

Flocking with Obstacle Avoidance: A New Distributed Coordination Algorithm Based on Voronoi Partitions*

Magnus Lindhé and Petter Ögren
Autonomous Systems
Swedish Defence Research Agency
SE-172 90 Stockholm, Sweden
malind@foi.se, petter.ogren@foi.se

Karl Henrik Johansson
Signals, Sensors & Systems
Royal Institute of Technology
SE-100 44 Stockholm, Sweden
kallej@s3.kth.se

Abstract—A new distributed coordination algorithm for multi-vehicle systems is presented in this paper. The algorithm combines a particular choice of navigation function with Voronoi partitions. This results not only in obstacle avoidance and motion to the goal, but also in a desirable geographical distribution of the vehicles. Our algorithm is decentralized in that each vehicle needs only to know the position of neighboring vehicles, but no other inter-vehicle communication or centralized control are required. The algorithm gives a natural priority to safety, goal convergence, and formation keeping, in that (1) collision avoidance is guaranteed under all circumstances, (2) the vehicles will move toward the goal as long as a given optimization problem is feasible, and (3) if prior criteria admit, the vehicles tend to a desirable lattice formation. These theoretical properties are discussed in the paper and the performance of the algorithm is illustrated in simulations with realistic models of twenty all-terrain vehicles. Planned experimental evaluation using customized miniature cars is also briefly described.

Index Terms—Multi-robot systems; Flocking; Swarming; Obstacle avoidance

I. INTRODUCTION

A growing number of applications motivate the development of coordination algorithms for autonomous multi-vehicle systems and problems concerning formation control have recently attracted the attention of several researchers [3], [4], [5]. Most of these algorithms, however, do not treat obstacle avoidance in complex environments in a satisfying way. Obstacle avoidance is fundamental for any practical mobile system and has been extensively studied for single-robot systems [1], [2]. In general, it is non-trivial to extend or combine existing methods for obstacle avoidance and formation control in a systematic way. This paper proposes a solution to this problem.

Recent contributions to obstacle avoidance for multi-vehicle systems is presented in [7], [8], [9], [10], [11]. Most of these studies use appealingly minimalistic controllers to achieve quite remarkable group behavior. These

controllers can sometimes be motivated by observations of biological systems. There is, however, a prize to pay for the simplicity. In many cases assumptions need to be imposed on the environment, such as convexity constraints on the obstacles. These assumptions are artificial and seldom valid in either man-made or natural environments. General obstacles are allowed in [10], but the scheme is limited to rigid centralized leader follower formations. The idea to map certain obstacle shapes to points and then apply a scheme using switching interaction potentials is discussed in [9].

The main contribution of this paper is a new coordination algorithm for flocking in a terrain with obstacles of general shape. Our solution is based on a modified navigation function [2] and a coverage control algorithm [12]. The navigation function considered provides a computationally efficient way to achieve motion to the goal. In the coverage control algorithm, a Voronoi region is updated continuously for each vehicle and the control makes the vehicle move to the centroid of its region. The naive obstacle extension would be to just intersect the Voronoi regions with the obstacles and apply the approach of [12]. Doing that would however render all the local minima problems of the early obstacle avoidance schemes. We propose to let the obstacles influence the density function used to compute the centroid, as well as the Voronoi regions themselves. Furthermore, we introduce artificial mirror neighbors to achieve formation maintenance when the vehicles are moving in an open space.

The organization of the paper is as follows. In Section II we briefly present the navigation function tool and the methods we borrow from coverage control. Section III describes the proposed coordination algorithm, which is decentralized and thus to be executed in parallel on all vehicles. In Section IV analytical results on the properties of the algorithm are presented. Particularly, safety, goal convergence, and formation invariance are investigated. The coordination algorithm is then evaluated through simulations in Section V. The simulations illustrate the nice multi-vehicle performance generated by the algorithm both

* The work was partially supported by the Swedish Defence Research Agency, the European Commission through the HYCON Network of Excellence, the Swedish Foundation for Strategic Research through an Individual Grant for the Advancement of Research Leaders and the Swedish Research Council.

for simple integrator vehicle dynamics as well as for quite realistic vehicle models. The latter is motivated from ongoing development of an experimental multi-vehicle platform, which is also briefly described. Some concluding remarks are given in Section VI.

