Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Decentralized Cooperative Collision Avoidance for Acceleration Constrained Vehicles

Gabriel M. Hoffmann and Claire J. Tomlin

Abstract— **Safety must be ensured in the deployment of multi-agent vehicle systems. This paper presents decentralized collision avoidance algorithms for systems with second order dynamics and acceleration constraints, using a switching control law. The technique augments existing multi-agent control laws with the capability to switch to provably safe collision avoidance maneuvers when required. Two algorithms with low computational cost are presented, one for two vehicles and one for more vehicles. In both methods, each vehicle computes avoid sets with respect to every other vehicle. When one or more vehicles are on the boundary of their avoid sets, collision avoidance action is taken. These algorithms are applied in simulation scenarios for which existing techniques either fail or are computationally expensive, and used for information theoretic control of a mobile sensor network to reduce the computational complexity. Finally, they are demonstrated in quadrotor helicopter flight experiments.**

I. Introduction

Advances in technology are making possible the deployment of multi-agent systems of vehicles for a variety of applications. It is desirable to distribute decision making to individual agents for scalability. However, the requirement to ensure safety by avoiding collisions has proven challenging to distribute. This paper presents a decentralized cooperative switching control law for collision avoidance between cooperative mobile agents with simplified rotorcraft dynamics: second order dynamics with acceleration constraints.

A motivating application is mobile sensor networks, such as the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC) [1]. They can deliver exciting new capabilities in surveillance, reconnaissance, and scientific discovery through their ability to move sensors to vantage points rich in information [2], [3], [4]. However, as the number of vehicles increases, safety becomes challenging, even for human pilots, as shown anecdotally in Fig. 1.

Related work has used rule- and optimization-based approaches. One rule-based approach is potential methods, such as algorithms for sensor network control [2] and for systems with second order dynamics [5], [6]. These approaches do not consider control constraints. Switching rules have been proposed for decentralized control, without collision avoidance guarantees [7]. For optimization-based approaches, one method is an iterative distributed optimization formulated [8] and another is a centralized nonlinear program [9], though computational expense scales poorly with the number of vehicles. Another approach uses a mixed integer linear program (MILP) with constraints on dynamics [10], [11]. Decentralized control has been formulated using the dual, without collision avoidance [12]. Decentralized nonlinear model predictive control was formulated using potential functions for collision avoidance, without guarantees [13].

Fig. 1. A mid-air collision between two out of three remote-piloted quadrotor helicopters operating in close proximity. The proposed algorithm would assume control when required to prevent imminent collisions.

The algorithm presented here uses a rule-based approach derived using optimal control and reachability analysis [14]. It provides collision avoidance, or acts as an added layer of safety, for higher level logic. The goal is to be minimally invasive and only affect control inputs when required to avoid collision. Hence, trajectories are likely suboptimal when a collision avoidance action is taken, however when no collision is imminent, the algorithm does not interfere.

Two control laws are presented, one for two vehicles, and one for n_v vehicles. In both, the vehicles compute analytical avoid set boundaries with respect to each other vehicle. When any vehicles are on the boundary of their avoid sets, collision avoidance action is taken. In the two vehicle scheme, a boundary is computed using optimal control that is proven safe analytically, and computed with trigonometric functions. The optimal action is computed numerically. Run time for two simulated vehicles was 0.2 ms. In the n_v vehicle scheme, pairwise avoid sets can be computed by linear algebra. A control law is presented with safety proven analytically for three vehicles, and validated for $n_v > 3$ in simulation and analysis of scenarios. Computational complexity is $O(n_v)$. Run time for 200 interacting simulated vehicles was 9.2 ms. These control laws are applied in simulations of 2 to 200 vehicles, and to the mobile sensor network control problem of [4] where they yield no discernible change in performance, but improve run time over previous methods. Finally, they are demonstrated in quadrotor helicopter flight experiments.

This research was supported by ONR under the CoMotion MURI contract N00014-02-1-0720, and in part by AFOSR under the Multi-University Research Initiative (MURI) grant #FA9550-06-0312.

G. M. Hoffmann is with the Department of Aeronautics and Astronautics, Stanford University; gabeh@stanford.edu.

C. J. Tomlin is with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley; tomlin@eecs.berkeley.edu.

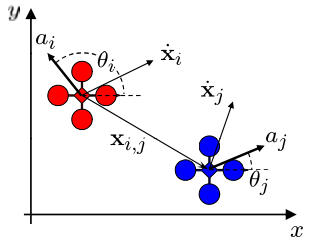


Fig. 2. Relative states of two acceleration constrained quadrotor helicopters. The control inputs for aircraft i and j are accelerations $a_i \in [0, a_{max}]$ and $a_j \in [0, a_{max}]$ in directions θ_i and θ_j, respectively.

II. Problem Formulation and Approach

Consider a set of n_v vehicles where the state of the i^{th} vehicle is $\mathbf{x}_i = [\ x_i \quad y_i \quad \dot{x}_i \quad \dot{y}_i\ ]^T$. Note this formulation neglects possible motion in $\mathbb{R}^3$. The methods developed here can be extended to such scenarios, though it is often desirable to restrict the dimension due to operational concerns.

The vehicles are modeled to have undamped second order dynamics with acceleration control inputs $\mathbf{u}_i = [\ \theta_i \quad a_i\ ]^T$, where the acceleration direction is $\theta_i \in [0, 2\pi)$ and the magnitude is $a_i \in [0, a_{max}]$. This model was found in experiments to adequately approximate the dynamics of quadrotor helicopters[15], and is similar to that of typical rotorcraft.

