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

Decentralized Reactive Collision Avoidance for Multivehicle Systems

Emmett Lalish and Kristi A. Morgansen

Abstract— **This paper addresses a novel approach to the n-vehicle collision avoidance problem. The vehicle model used is a planar unicycle, chosen for its wide applicability to ground, sea, and air vehicles. This paper generalizes previous work with constant-speed vehicles to models that include the ability to slow down, stop and reverse. An algorithm is developed that works in conjunction with any desired controller to guarantee all vehicles remain free of collisions while attempting to follow their desired control. This algorithm is reactive and decentralized, making it well suited for real time applications, and explicitly accounts for actuation limits. Results are demonstrated in simulation.**

I. INTRODUCTION

As multi-vehicle autonomous systems are studied and implemented, the issue of conflict resolution becomes an increasingly important point. From mobile robots performing a cooperative search to air traffic control for unmanned aerial vehicles (UAVs), collision avoidance is of utmost importance for safety.

The work here presents the Decentralized Reactive Collision Avoidance (DRCA) algorithm as a solution for conflict resolution. The purpose of this paper is to provide a more detailed description of the work first presented in [1], and generalized in [2], especially including a wider range of simulations.

Each vehicle is modeled as a unicycle in order to capture the essential dynamics of a wide range of vehicles. Additionally, arbitrary speed restrictions are allowed, making constant-speed vehicles a subset of this more general framework. The vehicles modeled can thus range from mobile robots and ships to submarines and aircraft. This framework also implicitly allows static obstacles to be avoided, since they can be modeled as zero-speed vehicles. Currently the DRCA algorithm is designed for planar applications, but extensions to three dimensions are in development. The third dimension adds a degree of freedom for re-routing which should improve performance for aircraft and other vehicles capable of three-dimensional movement.

A useful overview of papers on deconfliction can be found in [3]. The authors divide autonomous conflict resolution methods into three categories: prescribed, optimized, and force field. In prescribed maneuver approaches [4], [5], [6], all vehicles follow a set protocol, not unlike the rules of the road. While this approach can lead to straightforward proofs, it also tends to be less flexible with respect to changing conditions. Optimization schemes are also quite common ([7], [8], [9], [10]), but suffer in real time applications from non-deterministic computation time. Additionally, these approaches tend to be centralized, which often limit their applicability in real systems. Most force field approaches treat each vehicle as a charged particle that repels all the other vehicles, based primarily on position information (a zeroth order look ahead) [11], [12], [13], [14].

The DRCA algorithm fits most closely into the force field category, though it differs widely from most other algorithms which use force fields or potential functions. The force field defined in the work here differs in that it makes use of the collision cone concept, introduced in [15] and subsequently used often in the deconfliction literature [9], [16], [17]. This method involves a first order look ahead for detecting conflicts, which takes the restrictions of the unicycle model into account directly. An implicit assumption is that no antagonistic vehicles are present in the system; either all vehicles are trying to avoid conflicts, or at worst some are maintaining constant velocity.

The authors of [3] also make a distinction between pairwise and global conflict resolution maneuvers. While the collision cone is fundamentally a pairwise conflict detection scheme, the DRCA algorithm takes into account all of the other vehicles in order to compute the control, making it a global approach.

No homogeneity is required among the vehicles that make up the system. The DRCA algorithm allows each vehicle to have different size, speed, actuation limitations, and gains. The vehicles can even have completely different tasks they are performing.

As the name implies, the DRCA algorithm is also decentralized, in that no communication or agreement is required between the vehicles. Each vehicle does require the states of every other vehicle, but this information can come equally from sensing (e.g. radar) or from communication. If communication is chosen, the required n to n topology is relatively easy to implement through a broadcast. The effect of limited sensor or communication range on this system is a subject of current research and is beyond the scope of this paper.

One caveat of the DRCA algorithm is that it can only keep a system conflict-free. When vehicles start on a collision course, an initial deconfliction maneuver is required to bring the system to a safe state where the DRCA algorithm can take over. One such maneuver is discussed in this paper, and an initial separation bound is presented for which collision avoidance is guaranteed.

The remainder of the paper is organized as follows. Section II gives the problem statement and introduces definitions and notation used throughout the paper. Initial deconfliction maneuvers are discussed in Section III. The DRCA algorithm

This work supported by NSF grant CMS-0234861.

E. Lalish and K. A. Morgansen are with the Department of Aeronautics and Astronautics, University of Washington, Box 352400, Seattle, WA 98195-2400. {emmettl, morgansen}@aa.washington.edu.

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

is described in Section IV. Simulations and discussions of performance are given in Section V. Conclusions and future work are in Section VI.

II. Problem Statement

The work here presents a method for deconflicting n unicycle vehicles. Each vehicle has a nominal desired control input, $u_d(t)$, which comes from an arbitrary outer-loop controller. This controller is designed for the vehicle to perform a desired task, which could range from target tracking to way-point navigation, area searching, etc. The goal of the DRCA algorithm is to adjust the control input on each vehicle to guarantee collision avoidance while simultaneously staying close to the desired control input (keeping in mind that this desired control can change with time).