II. PRELIMINARIES

The proposed multi-vehicle coordination algorithm is based on a navigation function and on Voronoi partitions, which are both discussed in this section.

A. A Modified Navigation Function

Navigation functions were introduced by Rimon and Koditschek [6] and denote a special version of the potential functions used in many navigation and obstacle-avoidance schemes, e.g., [13]. Here we introduce the slightly modified navigation function used in [2].

Consider an obstacle-free bounded set $\Omega \subset \mathbb{R}^2$ containing the goal point p^* . The navigation function $f : \Omega \rightarrow \mathbb{R}_+$ is defined as a continuous function that approximately maps every point $p \in \Omega$ to the length of the shortest and collision-free path going from p to p^* . The function values are first calculated on a uniform rectangular grid covering Ω . These values are then interpolated to the whole of Ω . When computing the path lengths in the grid one can imagine adding the possibility of diagonal steps, thus allowing steps from an 8-neighborhood, or just use the 4-neighborhood of the original grid. Both these options are depicted in Figure 1. Although the 8-neighborhood in one sense is a better approximation to the Euclidean length of the shortest path, the 4-neighborhood, combined with a particular interpolation scheme, has the property of constant gradient norm. This is important, since the 8-neighborhood might give raise to problems similar to those caused by local minima. Obstacle configurations can be constructed, where two adjacent gridpoints have values that are arbitrarily close, making the gradient arbitrarily small. We use the 4-neighborhood to guarantee the absence of such problems. See [2] for details on the interpolation scheme.

Note that f does not have any undesired local minima, and the negative gradient of f at p always gives a direction in which the length of the shortest path to p^* decreases. In comparison to the navigation function introduced by Rimon and Koditschek [6], the modified navigation function described above is easier to compute, but on the other hand is only piecewise differentiable and requires the introduction of a rectangular mesh covering Ω .

B. Voronoi Partitions

Next we review briefly some notation on coverage control based on Voronoi partitions from Cortes et al. [12].

Consider $N > 1$ vehicles in a convex polytope $Q \subset \Omega$ positioned at $p_i \in Q$ and collected in the set

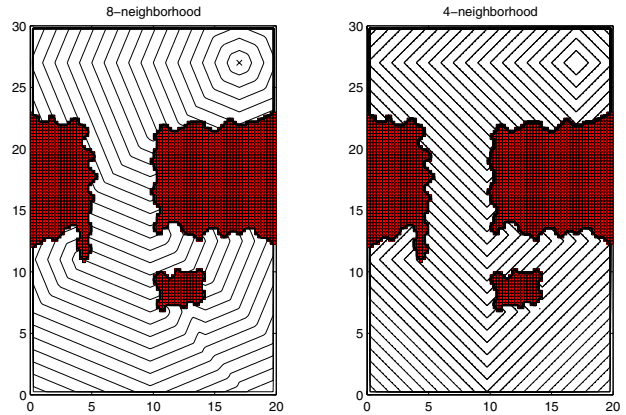


Fig. 1. Level curves of the navigation function f , computed using both 8- and 4-neighborhoods.

$P = \{p_1, p_2, \dots, p_N\}$. A collection of regions $V(P) = \{V_1, V_2, \dots, V_N\}$, $V_i \subset Q$, is denoted a Voronoi partition generated by P if

$$V_i = \{q \in Q : \|q - p_i\| < \|q - p_j\|, \forall j \neq i\},$$

where $\|\cdot\|$ is the Euclidean norm. The sets V_i are called Voronoi regions and a typical partition can be seen in Figure 2.

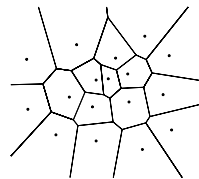


Fig. 2. Example of Voronoi regions.

Let $\phi : \Omega \rightarrow \mathbb{R}_+$ be a given map and introduce the function

$$H_V(P) = \int_Q \min_i \|q - p_i\|^2 d\phi(q).$$

Note that $H_V(P)$ can be interpreted as the expected value of the least-square distance to a vehicle $p \in P$ from a randomly located source $q \in Q$ having probability density ϕ . It turns out that the gradient of $H_V(P)$ with respect to the position p_i is equal to

$$\frac{\partial H_V(P)}{\partial p_i} = 2M_{V_i}(p_i - C_{V_i}),$$

where

$$\begin{aligned} M_{V_i} &= \int_{V_i} \phi(q) dq, \\ C_{V_i} &= \frac{1}{M_{V_i}} \int_{V_i} q\phi(q) dq. \end{aligned} \quad (1)$$

The quantities M_{V_i} and C_{V_i} are the generalized mass and center of mass (centroid), respectively, and V_i , $i = 1, \dots, N$, are the Voronoi regions associated with P .