Define the relative state of vehicle j with respect to i as $\mathbf{x}_{i,j} = \mathbf{x}_j - \mathbf{x}_i$, with distance $d_{i,j} = \sqrt{x_{i,j}^2 + y_{i,j}^2}$ between them, shown in Fig. 2. The collision avoidance requirement is that

$$d_{i,j} \geq d_{min} \ \forall\ \{i, j | i \in [1, n_v],\ j \in [1, n_v],\ j \neq i\} \qquad (1)$$

where d_{min} is the minimum allowed distance between vehicle centers. Define the speed of i to be $v_i = \sqrt{\dot{x}_i^2 + \dot{y}_i^2}$ and the relative speed to be $v_{i,j} = \sqrt{\dot{x}_{i,j}^2 + \dot{y}_{i,j}^2}$. The equations of motion of the relative dynamics for any pair of vehicles are

$$f(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}) = \frac{\partial}{\partial t} \begin{bmatrix} x_{i,j} \\ y_{i,j} \\ \dot{x}_{i,j} \\ \dot{y}_{i,j} \end{bmatrix} = \begin{bmatrix} \dot{x}_{i,j} \\ \dot{y}_{i,j} \\ -a_i \cos\theta_i + a_j \cos\theta_j \\ -a_i \sin\theta_i + a_j \sin\theta_j \end{bmatrix} \qquad (2)$$

where $\mathbf{u}_{i,j} = [\ \mathbf{u}_i^T \quad \mathbf{u}_j^T\ ]^T$.

The cooperative switched control law is an adaptation of the non-cooperative pursuit-evasion game [14]. The goal for safe operation is to keep $\mathbf{x}_{i,j}$ out of "keepout set" K, defined by (1). Pairwise loss function $l(\mathbf{x}_{i,j})$ is defined such that

$$K = \{\mathbf{x}_{i,j} \in \mathbb{R}^4 | l(\mathbf{x}_{i,j}) < 0\} \text{ and } \partial K = \{\mathbf{x}_{i,j} \in \mathbb{R}^4 | l(\mathbf{x}_{i,j}) = 0\} \qquad (3)$$

Set $Pre_t(K) \subset \mathbb{R}^4$ can be computed from which the control strategy causes a vehicle to enter K in at most t time. The problem is formulated as a two-person, zero-sum dynamical game, where the "losing" states are calculated for the vehicles, with value function $J(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}, t) = l(\mathbf{x}_{i,j}(0))$. This is the cost of a trajectory that starts at $t \leq 0$ and ends at $\mathbf{x}_{i,j}(0)$.

First, define the *unsafe* portion of ∂K, the boundary of K, as states for which the control strategy results in the state flowing into K. The outward normal to K is $\nu^T = Dl(\mathbf{x}_{i,j})$, hence the unsafe portion is where $\nu^T f(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}) < 0$.

The optimal control inputs for an evader are computed by maximizing $l(\mathbf{x}_{i,j}(0))$. It is successful if $l(\mathbf{x}_{i,j}(0)) \geq 0$. The optimization can be posed in terms of Hamilton's equations. To maximize $J(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}, t)$, the controller maximizes [14]

$$\frac{\partial J}{\partial t} = \left(\frac{\partial J}{\partial \mathbf{x}_{i,j}}\right)^T \frac{\partial \mathbf{x}_{i,j}}{\partial t} = \mathbf{p}^T f(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}) \qquad (4)$$

where the costate, $\mathbf{p} = \nabla_{\mathbf{x}_{i,j}} J(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}, t)$, with elements $p_k : k \in [1,4]$. At $t = 0$, the game terminates, so $\mathbf{p}(0)$ is the gradient of $J(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}, t)$ at the terminal condition, $l(\mathbf{x}_{i,j}(0))$. By maximizing $\mathbf{p}^T f(\mathbf{x}_{i,j}, \mathbf{u}_{i,j})$, with respect to the control inputs, the rate of retreat from the next level set can be controlled. Level sets can be back-propagated from ∂K.

An analytic solution is found in Section III for optimal control inputs for a region of minimum size. For more than two vehicles, computing the reachable set becomes expensive. Consequently, a suboptimal control law is presented in Section IV, inspired by the optimal control law, that has low computational overhead, and is trivially decentralized.

III. Two Vehicle Collision Avoidance

This section presents the optimal switching control law for two vehicle collision avoidance with acceleration constraints. First, the optimal control inputs are derived. Then, the set of relative states for which $l(\mathbf{x}_{i,j}(0)) \leq 0$ is found, the "avoid set", $A_{i,j}$. Its boundary, $\partial A_{i,j}$, is the surface where vehicles transition from nominal control to collision avoidance.

A. Optimal Control Input

The final time for the game is the time of closest approach, at $t = 0$. The cost function is,

$$J(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}, t) = l(\mathbf{x}_{i,j}(0)) = d_{i,j}^2 - d_{min}^2 \qquad (5)$$

The objective function, the Hamiltonian, is the rate of change of $J(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}, t)$, (4). Expanding it yields

$$\begin{aligned} H(\mathbf{x}_{i,j}, \mathbf{p}) &= \mathbf{p}^T f(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}) \\ &= p_1 \dot{x}_{i,j} + p_2 \dot{y}_{i,j} + p_3 (-a_1 \cos\theta_i + a_2 \cos\theta_j) \\ &\quad + p_4 (-a_1 \sin\theta_i + a_2 \sin\theta_j) \end{aligned}$$

The optimization problem for evasion-evasion is

$$\begin{aligned} &H^*(\mathbf{x}_{i,j}, \mathbf{p}) = \\ &\max_{\mathbf{u}_1} . \max_{\mathbf{u}_2} . ((p_1 \dot{x}_{i,j} + p_2 \dot{y}_{i,j}) - a_1 (p_3 \cos\theta_i + p_4 \sin\theta_i) \\ &\qquad + a_2 (p_3 \cos\theta_j + p_4 \cos\theta_j)) \end{aligned} \qquad (6)$$