The dynamics of the i^{th} vehicle are

$$\frac{d}{dt}\begin{bmatrix} x_i \\ y_i \\ s_i \\ \psi_i \end{bmatrix} = \begin{bmatrix} s_i\cos(\psi_i) \\ s_i\sin(\psi_i) \\ u_{t_i} \\ u_{n_i} \end{bmatrix}, \tag{1}$$

where x_i and y_i are inertial position coordinates, ψ_i is the heading relative to an inertial frame, and s_i is the speed. The inputs are forward acceleration and heading rate, $u_i = [u_{t_i}, u_{n_i}]^T$.

The position vector of vehicle i is denoted $r_i \equiv [x_i, y_i]^T$, while the tangent (heading) vector, t, and normal to the tangent vector, n, of vehicle i are:

$$t_i \equiv \begin{bmatrix} \cos\psi_i \\ \sin\psi_i \end{bmatrix} \quad \text{and} \quad n_i \equiv \begin{bmatrix} -\sin\psi_i \\ \cos\psi_i \end{bmatrix}.$$

The velocity vector is then: $v_i = s_i t_i$. The relative position vector from vehicle i to vehicle j is denoted $\tilde{r}_{ij} \equiv r_j - r_i$, while the relative velocity vector is defined in the opposite sense: $\tilde{v}_{ij} \equiv v_i - v_j$. Note that these definitions imply that $\dot{\tilde{r}}_{ij} = -\tilde{v}_{ij}$, and

$$\dot{\tilde{v}}_{ij} = u_{t_i}t_i + u_{n_i}s_i n_i - u_{t_j}t_j - u_{n_j}s_j n_j. \tag{2}$$

The speed of the vehicle, s_i, is restricted to lie in a closed interval $\mathcal{S}_i$:

$$\mathcal{S}_i = \{s_i \in \mathbb{R} | s_{i,min} \le s_i \le s_{i,max}\}. \tag{3}$$

The minimum speed can be zero for a vehicle that can stop but cannot reverse, negative for a vehicle that can reverse, or positive for a vehicle with a positive minimum speed (e.g. aircraft). One can also have $s_{min} = s_{max}$ for constant-speed applications.

The inputs are restricted to a constrained domain by the specific physical limitations of the vehicle being studied:

$$u_i \in \mathcal{C}_i, \tag{4}$$

where

$$\mathcal{C}_i = \{u_i \in \mathbb{R}^2 \big| u_{t_{i,min}} \le u_{t_i} \le u_{t_{i,max}}, \quad u_{n_{i,min}} \le u_{n_i} \le u_{n_{i,max}}\}.$$

The constraint set, $\mathcal{C}_i$, can vary with time, but must always include the origin (i.e. the vehicle must always be capable of maintaining its current velocity). In fact, speed restrictions can be implicitly modeled by choosing $u_{t,max} \to 0$ as $s \to s_{max}$, which is an accurate model of most real vehicles.

Additionally, many vehicles are not modeled well by a rectangular constraint set (i.e. the maximum acceleration and turning are coupled). To handle this case, one can simply use a constraint set that encloses the true input constraint set, then apply saturation to the input generated by the DRCA algorithm. The guarantees will hold as long as the signs of the inputs are correct (hence the requirement that the input constraint set contains the origin).

The vehicles considered here are modeled as nonholonomic point masses, however physical vehicles have finite size. Therefore to account for physical constraints in the theoretical model, the condition for conflict is not to attain the same position in space at the same time, but rather to come within a minimum allowed distance of each other at some point in time. This minimum distance could be, for example, the five nautical mile separation between aircraft required by the FAA or the sum of the radii of two mobile robots.

Definition 1 (Collision): A collision occurs between vehicles i and j when

$$\|\tilde{r}_{ij}\| < d_{sep,ij}, \tag{5}$$

where $d_{sep,ij}$ is the minimum allowed separation distance between the vehicles' geometric centers.

For two vehicles not actively in a collision, the next question is whether they will collide if they remain on their present headings. This situation will be called a conflict.

Definition 2 (Conflict): A conflict occurs between vehicles i and j if they are not currently in a collision, but with zero control input (i.e. constant velocity), at some future point in time they will enter a collision:

$$d_{min,ij} \equiv \min_{t>0} \|\tilde{r}_{ij}\| < d_{sep,ij}. \tag{6}$$

The following lemma provides a useful way to check for conflicts. To simplify the notation in the rest of this paper, the ij subscripts will generally be suppressed (for example, $\tilde{r}_{ij}$ will be written as $\tilde{r}$).

Lemma 1: Let $\beta = \angle\tilde{v} - \angle\tilde{r}_0$, $\alpha = \arcsin\left(\frac{d_{sep}}{\|\tilde{r}_0\|}\right)$ and $\tilde{r}_0$ be the relative position vector at the time conflict is being checked. A necessary and sufficient condition for no conflict to occur is

$$|\beta| \ge \alpha. \tag{7}$$

The angle α represents the half-width of the collision cone ([15], [9], [17]), which is depicted in Fig. 1.