If P^* is a local minimum of $H_V(P)$, then

$$V(P^*) = \{V_1^*, V_2^*, \dots, V_N^*\}$$

are the Voronoi regions generated by P^* , and P^* coincide with the centers of mass of the Voronoi regions defined by $V(P^*)$, i.e.,

$$C_{V_i^*} = \operatorname{argmin}_{p_i} H_V(P).$$

We are now ready to describe the proposed scheme.

III. DISTRIBUTED COORDINATION ALGORITHM

In this section we present our new multi-vehicle coordination algorithm for flocking and obstacle avoidance. Consider N vehicles governed by the discrete dynamics

$$p_i(k+1) = p_i(k) + u_i(k), \quad i = 1, \dots, N,$$

where $\|u_i\| \leq u_{\max}$ for some positive u_{\max} . The problem we are considering is that of driving the vehicles from an initial configuration $p_1(0), \dots, p_N(0)$ towards a goal point p^* , without colliding with each other or with obstacles. During the maneuver, we want the vehicles to maintain a flock, if possible, i.e., to keep the distance to nearby vehicles as close to some desired value $d > 0$ as possible. Each vehicle has only local sensor information about its immediate surroundings. Regarding the navigation function, the standard approach is to let f reflect all known obstacles and add new ones, and recalculate f , as they are encountered.

The reason for why first-order discrete vehicle dynamics suffice is that the algorithm is intended to be used as a high-level control component in a hierarchical control structure. When applying the algorithm to vehicles with complex dynamics, such as car-like robots, a lower-level control component is handling the task of achieving the motion from $p_i(k)$ to $p_i(k+1)$. In these cases, it is also convenient to supply the lower-level control algorithm with a guaranteed safe region W_i (as defined below), in which to plan the trajectory. Section V describes the hierarchical control structure further.

Before specifying the algorithm, we need some more definitions and notations. The map ϕ in the definition of H_V in previous section is set to

$$\phi(q) = e^{-kf(q)},$$

where f is the navigation function and the gain $k \geq 0$ is a design parameter, which affects the tradeoff between goal direction and formation maintenance. A nice property that follows from choosing ϕ to be an exponential function, is that the control law is independent of the actual distance to the goal when the goal is relatively far away, see the discussion at the end of Section IV.

Let the sensing radius of each vehicle be R , let furthermore $B_r(p) \subset \Omega$ denote the (open) disc with radius r and

center in p . We define P_i as the neighbors of p_i within R , i.e.,

$$P_i = \{p_j \in B_R(p_i) : j \neq i\}.$$

Let $S_i \subset \Omega$ denote the visible subset from $p_i \in \Omega$. Given a Voronoi partition $V(P) = \{V_1, V_2, \dots, V_N\}$ of Ω generated by $P = \{p_1, p_2, \dots, p_N\}$, define the sets

$$W_i = V_i \cap S_i \cap B_{R/2}(p_i), \quad i = 1, \dots, N. \quad (2)$$

The rationale behind the choice $B_{R/2}(p_i)$ is explained in Remark 3.2 below. For each $i \in \{1, \dots, N\}$, introduce the optimization problem

$$\begin{aligned} \min_{p \in W_i} \quad & \|p - C_{V_i}\|^2 \\ \text{s. t.} \quad & f(p) < f(p_i) - \epsilon \end{aligned} \quad (3)$$

where $\epsilon > 0$ is a design parameter. Let $d > 0$ denote the desired inter-vehicle spacing, when the vehicles are moving in an open space.

The following distributed control algorithm is executed in parallel on all vehicles. Initially it is assumed that the design parameters k, R, ϵ, d are fixed.