To solve for each player separately, note that at the extrema of the objective, $a_1 = a_2 = a_{max}$. The derivative is taken with respect to the remaining control inputs. For vehicle j,

$$\frac{\partial}{\partial \theta_j} (p_3 \cos\theta_j + p_4 \sin\theta_j) = -p_3 \sin\theta_j + p_4 \cos\theta_j \qquad (7)$$

The extrema, then, are the solutions to

$$-p_3 \sin\theta_j^* + p_4 \cos\theta_j^* = 0 \qquad (8)$$

Thus,

$$\theta^* = \arctan\left(\frac{p_4}{p_3}\right) + n\pi \qquad (9)$$

From the calculus of variations,

$$\dot{\mathbf{p}} = -\frac{\partial}{\partial \mathbf{x}_{i,j}} H^*(\mathbf{x}_{i,j}, \mathbf{p}) = [\ 0 \quad 0 \quad -p_1 \quad -p_2\ ]^T \qquad (10)$$

and

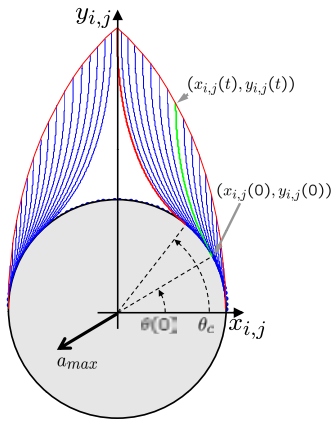


Fig. 3. The keepout set (gray) and the avoid set (white) for two-vehicle collision avoidance, in the rotated coordinate frame. The parabolas are the trajectories that a vehicle follows from any point on the boundary. The avoid set evolves as the parabola is followed such that its boundary is not crossed.

$$\mathbf{p}(0) = \frac{\partial}{\partial \mathbf{x}_{i,j}} J(\mathbf{x}_{i,j}(0)) = \begin{bmatrix} 2x_{i,j}(0) \\ 2y_{i,j}(0) \\ 0 \\ 0 \end{bmatrix} \tag{11}$$

Then, integrating to find the optimal control inputs yields

$$\mathbf{p}(t) = \begin{bmatrix} 2x_{i,j}(0) \\ 2y_{i,j}(0) \\ -\int_t^0 p_1 dt \\ -\int_t^0 p_2 dt \end{bmatrix} = \begin{bmatrix} 2x_{i,j}(0) \\ 2y_{i,j}(0) \\ -tx_{i,j}(0) \\ -ty_{i,j}(0) \end{bmatrix} \tag{12}$$

Substituting (12) into (9) yields the optimal input,

$$\theta_j^* = \arctan\left(\frac{y_{i,j}(0)}{x_{i,j}(0)}\right) + n\pi \tag{13}$$

Substituting (13) into (6) it is found that $n = 1$ maximizes the optimization. Thus, it is optimal to accelerate away from the point of closest approach on ∂K.

B. Avoid Set

The boundary of the avoid set is the locus of points along which collision avoidance control is required. The collision avoidance control input is constant acceleration perpendicular to ∂K at the point of closest approach, so the path in relative coordinates is a parabola, depicted in Fig. 3.

For analysis, a change of coordinates is performed. The change of coordinates rotates the relative coordinate frame such that in the rotated frame, $\dot{x}_{i,j} = 0$ and $\dot{y}_{i,j} < 0$. Define the rotation angle of the avoid region, with respect to the relative coordinate frame, to be[1]

$$\phi_{i,j} = \text{atan2}\left(\dot{y}_{i,j}, \dot{x}_{i,j}\right) - \frac{\pi}{2} \tag{14}$$

where the $\frac{\pi}{2}$ offset orients the avoid set along the $y_{i,j}$ axis for analysis; an arbitrary choice. Analysis of the avoid set for any $\phi_{i,j}$ can be performed in the coordinates shown in Fig. 3, removing the need to consider $\dot{x}_{i,j}$ in this frame.

For the remainder of this section, the relative coordinate frame is rotated by $-\phi_{i,j}$. To avoid over complicating the notation, the same variable names are used for the rotated frame; to return results to the original coordinate frame, they must be rotated by $\phi_{i,j}$, as depicted in the results in Fig. 5.

[1]As in ANSI-C, atan2(y,x) is similar to arctan$\left(\frac{y}{x}\right)$, but by using the signs of x and y, it returns the angle in the domain $(-\pi, \pi]$ rather than $(-\frac{\pi}{2}, \frac{\pi}{2}]$.

Boundary $\partial A_{i,j}$ of the avoid set can be found such that use of the optimal control input results in a closest approach of d_{min}, shown in Fig. 3. The conditions defining the rotated frame are used: $\dot{x}_{i,j} = 0$, $\dot{y}_{i,j} = -v_{i,j}$. The parabola is rotated by $\theta(0) - \pi/2$, and has a second derivative of $2a_{max}$ in that direction due to collision avoidance action by both vehicles. Thus, the usable part of $\partial A_{i,j}$ is defined by,

$$x_{i,j}(t) = d_{min}\cos\theta(0) - \frac{v_{i,j}(t)^2}{2a_{max}}\sin^2\theta(0)\cos\theta(0) \tag{15}$$

$$y_{i,j}(t) = \left(d + \frac{v_{i,j}(t)^2}{a_{max}}\right)\sin\theta(0) - \frac{v_{i,j}^2}{2a_{max}}\sin^3\theta(0) \tag{16}$$

To numerically test if vehicle j has crossed the boundary, the region can be approximated by a polygon with vertices generated using discrete values of $\theta(0)$ from 0 to the critical angle, θ_c, and then mirroring about the $y_{i,j}$-axis. The angle θ_c is the one past which no points of closest approach occur, given the use of optimal control. It can be shown that $\theta_c < \frac{\pi}{2}$ if $2d_{min}a_{max} < v_{i,j}^2$. To find θ_c, (15) is solved with $x_{i,j} = 0$,

$$\theta_c = \begin{cases} \arcsin\sqrt{\frac{2d_{min}a_{max}}{v_{i,j}(t)^2}} & \text{if } 2d_{min}a_{max} < v_{i,j}^2 \\ \pi/2 & \text{otherwise} \end{cases} \tag{17}$$

Then, $(x_{i,j}, y_{i,j})$ can be tested to see if it is in the polygon.[2] If it is, then collision avoidance action must be taken.

