs sampler-based simulated annealing complements the gradient 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 times and traveling distance. The hybrid control scheme aims to combine advantages of both schemes while avoiding their disadvantages.

In the proposed scheme vehicles make moving decisions simultaneously and hence the scheme is fully parallel and scalable. The algorithm works as follows:

- Step 1. Each vehicle starts with the gradient-flow method (see below for more detail on the implementation of the gradient-flow method) and goes to Step 2;
- Step 2. If for d consecutive time instants a vehicle cannot move under the gradient method and its location is not within the target area, then it is considered to be trapped. The vehicle then switches to the simulated annealing method with a predetermined cooling schedule (see below for more detail) and goes to Step 3;
- Step 3. After performing simulated annealing for N time instants, the vehicle switches to the gradient method and goes to Step 2.

In the case of a conflict (multiple nodes contend for one spot), a uniform sampling is performed, and the winner will take the spot while the other contenders will stay put for the current time instant. Note that the resolution of conflict can be achieved *locally* since $R_s \geq 2R_m$ and potentially contending nodes are within the local communication range. In the simulation the algorithm will stop if

$$u_g = \sum_{s \in S} \|p_s - p^g\|^2 \leq \varepsilon, \tag{4}$$

where u_g is an indicator measuring how far the vehicles, as a whole, are away from the target area.

Implementation of the gradient-flow scheme and the simulated annealing scheme is provided next for completeness.

1) The gradient-flow scheme: In the gradient-flow method the velocities of vehicles follow the (negative) gradient flows of their potential surfaces. To be specific, at each time instant,

- Step 1. For each vehicle s determine the set Λ_s of candidate locations for the next move, i.e., the set of cells within the distance R_m and not occupied by other vehicles or obstacles;
- Step 2. For each $l \in \Lambda_s$, evaluate the potential function $\Phi_s(X(S\backslash s) = x(S\backslash s), x_s = l)$, where $S\backslash s$ denotes the complement of s in S;
- Step 3. Update the location of vehicle s by taking

$$x_s = \arg\min_{l \in \Lambda_s} \Phi_s(X(S\backslash s) = x(S\backslash s), x_s = l).$$

2) Gibbs sampler-based simulated annealing: Unlike the gradient-flow scheme, in simulated annealing each vehicle updates its next location by sampling a probability distribution. First a cooling schedule $T(n)$ is determined (how to choose a cooling schedule for best convergence performance is itself a vast subject and is beyond the scope of this paper).

- Step 1. Let $n = 1$;
- Step 2. The vehicle s determines the set Λ_s of candidate locations for the next move as in the gradient-flow method;
- Step 3. For each $l \in \Lambda_s$, the vehicle evaluates the Gibbs potential function $\Phi_s(l) = \Phi_s(X(S\backslash s) = x(S\backslash s), x_s = l)$,

and calculates the probability distribution

$$P(x_s = l) = \frac{e^{-\frac{\Phi_s(l)}{T(n)}}}{\sum\limits_{z \in \Lambda_s} e^{-\frac{\Phi_s(z)}{T(n)}}};$$

- Step 4. Update x_s to $l \in \Lambda_s$ with probability $P(x_s = l)$;
- Step 5. Let $n = n+1$, and repeat Step 2 to step 5.

B. Choice of Switching Parameters

In the hybrid algorithm there are two key parameters that determine the performance of the system: d and N. d is the waiting time that triggers a vehicle to switch from the deterministic approach to the stochastic one, and N is used to determine the duration of stochastic exploration.

1) Waiting time d: When d is smaller, it's 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 is verified through simulations. In the simulations there are 20 vehicles on a 48 × 48 grid (see Fig. 1). The target is located at the corner (5,48) with radius $R_g = 5$, and two overlapped circular obstacles with radius 5 are centered at (17, 23) and (23, 17), respectively. Initially the vehicles are randomly distributed close to the other corner which is opposite to the target. The parameters used are: $\lambda_g = 10$, $\lambda_o = 1$, $\lambda_n = 5$, $R_m = \sqrt{2}$, $R_s = 6\sqrt{2}$, and the cooling schedule $T(n) = \frac{100}{\log(n)}$. The stopping criterion $\varepsilon = 200$ (for the distance indicator u_g) is chosen. 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. While fixing the duration N to be 100, the waiting time d is increased from 2 to 100. For each d, 10 simulation runs are performed and the traveling times are averaged. Fig. 2 shows the average traveling time versus the waiting time d.