Coordination Algorithm

- 1) [*Sensor reading*] Set $p_i = p_i(k)$ and determine S_i and P_i from sensor measurements.
- 2) [*Mirror neighbor*] If p_i is not inside the convex hull of P_i , then for each $p_j \in P_i$ create a *mirror neighbor* \hat{p}_j as

$$\hat{p}_j = p_i - d \frac{p_j - p_i}{\|p_j - p_i\|},$$

Let \hat{P}_i be the union of P_i and all such mirror neighbors, i.e., $\hat{P}_i = P_i \cup \{\hat{p}_j\}$

- 3) [*Voronoi region*] Derive the Voronoi region V_i , based on \hat{P}_i , and compute $W_i = V_i \cap S_i \cap B_{R/2}(p_i)$ as in (2).
- 4) [*Optimization*] Derive the centroid

$$C_{W_i} = \frac{1}{M_{W_i}} \int_{W_i} q \phi(q) dq,$$

as in (1), and determine the minimizer of the optimization problem (3). If there is no feasible solution, set $p_i(k) = p_i$, otherwise set $p_i(k)$ equal to the minimizer.

- 5) [*Control actuation*] Let $\hat{u}_i(k) = p_i(k) - p_i$ and apply the control

$$u_i(k) = \begin{cases} \hat{u}_i(k), & \text{if } \|\hat{u}_i(k)\| < u_{\max} \\ \frac{\hat{u}_i(k)}{\|\hat{u}_i(k)\|} u_{\max}, & \text{otherwise.} \end{cases}$$

Let $k := k + 1$ and goto Step 1.

Remark 3.1: Safety, goal convergence and formation maintenance are three desirable properties of the multi-vehicle system. Note that they in many situations are contradictory. The proposed coordination algorithm, however, suggests a clear priority between them: (1) if there is no safe direction to move in, the vehicles stand still; (2) if there

are safe directions to move in, then motion towards the goal is given higher priority than formation maintenance; and (3) if there are several safe movements towards the goal, then the one with the best formation properties is chosen.

Remark 3.2: The Voronoi region V_i in the coordination algorithm is calculated with information only from $B_R(p_i)$. It is easy to construct examples when V_i does not agree with the Voronoi region \hat{V}_i , say, obtained with global information. An important observation is that V_i and \hat{V}_i agree inside $B_{R/2}(p_i)$, which leads to the control u_i being unaffected by this limitation on sensor information.

Remark 3.3: The optimization constraint $f(p) < f(p_i) - \epsilon$ in (3) guarantees that the navigation function decreases at each step, if a feasible point p exists. If no feasible point exists, the algorithm sets $p = p_i$ so the navigation function value for that particular vehicle remains unchanged. Since the navigation function approximates the distance to the goal, the constraint makes the vehicle move towards the goal or stand still. Note that even if one vehicle p_i has to stand still at a step due to the lack of feasible update points, the movement of the other vehicles at that step may generate feasible points for p_i at the next step.

Remark 3.4: The mirror neighbors are introduced in the algorithm to achieve flocking in open areas. They make the vehicles converge to a lattice formation with inter-vehicle distances equal to d . Without the mirror neighbors, the vehicles at the boundary of a flock will tend to “float away” in open areas. The coordination algorithm makes the multi-vehicle system perform suitable squeeze, split and join maneuvers under some obstacle configurations, as further described in Section V.

IV. PROPERTIES

In this section we present results on safety, goal convergence, and flocking for the proposed flocking algorithm. They support part of the previous remarks. The performance of the algorithm is then illustrated in the next section. We start by examining the property of highest priority.

Proposition 4.1 (Safety): The vehicles avoid collision with obstacles and other vehicles when coordinated by the algorithm.

Proof. First note that given a vehicle configuration $P = \{p_1, \dots, p_N\}$ at time k , the configuration $P' = \{p'_1, \dots, p'_N\}$ at time $k + 1$ must necessarily be such that $p'_i \in W_i$ for all $i = 1, \dots, N$, where $W_i = V_i \cap S_i \cap B_{R/2}(p_i)$. Obstacle avoidance follows from that W_i is disjoint from the subset of \mathbb{R}^2 that is occupied by the obstacles. Inter-vehicle collision avoidance follows similarly, since $W_i \subset V_i$ and $W_j \subset V_j$, $j \neq i$, are disjoint due to that the Voronoi regions are disjoint by construction. ■

The second priority, goal convergence, is investigated in the following lemma.

Proposition 4.2 (Goal convergence): At each iteration of the algorithm, the vehicles either move towards the goal, in the sense that f decreases along the trajectory, or stand still.