To find the optimal control, the point of closest approach must be found. This can be found by solving (16) for $\theta(0)$ using the current value of $y_{i,j}$. Note that by substituting $\zeta = \sin\theta(0)$, this is cubic function, with roots easily found numerically; the only physical root is in the domain $[-1, 1]$.

Note that synchronous control is not required, though to strictly avoid entering K, $A_{i,j}$ must be grown to account for all possible control inputs. When implemented in discrete time, the control law must test if the vehicles are at $\partial A_{i,j}$, which requires crossing the boundary of the avoid set by at most $v_{i,j}\Delta t$, where Δt is the discrete time step. To guarantee strict avoidance, d_{min} must be increased by this quantity.

The optimal collision avoidance action is to accelerate away from $\theta(0)$, or in the original relative coordinate frame, $\theta(0) + \phi_{i,j}$. This control law guarantees that no collisions will occur between two vehicles, and is required for a minimum portion of the physical area. However, to address many interacting vehicles, further development is required.

IV. Multi-Vehicle Collision Avoidance

A. Optimal Control Input

Following the same methods used to derive (13), more vehicle interactions can be included, yielding the optimal control law for many vehicles,

$$\theta_i^* = \arctan\left(\frac{y_{i,1}(0) + y_{i,2}(0) + \ldots}{x_{i,1}(0) + x_{i,2}(0) + \ldots}\right) + \pi \tag{18}$$

However, it is computationally expensive to find $\mathbf{x}_{i,j}(0)$ for $n_v > 2$, so this control law will not be used used here. Rather, it is used to inspire a proposed alternate control strategy,

[2]Standard algorithms can test if $(x_{i,j}, y_{i,j})$ is in the polygon, though here it is more efficient to use a cross product test with the points on the boundary horizontally closest to $x_{i,j}$.

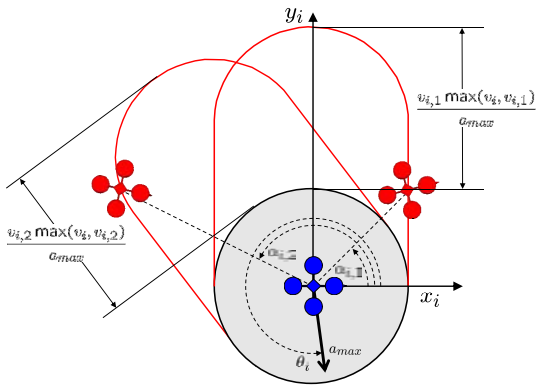


Fig. 4. Vehicle v accelerates at a_{max} in direction θ_v to avoid vehicles 1 and 2, which are at the edge of their respective avoid sets. The orientation of the avoid set is that of the relative velocity vectors.

$$\theta_i = \arctan\left(\frac{y_{i,1}(t)+y_{i,2}(t)+\ldots}{x_{i,1}(t)+x_{i,2}(t)+\ldots}\right)+\pi \tag{19}$$

where the vehicles considered are those at the boundaries of their respective avoid sets, $A_{i,j}$. Again, use $a_i = a_{max}$. The control law is equivalent to accelerating away from the centroid of all vehicles that must be actively avoided. Additional logic is required as described in Section IV-B.

This control law is suboptimal in the sense that its use may be required in regions for which another collision avoidance control law would not require action. However, it yields enormous computational savings, and will be shown to yield collision avoidance using a reasonably small avoid set. Now the avoid sets must be found.

B. Avoid Set

Consider the proposed separate avoid sets for each vehicle shown in Fig. 4, inspired by the avoid set for two vehicle collision avoidance, again oriented opposite the direction of relative velocity. The length of the proposed region is

$$L_{i,j} = \frac{v_{i,j}\max(v_i, v_{i,j})}{a_{max}} \tag{20}$$

in order to maintain the minimum separation distance, as shown in the remainder of the section. This avoid set is designed such that the control law provably works for two and three vehicles. It is validated analytically and numerically for $n_v > 3$ in selected configurations assumed to be most challenging to the algorithm.

This section analyzes the dynamics of vehicles on their respective $\partial A_{i,j}$ to validate that they do not cross into $A_{i,j}$. Because $K \subset A_{i,j}$, this validates that no vehicle can enter K.

1) Behavior on avoid set straight-edge boundaries: To analyze each of the $n_v - 1$ avoid sets for vehicle i, again a change of coordinates rotates the relative coordinate frame by $\phi_{i,j}$, as defined in (14). The rotation rate of this frame of reference is found by differentiating, using (2), and using the fact that $\dot{x}_{i,j} = 0$ in this rotated frame.

$$\dot{\phi}_{i,j} = \frac{\dot{y}_{i,j}}{v_{i,j}^2}(-a_i\cos\theta_i + a_j\cos\theta_j) \tag{21}$$

Therefore, if vehicle j is on a straight edge of $\partial A_{i,j}$, to guarantee collision avoidance, it must accelerate away from that edge more than vehicle i accelerates toward it. For two vehicles, this is clearly the case; the vehicles accelerate away from the centroid, yielding the required behavior. For three vehicles, this is achieved by adding additional logic to accelerate directly away from the interior when an edge is crossed. For more than three vehicles, the same logic is applied. In this case, the number of permutations of active constraints makes an analytical proof of safety difficult: simulation results were used to validate safety of this algorithm along the straight-edged boundary.