In the figure, when d is very small ($d = 2$), vehicles take about 950 steps to arrive at the target. Then the traveling time drops to 850 when d is between 4 and 18. After that, as d becomes larger and is comparable to stochastic exploration duration N, the performance is dramatically degraded. Clearly, a moderate d should be chosen for the best efficiency.

2) Duration N: Duration time N for stochastic perturbation is another key parameter. Intuitively a very small N may not provide a trapped vehicle enough opportunities to get out; and a very large N will kill the time-saving advantage offered by the gradient-flow algorithm. Therefore, it is of interest to study how the duration time N affects performance analytically.

A simplifying assumption is adopted to make the analysis tractable. 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 the time spent on the stochastic approach, and can be neglected. To justify the latter assumption, it has been found in simulations that a vehicle takes 40-50 times more steps to get to the target using the stochastic approach than using the gradient-flow approach in the absence of obstacles. 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 steps of simulated annealing with the Gibbs sampler. Then the expected time for the vehicle to reach the target is approximately

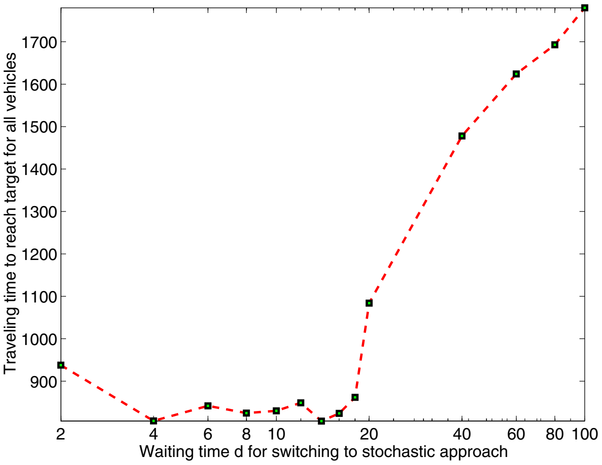

Fig. 2. Average traveling time versus the switching parameter d (waiting time).

$$\begin{aligned} T_{total} &= \sum_{k=1}^{\infty} k \cdot N \cdot P(N)(1-P(N))^{k-1} \\ &= N \cdot P(N)^{-1}, \end{aligned} \tag{5}$$

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

The key question then becomes how to evaluate $P(N)$. Fortunately, a bound on $P(N)$ can be obtained based on some recent results by the authors [23]. In particular,

$$\|\nu Q_1 \ldots Q_n - \Pi_\infty\|_1 \le const \cdot n^{-\frac{2\lambda\tilde{m}}{2\tilde{m}+\lambda\Delta\tau}} \tag{6}$$

where $\nu Q_1 \ldots Q_n$ represents the probability distribution of the (single) vehicle after n annealing steps, and Π_∞ denotes the probability distribution of sampled configurations as the annealing temperature reaches 0, and $\tilde{m}, \lambda, \Delta$, and τ are all constants dependent on the potential function. By designing the potential function so that the target location has the

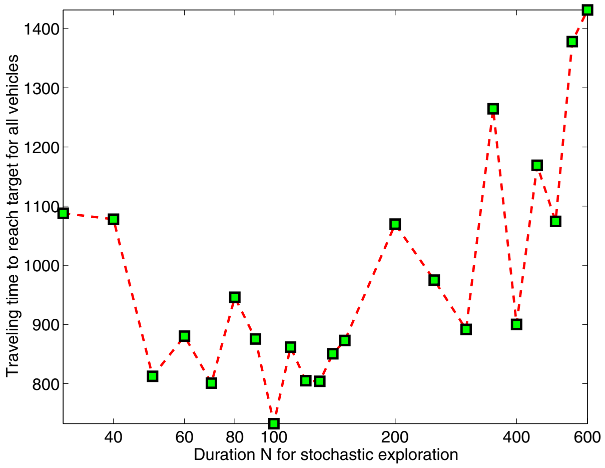

Fig. 3. Average traveling time versus the duration N for stochastic exploration.