Proof. If the optimization problem (3) has a feasible solution, then $f(p'_i) < f(p_i) - \epsilon$ where p_i denotes the current position of vehicle i and p'_i the next. If the optimization problem has no feasible solution, then p'_i is set to p_i so the vehicle stands still. ■

Remark 4.3: Let \hat{f} denote the map for the true distance to the goal. It is easy to show that for a vehicle at p , $f(p)/\sqrt{2} < \hat{f}(p) \leq f(p)$. Hence, if $f(p)$ tends to zero along the trajectory of a vehicle, necessarily $\hat{f}(p)$ also tends to zero, i.e., the vehicle tends to the goal.

Remark 4.4: Note that there is always an obstacle-free direction along which the navigation function decreases. A vehicle thus only stands still when all descent directions are blocked by neighboring Voronoi regions. It is possible to construct examples in which this happens, so the algorithm does not give global convergence. Our experience shows, however, that such examples are quite rare and that in most practical situations the algorithm gives global convergence.

The third property to be discussed is formation maintenance.

Proposition 4.5 (Formation invariance): Consider a lattice of N vehicles in an open space. Suppose all vehicles have six neighbors, either real vehicles or mirror neighbors, and that the distances to all six neighbors are equal to d . Moreover, suppose that for all $j \neq i$, there exist $a_j \in \mathbb{R}^2$ and $A_j \in \mathbb{R}$ such that

$$W_j = \{p : p = q + a_j, q \in W_i\}$$

$$f(q) = f(q + a_j) + A_j, \quad \forall q \in W_i.$$

Then, the formation is invariant under the coordination algorithm.

The proof is a straightforward application of the following lemma, which shows the appealing translational invariance of the proposed control scheme. In other words, two locally similar configurations at different distances to the goal will give rise to the same control actions.

Lemma 4.6 (Control invariance): Consider two vehicles p_i and p_j with corresponding sets W_i and W_j of the same shape. Suppose their corresponding values of the navigation function only differ by a constant, i.e., suppose there exist $a \in \mathbb{R}^2$ and $A \in \mathbb{R}$ such that $W_j = \{p : p = q + a, q \in W_i\}$, $f(q) = f(q + a) + A$, $\forall q \in W_i$. Then the controls for vehicle i and j are equal:

$$u_i(k) = u_j(k).$$

Proof. Note that

$$\begin{aligned} M_{W_i} &= \int_{W_i} \phi(q) dq = \int_{W_i} e^{-kf(q)} dq \\ &= \int_{W_i} e^{-k(f(q+a)+A)} dq = \int_{W_j} e^{-k(f(q)+A)} dq \\ &= e^{-kA} \int_{W_j} e^{-kf(q)} dq = e^{-kA} M_{W_j}. \end{aligned}$$

Thus,

$$\begin{aligned} C_{W_i} &= \frac{1}{M_{W_i}} \int_{W_i} q\phi(q) dq = \frac{1}{e^{-kA} M_{W_j}} \int_{W_i} q\phi(q) dq \\ &= \frac{1}{e^{-kA} M_{W_j}} e^{-kA} \int_{W_j} (q-a)\phi(q) dq \\ &= \frac{1}{M_{W_j}} \left(\int_{W_j} q\phi(q) dq - aM_{W_j} \right) = C_{W_j} - a. \end{aligned}$$

Since the relative centroids are the same, the optimization problems yield the same result. Hence, $u_i(k) = u_j(k)$. ■

V. SIMULATION EVALUATION

In this section we illustrate how the proposed coordination algorithm performs in numerical simulations. First we study the case with integrator vehicle dynamics and then a more realistic setting with car-like vehicle dynamics. We illustrate group cohesion, goal convergence and obstacle avoidance in a setting with twenty vehicles.

A. Simulations with simple vehicle models

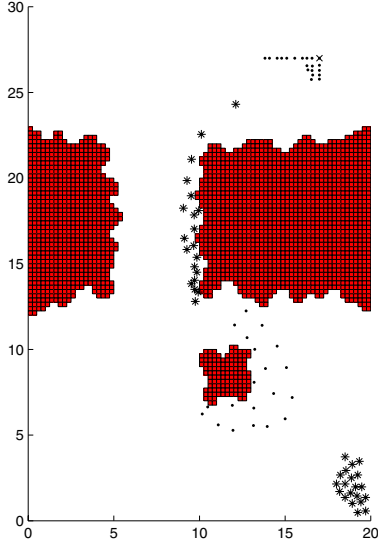


Fig. 3. Four snapshots of twenty vehicles moving around irregular obstacles. The goal is in the upper right corner. The parameters are set to $d = 1$, $R = 3$ and $k = 1$.