2) Behavior on avoid set curved boundaries: From (20), $L_{i,j} \geq v_{i,j}^2/a_{max}$. In the rotated frame, $\dot{L}_{i,j} \leq 2\dot{y}_{i,j}\ddot{y}_{i,j}/a_{max}$. To prove that $d_{i,j} \geq d_{min}$ is not violated, it is sufficient to show that $\dot{L}_{i,j} \leq \dot{y}_{i,j}$ because when $L_{i,j} = 0$, $A_{i,j} = K$. Using (2), this is equivalent to

$$\sin\theta_j \geq \sin\theta_i + 1/2 \tag{22}$$

This is clearly satisfied by substituting in (19) for two or three vehicles; at worst for three vehicles, one vehicle is directly between two other vehicles, in which case margin still exists in the above inequality. For $n_v > 3$, again the permutations of active constraints make an analytical proof of safety difficult. Analytical verification for challenging scenarios is presented in Section IV-B.4.

3) Behavior for combined boundary types: For three vehicles, with vehicle 1 on the straight-edge boundary of $A_{i,1}$ and vehicle 2 on the curved boundary of $A_{i,2}$, the vehicle on the straight edge is alone capable of ensuring that the rotation rate of the avoid set given by (21) rotates the set away from the vehicle, or at least keeps the rotation angle stationary. The vehicle on the curved boundary is capable of satisfying (22) provided vehicle i does not accelerate toward it. This final condition is satisfied by design of the discrete time implementation to prevent chattering effects. As mentioned previously, the boundary must be crossed to be detected for numerical implementation. Rather than chatter arbitrarily, vehicle i uses the input corresponding to the boundary crossed by the largest distance, guaranteeing separation in the three vehicle scenario. Again, for $n_v > 3$, simulation is used to validate safety for this configuration.

4) Analysis for more than three vehicles: Consider two configurations with potential to cause failure: a large circle of vehicles converging to one point, and a large line of vehicles converging to one point. The algorithm is proven analytically to guarantee that separation is maintained in both cases.

To analyze the converging circle, let d be the spacing between adjacent vehicles with initial speeds v. To find the initial d at time t_0 such that the vehicles can apply a_{max} to stop by $t = 0$ with $d(0) = d_{min}$, start by computing relative speed. The angle between the velocity vectors for n_v vehicles equally spaced on a circle is $2\pi/n_v$. For adjacent vehicles,

$$v_{i,j} = \dot{d}(t_0) = -2v\sin(\pi/n_v) \tag{23}$$

When the vehicles begin to apply a_{max}, $\dot{v} = -a_{max}$, so $\ddot{d} = 2a_{max}\sin(\pi/n_v)$. Integrating this and using (23),

$$\dot{d}(t) = -2v(t_0)\sin(\pi/n_v) + 2a_{max}t\sin(\pi/n_v) \tag{24}$$

The rate of change of spacing at the point of closest approach is zero, so by setting (24) to zero it is found that $t_f = v(t_0)/a_{max}$. Integrating once more,

$$d(t) = d(t_0) - 2v(t_0)t\sin(\pi/n_v) + a_{max}t^2\sin(\pi/n_v) \tag{25}$$

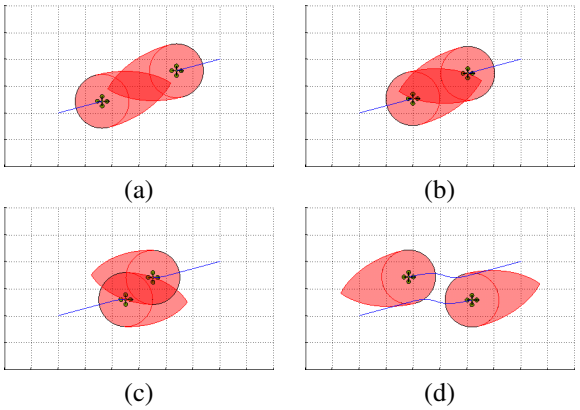


Fig. 5. Simulation of two quadrotor helicopters approaching one another tracking trajectories (a). When the vehicles (b) touch each other's avoid set they (c) apply collision avoidance control inputs. After the conflict is resolved, they (d) resume their previous trajectories.

TABLE I

COLLISION AVOIDANCE COMPUTATION TIME

Algorithm	# Vehicles	Time (ms)
Two-Veh.	2	0.23
Many-Veh.	2	0.16
Many-Veh.	8	0.41
Many-Veh.	32	1.6
Many-Veh.	64	3.6
Many-Veh.	100	4.6
Many-Veh.	200	9.2

Setting $d(t_f) = d_{min}$,

$$d(t_0) = d_{min} + \frac{v(t_0)^2}{a_{max}} \sin(\pi/n_v) \tag{26}$$

Therefore, violation is avoided if

$$L \geq \frac{v(t_0)^2}{a_{max}} \sin(\pi/n_v) = \frac{v_i v_{i,j}}{2a_{max}} \tag{27}$$

This length is no more than half that of (20), hence the proposed control law is guaranteed to maintain $d > d_{min}$.

To analyze the converging line of vehicles, consider vehicles $i = 1$, $j = 2$, with $y_{1,2} = 0$, $x_1 > x_2$, $x_1 > 0$, and $\dot{x}_1 > \dot{x}_2$. Then, $\dot{L}_{1,2} = \frac{\dot{x}_1(\dot{x}_1 - \dot{x}_2)}{a_{max}}$, so

$$\dot{L}_{1,2} = \frac{1}{a_{max}} (2\dot{x}_1\ddot{x}_1 - \ddot{x}_1\dot{x}_2 - \ddot{x}_2\dot{x}_1) \tag{28}$$

To guarantee collision avoidance, it is sufficient for this scenario that $\dot{L}_{1,2} \leq \dot{x}_2 - \dot{x}_1$. The inequality becomes

$$\ddot{x}_2 \geq a_{max}(1 - \frac{\dot{x}_2}{\dot{x}_1}) + \ddot{x}_1(2 - \frac{\dot{x}_2}{\dot{x}_1}) \tag{29}$$

If $\ddot{x}_1 = -a_{max}$,

$$\ddot{x}_2 \geq -a_{max} \tag{30}$$

which is true due to constraints. Therefore, one vehicle following another always has the ability to stop by applying maximum acceleration away from that vehicle using (20). Thus, any string of vehicles following one another can stop.