lowest potential, Π_∞ has mass 1 in the target area. Since the target belongs to the reachable area, the probability $P(N)$ satisfies the following inequality:

$$\begin{aligned} P(N) &\geq P(\text{vehicle reaches target}) \\ &\geq 1-\frac{1}{2}const\cdot N^{-\frac{2\lambda\tilde{m}}{2\tilde{m}+\lambda\Delta\tau}}. \end{aligned} \tag{7}$$

Combining (5) and (7), one obtains

$$T_{total} \leq \frac{N}{1-\frac{1}{2}const\cdot N^{-\frac{2\lambda\tilde{m}}{2\tilde{m}+\lambda\Delta\tau}}}. \tag{8}$$

Eq. (8) clearly indicates that an optimal N exists to minimize the bound on T_{total}. This analytical result was confirmed by simulations. Same scenario and parameters are used as in earlier simulations except that the waiting time d for switching is now fixed to 6 and the duration N for stochastic exploration is varied from 30 to 600. Fig. 3 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.

C. *Comparison with Other Random Perturbation Schemes*

The hybrid control scheme proposed is essentially a kind of stochastic relaxation algorithm. However, simulated annealing based on the Gibbs sampler provides advantages over many other stochastic perturbation methods.

For ease of discussion, consider the single-vehicle case. One can show that the probability of a vehicle getting trapped again after N annealing steps goes to zero as N goes to infinity. This is not the case with just any stochastic scheme. Take a random walk-type perturbation scheme [18] as an example. As the number N of random walks goes to infinity, the distribution of the vehicle approaches the stationary distribution of the Markov chain defined by the random walk. In general there will be positive masses falling in the unreachable area. This implies that there is no guarantee for the vehicle to reach the target no matter how many steps are used for the random perturbation.

V. Hybrid Scheme with Memory

In this section the notion of memory is introduced to further enhance the hybrid control 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 modifies its potential surface to lower the probability of accessing high-risk regions. To be specific, the enhanced algorithm works as follows:

- Step 1. Initialize the algorithm, set parameter d, N, and the cooling schedule $T(n)$, 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 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 Λ_s of candidate locations for its next move. For $l\in\Lambda_s$, it evaluates the Gibbs potential function $\Phi_s(X(S\backslash s)=x(S\backslash s),x_s=l)$, which is simply denoted as $\Phi_s(l)$. Then vehicle s will take l with the probability
$$P(x_s=l)=\frac{e^{-\frac{\Phi^s(l)}{T(n)}}/R_l^s}{\sum_{z\in\Lambda_s} e^{-\frac{\Phi^s(z)}{T(n)}}/R_z^s};$$
- 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 indicator $u_g\leq\varepsilon$.

To compare the performance with the original hybrid control scheme, simulations were performed with the same setup as in the previous section. The waiting time was set to be $d=6$ and the duration N was varied from 30 to 600. As seen in Fig. 4, the hybrid control scheme with memory always achieves better performance than the original one for all N.

VI. Conclusions and Discussions

A hybrid control scheme was proposed for path generation of autonomous swarms. The scheme combines the advantages of the deterministic gradient-flow approach and of the stochastic simulated annealing based on the Gibbs sampler. The selection of the two key parameters involved in the switching was investigated through analysis and simulation. An enhanced version of the algorithm was also presented by adaptively adjusting the local sampling probabilities based on the risk levels perceived by individual vehicles.