For the first simulation we chose irregular and non-convex obstacles to underline the advantage of not having

to make geometric assumptions about obstacle shape. In Figure 3 the desired inter-vehicle distance was set to $d = 1$, the sensing radius was $R = 3$ and ϵ was set to machine precision, i.e., roughly 10^{-16} on our system. The figure shows four snapshots. The vehicle positions are denoted by stars for the first and third snapshots and by points for the second and fourth snapshots. The twenty vehicles first perform a split-rejoin maneuver and then gather around the goal, marked by “x”. Due to the constraint in the optimization problem, the vehicles are distributed mainly in the third quadrant around the goal.

To explore the flocking behavior in detail, we ran a simulation in an open area without obstacles. Figure 4 shows an almost perfect hexagonal lattice resulting from the parameters $k = 1$, $d = 2$ and $R = 3$. The distances between vehicles agree well with the desired value $d = 2$.

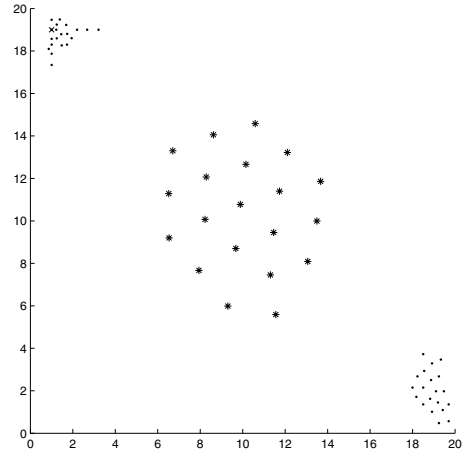


Fig. 4. Three snapshots of twenty vehicles moving on an open field. The goal is in the upper left corner. The parameters are set to $d = 2$, $R = 3$ and $k = 1$.

B. Simulations with detailed vehicle models

We have also implemented the multi-vehicle coordination algorithm in a realistic simulation environment that consists of both physical vehicles and a graphical computer simulator, sharing a common controller interface. The hardware consists of redesigned radio-controlled cars, see Figure 5(a).

Every car is equipped with a card-PC, actuators for thrust and steering, a WLAN transceiver for short-range communication, and a GPS receiver for navigation. The platform is rugged enough for outdoor terrain use, although the surface needs to be reasonably smooth. At the moment only two cars are operational, so to better demonstrate the flocking properties of our algorithm we have so far only implemented it in an animated computer-simulated environment. The environment has a high fidelity 18-DOF car model whose actuators have the same interface as the actuators of the physical cars. The information available to

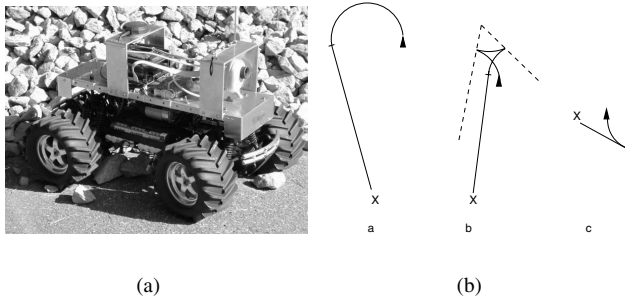


Fig. 5. (a) A redesigned radio-controlled car, which is the physical part of our test environment. The car is equipped with a card-PC, a GPS receiver and a short-range WLAN transceiver. (b) Three trajectories generated by the car controller. The triangle shows the initial position and orientation of the car and “x” shows the target location. In (b)_a the car has sufficient space to turn, while in (b)_b the dashed line shows the boundaries of the safe region, W_i , that must not be passed. In (b)_c the target is too close to reach in a left turn, so the car reverses before driving straight ahead.

the controller is also the same as in reality. The controller can extract the position of the car (although there is no error as in a GPS measurement). Inter-vehicle communication is simulated by using a global list of positions where every single vehicle can only get a list of its neighbors within a specified radius. A side-effect of having a GPS receiver on the car is that it has access to a very accurate time estimate. This is used to synchronize all cars and make them plan their next step simultaneously. This is crucial for avoiding ambiguities in the Voronoi partitioning of the available space.