V. RESULTS

The proposed control laws were tested in simulation and flight experiments. All simulated vehicles are quadrotor helicopters with second order dynamics and ℓ^2-norm constraints on acceleration. Simulations were run in Matlab using one core of a Core Duo 2.16 GHz. The number of vehicles

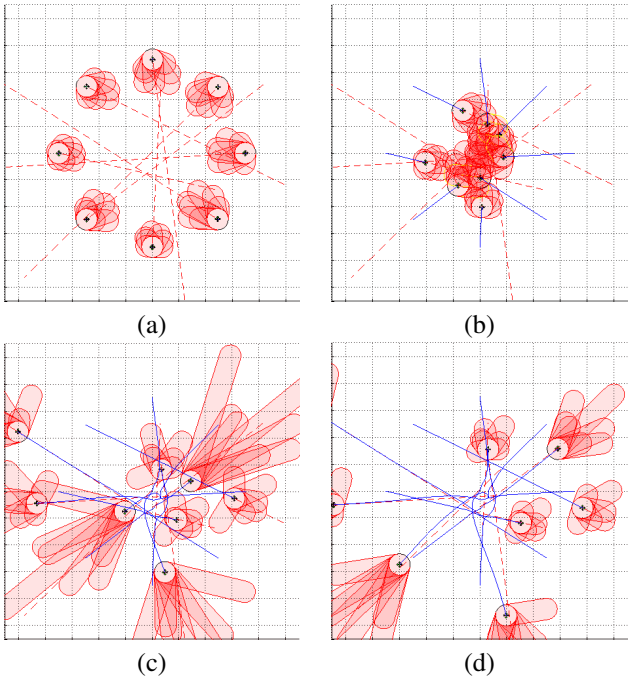


Fig. 6. Simulation of 8 vehicles tracking randomly generated trajectories, using the many-vehicle collision avoidance law. The keepout sets are circles, and shaded long red regions are avoid sets. Only a few regions touch the vehicles to which they correspond, leading to collision avoidance action, as highlighted in (b).

demonstrated would be computationally overwhelming for other methods known to the authors. The timing results for the simulations are shown in Table I.

Flight experiments used STARMAC II quadrotor helicopters tracking attitude commands from human pilots [15], validating the results from simulations. Note that the collision avoidance algorithm does not prevent deadlock. However, deadlock was most severe when relative states have high symmetry to numerical precision–an improbable condition in real systems, as observed in flight experiments.

A. Simulation of Two Vehicles

The two vehicle collision avoidance algorithm was used for two trajectory tracking vehicles flown toward each other with a variety of crossing paths, speeds, and acceleration constraints. The circular set K and avoid set $A_{i,j}$ are shown in Fig. 5. As the control action is taken, relative velocities change, causing the avoid sets to morph, keeping the vehicles on one another's boundaries. The vehicles never enter the avoid set; avoidance action is required while the vehicles are on one another's boundaries. Then, they resume line tracking.

B. Simulation of Many Vehicles

Sets of 2 to 200 vehicles were simulated doing trajectory tracking using many-vehicle collision avoidance on a variety of collision-courses. One scenario is in Fig. 6, with the separation distance between each vehicle in Fig. 7. The vehicles navigate past one another, with deadlocks resolved by asymmetries when the trajectory tracking control is allowed to resume. Timing from these simulations, in Table I, shows that run time at each vehicle was approximately $n_v \times 0.05$ ms.

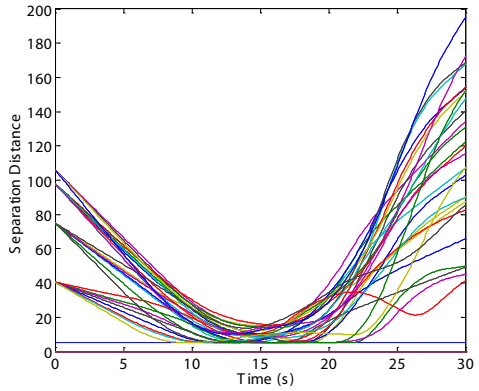
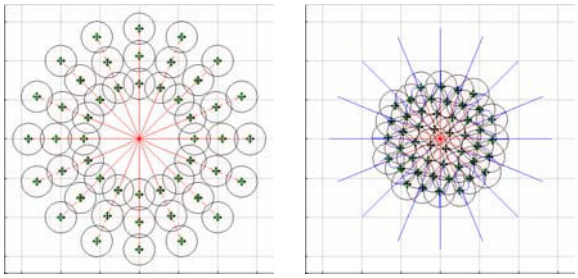


Fig. 7. Separation distances between each vehicle for the eight vehicle simulation in Fig. 6, and a line showing the minimum allowed distance, $d_{min} = 2$. Separation is maintained throughout the simulation.



(a) (b)

Fig. 8. Simulation of 3 rings of 16 vehicles converging toward the same point, with the outermost rings moving fastest, (a) at $t = 0$ and (b) at $t = 10$. Many avoid sets are active for each vehicle due to range of speeds of the rings.

Many challenging scenarios were simulated to validate performance in unreasonably complicated situations. One such case shown in Fig. 8 has three rings of 16 vehicles converging, with the outermost rings traveling faster. The $n_v - 1$ avoid sets at each vehicle are omitted from the plot for clarity. Collision avoidance must prevent lines of vehicles from piling up, and rings of vehicles being wedged to together. Spacing is maintained as shown in Fig. 7. Due to numerical precision, the vehicles are pushed off of their trajectories and approach their destination, the origin.