It should be noted that the scheme is meant to be a high-level planning algorithm. In particular, it does not address issues like collision avoidance at intermediate spots when

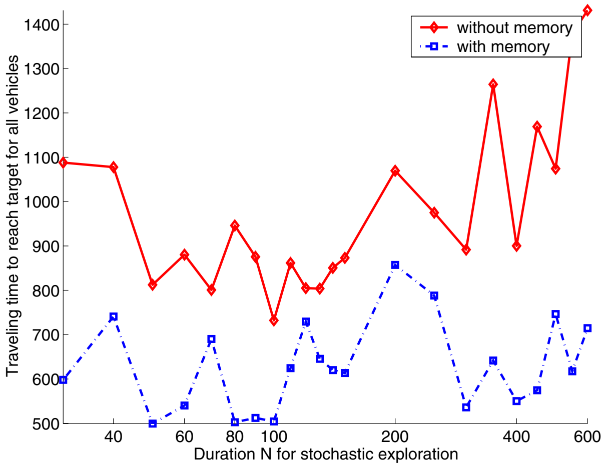

Fig. 4. Averaged traveling time under the hybrid scheme with memory, in comparison with that under the memoryless scheme.

nodes simultaneously take the moves (although the conflicts in destinations of each move were resolved). In a practical implementation, such issues need to be considered together with specific vehicle dynamics and constraints.

Other variants of the hybrid scheme can be developed. For instance, one could use simulated annealing with non-monotone cooling schedules with the trapping status reflected through the temperature. One could also use a weighted combination of the decisions produced through both the gradient-flow method and simulated annealing, where the weight depends on the trapping status.

References

[1] D. A. Schoenwald, "AUVs: In space, air, water, and on the ground," *IEEE Control Systems Magazine*, vol. 20, no. 6, pp. 15–18, 2000.

[2] R. Olfati-Saber and R. M. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," in *Proceedings of the 15th IFAC World Congress*, Barcelona, Spain, 2002.

[3] J. R. T. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 6, pp. 933–941, 2003.

[4] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[5] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable flocking of mobile agents, Part I: Fixed topology," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, 2003, pp. 2010–2015.

[6] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, FL, 2001, pp. 2968–2973.

[7] P. Song and V. Kumar, "A potential field based approach to multi-robot manipulation," in *Proceedings of the IEEE International Conference on Robots and Automation*, Washington, DC, 2002, pp. 1217–1222.

[8] J. S. Baras, X. Tan, and P. Hovareshti, "Decentralized control of autonomous vehicles," in *Proceedings of the 42nd IEEE Conference on Decision and Control, Maui*, vol. 2, Maui, Hawaii, 2003, pp. 1532–1537.

[9] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, 2004.

[10] D. H. Kim, H. O. Wang, G. Ye, and S. Shin, "Decentralized control of autonomous swarm systems using artificial potential functions: Analytical design guidelines," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, vol. 1, Atlantis, Paradise Island, Bahamas, 2004, pp. 159–164.

[11] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.

[12] O. Khatib, "Real time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotic Research*, vol. 5, no. 1, pp. 90–98, 1986.

[13] R. Shahidi, M. Shayman, and P. S. Krishnaprasad, "Mobile robot navigation using potential functions," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, CA, 1991, pp. 2047–2053.

[14] E. Rimon and D. E. Kodistschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[15] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, CA, 1991, pp. 1398–1404.

[16] R. Volpe and P. Khosla, "Manipulator control with superquadric artificial potential functions: Theory and experiments," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 6, pp. 1423–1436, 1990.

[17] J. Kim and P. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 338–349, 1992.

[18] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 2, pp. 224–241, 1992.

[19] C. Liu, M. H. A. Jr, H. Krishna, and L. S. Yong, "Virtual obstacle concept for local-minimum-recovery in potential-field based navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, 2000, pp. 983–988.

[20] X. Zou and J. Zhu, "Virtual local target method for avoiding local minimum in potential field based robot navigation," *Journal of Zhejiang University Science*, vol. 4, no. 3, pp. 264–269, 2003.

[21] J. S. Baras and X. Tan, "Control of autonomous swarms using Gibbs sampling," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004, pp. 4752–4757.

[22] G. Winkler, *Image Analysis, Random Fields, and Dynamic Monte Carlo Methods : A Mathematical Introduction*. New York: Springer-Verlag, 1995.

[23] W. Xi, X. Tan, and J. S. Baras, "Gibbs sampler-based path planning for autonomous vehicles: Convergence analysis," in *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, 2005, to appear.

[24] P. Bremaud, *Markov Chains, Gibbs Fields, Monte Carlo Simulation and Queues*. New York: Springer Verlag, 1999.

[25] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and automation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.