We have used a control hierarchy where the lower control layer is responsible for achieving the position changes commanded by the coordination algorithm.

When implementing the lower layer, i.e. the car controller, some care had to be taken not to make the car leave the designated safe region W_i . (Recall that W_i is safe, since by construction all W_i are disjoint and contain no obstacles.) The three scenarios shown in Figure 5(b) have been identified and resolved.

Figure 6 shows a snapshot from the detailed simulations. A group of cars, graphically represented as the US Army HMMWV all-purpose vehicle, is simulated in a virtual environment. The obstacles are represented as ditches to enable overview of the whole field and to make collisions unrecoverable. All cars are tightly parked at the start, but when moving towards the goal they maneuver into the looser flock formation. Eventually they reach a stationary configuration, gathered around the goal.

VI. DISCUSSION

The problem of combined obstacle avoidance, navigation towards a goal point, and flocking is hard for two reasons. Firstly, there is often a tradeoff between the three subgoals



Fig. 6. Snapshot from a detailed multi-vehicle simulation. The last cars in the group are gathering at the entrance of a passage between two ditches. The goal is indicated by a small tree, in the far right part of the image.

of safety, reaching the goal point, and group cohesion. Secondly, the choice of obstacle representation is far from obvious. In this paper, we address the tradeoff by assigning the subgoals as constraints and cost function in an optimization problem. The second issue is addressed by using subsets of the configuration space for both obstacles and neighboring vehicles. The vehicle subsets and control actions were derived based on Voronoi partitions, navigation functions and mirror neighbors.

REFERENCES

- [1] O. Brock and O. Khatib. High-Speed Navigation Using the Global Dynamic Window Approach, *IEEE International Conference on Robotics and Automation*, Detroit, Michigan, May 1999.
- [2] P. Ögren and N. E. Leonard. A Convergent Dynamic Window Approach to Obstacle Avoidance. To appear in *IEEE Transactions on Robotics and Automation*.
- [3] R. Olfati-Saber and R. Murray. Distributed Cooperative Control of Multiple Vehicle Formations Using Structural Potential Functions, *IFAC World Congress*, Barcelona, Spain, July 2002.
- [4] N. E. Leonard and E. Fiorelli. Virtual Leaders, Artificial Potentials and Coordinated Control of Groups, *Proc. 40th IEEE Conference on Decision and Control*, 2001, Page(s): 2968–2973.
- [5] J.R.T. Lawton, B.J. Young and Randal W. Beard. A Decentralized Approach to Elementary Formation Maneuvers. *IEEE Transactions on Robotics and Automation*, 2003, pp. 933–941.
- [6] E. Rimon and D. Koditschek. Exact Robot Navigation Using Artificial Potential Functions. *IEEE Transactions on Robotics and Automation*. pp. 501-518, Vol. 8, No. 5, October 1992.
- [7] R.O. Saber and R.M. Murray. Flocking with Obstacle Avoidance: Cooperations with Limited Communication in Mobile Networks *IEEE Conference on Decision and Control*, 2003.
- [8] D.E. Chang, S.C. Shadden, J.E. Marsden and R.O. Saber. Collision Avoidance for Multiple Agent Systems *IEEE Conference on Decision and Control*, 2003.
- [9] H. Tanner Flocking with Obstacle Avoidance in Switching Networks of Interconnected Vehicles, *IEEE International Conference Robotics and Automation*, pp. 3006-3011, New Orleans LA, April 26-May 1, 2004.
- [10] P. Ögren and N. E. Leonard. Obstacle Avoidance in Formation, *2003 IEEE International Conference Robotics and Automation Taipei*, Taiwan, September 2003.
- [11] T. Li and H. Chou, Motion Planning for a crowd of robots, *2003 IEEE International Conference Robotics and Automation Taipei*, Taiwan, September 2003.
- [12] J. Cortes, S. Martinez, T. Karatas, F. Bullo, Coverage Control for Mobile Sensing Networks *IEEE Transactions on Robotics and Automation*. Vol. 20, April 2004.
- [13] T. Balch and R. Arkin, Behaviour-Based Formation Control for Multirobot Teams *IEEE Trans. on Robotics and Automation*, Vol. 14, No.6, Dec 1998.