Another challenging case is a large circle of vehicles converging, shown in Fig. 10, where the separation was maintained, as computed in Section IV-B.4. Vehicles eventually slide past one another. In both the 48 and 64 vehicle scenarios, deadlock was observed, though it was eventually resolved due to machine precision. When deadlocked, separation was maintained, as was the case for all simulations.

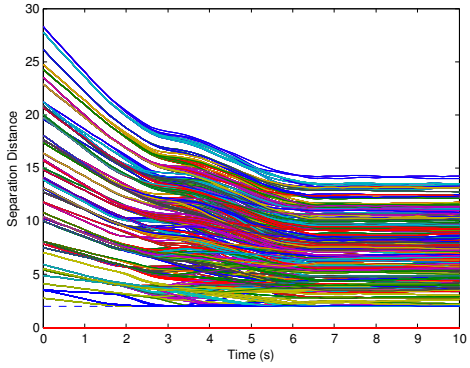


Fig. 9. Separation distances between each vehicle for the 48 vehicle simulation in Fig. 8, and a line at the minimum allowed distance, $d_{min} = 2$. Separation is maintained throughout the simulation.

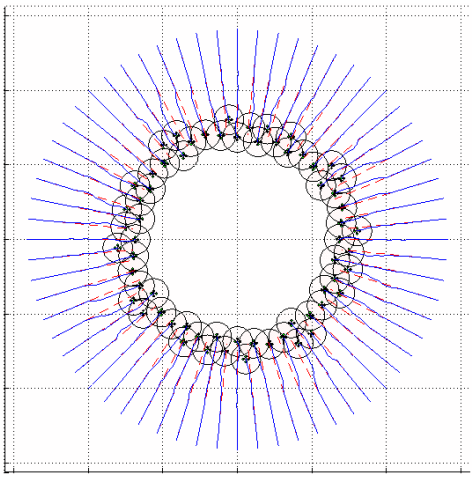
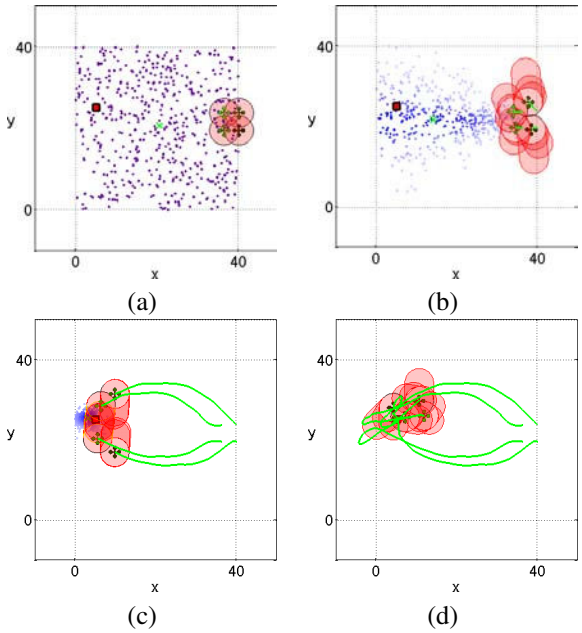


Fig. 10. Simulation of a ring of 64 vehicles converging toward a center. The many-vehicle collision avoidance algorithm safely prevents the vehicles from being wedged together.



(a) (b) (c) (d)

Fig. 11. Simulation of a bearings-only target search with 4 mobile sensors (quadrotor helicopters), using the particle filter distribution to compute mutual information and the pairwise-node approximation for distributed control. By using the collision avoidance method presented here, the optimization can be fully decentralized. Experimentally, the optimization was not adversely effected, and run time greatly decreased.

C. Mobile Sensor Network Control

Mobile sensor networks were used to do autonomous information seeking in [4]. It would be desirable to eliminate the need for distributed iterative optimization and improve on the ε-feasible guarantee for collision constraint satisfaction. The many-vehicle switching control law was used to replace

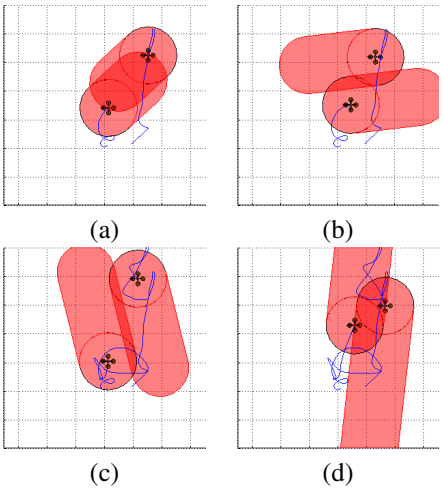


Fig. 12. Automatic collision avoidance using $d_{min} = 2$ m while human control inputs attempt to cause collisions. (a) A conflict is detected at $t =$ 25 s and (b) recovered from by $t = 26$ s. (c) The aircraft approach at $t = 35$ s, resulting in (d) the conflict at $t = 36$ s.

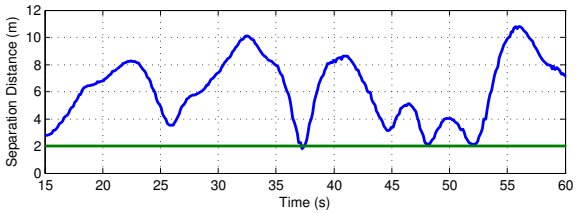


Fig. 13. Separation according to GPS data for the flight experiment shown in Fig. 12. Even without extending d_{min} to account for time discretization, at $v_{1,2} = 5\frac{m}{s}$, d_{min} was violated by 0.1 m, less than $v_{1,2}\Delta t$.

the distributed optimization with a fully decentralized one. Only vehicle states and observations need be communicated. Local optimizations are performed to estimate the vehicle's ability to contribute to the global information of the system.

Simulations performed well, as shown in Fig. 11 for a bearings-only target search simulation. It was found that the performance of the decentralized optimization, as measured by decrease in entropy, matched the distributed iterative optimization, even though collision avoidance regions use suboptimal control inputs when encroached. Simulations demonstrated consistent collision avoidance, fast localization of the search target, and reductions computation time.

D. Flight Experiments

The many-vehicle algorithm was implemented in the on-board computers of the STARMAC testbed. The software on each quadrotor broadcasts vehicle states to all other quadrotors at 10 Hz. The algorithm uses $d_{min} = 2$ m, $a_{max} = 1.7\frac{m}{s^2}$, and runs on the 600 MHz PXA270 on the aircraft.

The vehicles nominally used either attitude reference commands from a human pilot, or a waypoint tracking controller. The human pilots issued malicious commands to attempt to instigate collisions, which the collision avoidance algorithm successfully prevented, as shown in Figs. 12 and 13.

VI. Conclusion

Decentralized collision avoidance algorithms were presented for systems with second order dynamics and acceleration constraints. By using a switching control law with easily computed avoid sets, the control laws provide safety while using any desired external control law. Two control laws were presented, one for two vehicles and one for many vehicles. These control laws were applied in simulations of scenarios for which existing techniques either fail or are computationally expensive. The techniques were used to simplify the distributed optimization for information theoretic mobile sensor network control, greatly reducing computational complexity. Flight experiments using STARMAC validate the results of simulations with real-time collision avoidance between aircraft. For future work, it would be interesting to analyze the effect of communication delays, of different capabilities between vehicles, and of speed constraints.

References

[1] Hoffmann, G. M., Huang, H., Waslander, S. L., and Tomlin, C. J., "Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment," *Proc. AIAA Guidance, Navigation, and Control Conf.*, Hilton Head, SC, August 2007.

[2] Ogren, P., Fiorelli, E., and Leonard, N. E., "Cooperative Control of Mobile Sensor Networks: Adaptive Gradient Climbing in a Distributed Environment," *IEEE Trans. Automat. Contr.*, Vol. 49, No. 8, 2004, pp. 1292–1302.

[3] Grocholsky, B., Makarenko, A., Kaupp, T., and Durrant-Whyte, H., "Scalable Control of Decentralised Sensor Platforms," Vol. 2634 of *Lecture Notes in Comput. Science*, Springer, Palo Alto, CA, April 2003, pp. 1521–1526.

[4] Hoffmann, G. M., Waslander, S. L., and Tomlin, C. J., "Mutual Information Methods with Particle Filters for Mobile Sensor Network Control," *Proc. 45th IEEE Conf. Decision and Control*, San Diego, CA, December 2006, pp. 1019–1024.

[5] Chang, D. E., Shadden, S. C., Marsden, J. E., and Olfati-Saber, R., "Collision Avoidance for Multiple Agent Systems," *Proc. 42nd IEEE Conf. Decision and Control*, Maui, HI, December 2003, pp. 539–543.

[6] Nguyen, B. Q., Chuang, Y.-L., Tung, D., Hsieh, C., Jin, Z., Shi, L., Marthaler, D., Bertozzi, A., and Murray, R. M., "Virtual Attractive-Repulsive Potentials for Cooperative Control of Second Order Dynamic Vehicles on the Caltech MVWT," *AACC Amer. Control Conf.*, Portland, OR, June 2005, pp. 1084–1089.

[7] Shucker, B., Murphey, T., and Bennett, J. K., "Switching Rules for Decentralized Control with Simple Control Laws," *AACC Amer. Control Conf.*, New York, NY, July 2007, pp. 1485–1492.

[8] Inalhan, G., Stipanovic, D. M., and Tomlin, C. J., "Decentralized Optimization, with Application to Multiple Aircraft Coordination," *Proc. 41st IEEE Conf. Decision and Control*, Las Vegas, December 2002, pp. 1147–1155.

[9] Raghunathan, A. U., Gopal, V., Subramanian, D., Biegler, L. T., and Samad, T., "Dynamic Optimization Strategies for Three-Dimensional Conflict Resolution of Multiple Aircraft," *AIAA J. Guidance, Contr., and Dynamics*, Vol. 27, No. 4, 2004, pp. 586–594.

[10] Schouwenaars, T., How, J., and Feron, E., "Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees," *Proc. AIAA Guidance, Navigation, and Control Conf.*, Providence, RI, August 2004.

[11] Kuwata, Y. and How, J. P., "Robust cooperative Decentralized Trajectory Optimization using Receding Horizon MILP," *AACC Amer. Control Conf.*, New York, NY, July 2007, pp. 522–527.

[12] Raffard, R., Tomlin, C. J., and Boyd, S. P., "Distributed Optimization for Cooperative Agents: Application to Formation Flight," *Proc. 43rd IEEE Conf. Decision and Control*, Atlantis, Paradise Island, Bahamas, December 2004, pp. 2453–2459.

[13] Shim, D. H., Kim, H. J., and Sastry, S., "Decentralized Nonlinear Model Predictive Control of Multiple Flying Robots," *Proc. 42nd IEEE Conf. Decision and Control*, Maui, HI, December 2003, pp. 3621–3626.

[14] Tomlin, C., Pappas, G. J., and Sastry, S., "Conflict Resolution for Air Traffic Management: A Study in Multiagent Hybrid Systems," *IEEE Trans. Automat. Contr.*, Vol. 43, No. 4, 1998, pp. 509–521.

[15] Hoffmann, G. M., Waslander, S. L., and Tomlin, C. J., "Quadrotor Helicopter Trajectory Tracking Control," *Proc. AIAA Guidance, Navigation, and Control Conf.*, Honolulu, HI, August 2008.