This lemma is proven in [1]. The angle α denotes the half-width of the collision cone, similar to [9], [17], [15], and described geometrically in Fig. 1.

III. Initial Deconfliction Maneuvers

Section IV contains a proof that the DRCA algorithm can keep a system conflict-free indefinitely once a conflict-free state is achieved. However, generally the point of collision avoidance is to resolve conflicts that are already present.

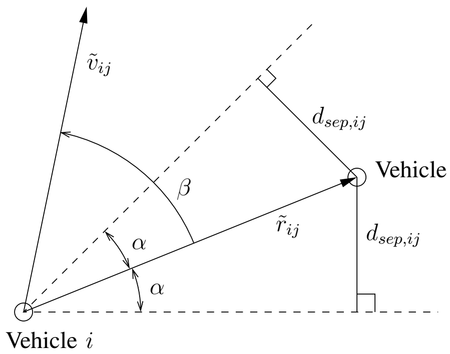

Fig. 1. Geometric definition of the collision cone. The area between the two dotted lines is the collision cone; a conflict occurs when the relative velocity vector, $\tilde{v}_{ij}$, lies within this area.

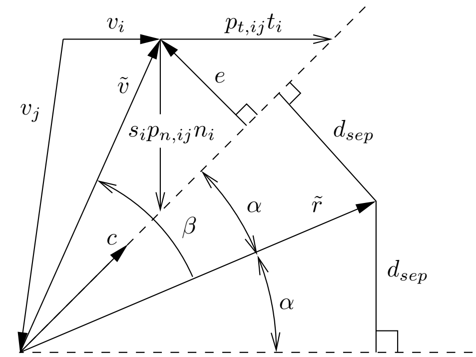

Fig. 2. Geometry of the e and c vectors and the conflict measures p_t and p_n ($c^T\tilde{v} > 0$ in this example). The dotted lines represent the collision cone.

Hence some initial maneuver is required by the vehicles to bring themselves from a conflicted state to a conflict-free state so that the DRCA algorithm can keep them that way.

The purpose of the initial deconfliction maneuver is to bring vehicles to a conflict-free state as quickly as possible with a guarantee that no collisions will occur during this maneuver, given certain bounds on the initial conditions. The initial condition bounds are necessary because a wide class of initial conditions exist for which collision avoidance is impossible (usually due to a lack of sufficient control authority).

Many possible maneuvers can be used for initial deconfliction, but this section will detail one simple option. A first consideration for how to maneuver out of conflict is which of the inputs should dominate. Changing speed is a problem for two reasons. First, most vehicles have less capability for forward acceleration than for lateral acceleration. Second, all vehicles have speed limits, which means that at some point, control authority (in one direction) vanishes entirely. Based on these considerations the deconfliction maneuver presented here will amount to a turning-only command.

One maneuver to achieve deconfliction, as will be proven below, is to have all vehicles turn the same way at maximum rate until a conflict-free state is reached for the whole system. If the direction is pre-programmed (e.g. positive or left), this maneuver mimics a rules-of-the-road approach, where vehicles always pass on the same side. A similar approach was used in [5], because for exact or nearly-exact conflicts, this maneuver results in a roundabout passing behavior, which tends to reduce the deviation of the vehicles from their paths.

However, more important than performance is a guarantee of safety. Collision avoidance for the "all hard left" maneuver can be proven because even when the system cannot reach a conflict-free state, the maneuver simply becomes a loiter pattern.

Theorem 1: If the initial separation of each pair of vehicles satisfies

$$\|\tilde{r}_{ij}\| \geq 2\frac{s_i}{u_{n_{i,max}}} + 2\frac{s_j}{u_{n_{j,max}}} + d_{sep_{ij}}, \qquad (8)$$

Then a system of n vehicles will remain collision free for all time if each vehicle constantly turns at its maximum rate while maintaining speed.

Proof: The trajectory each vehicle follows using constant turning is a circle of radius $\frac{s_i}{u_{n_{i,max}}}$. As long as a pair of vehicles is separated by at least the sum of the diameters of their loiter patterns and the minimum separation distance, then they can never collide. ■

For initial conditions where the conflicts are inexact (as defined in [5]), this rules-of-the-road approach can occasionally lead to poor performance (depending on the cost function chosen). Nowhere does the proof stipulate that each vehicle must turn the same direction, only that each must pick a constant turning direction. Therefore some kind of heuristic or even an optimization algorithm may be capable of giving better performance than the simple "all hard left" command. These alternate maneuvers are a topic of current research. Additionally, application-specific adjustments could include changing speed to attain the minimum vehicle turning radius.

IV. DRCA Algorithm

Once the initial deconfliction maneuver has been performed and the system is in a conflict-free state, the DRCA algorithm can be used and allows each vehicle to use its desired control input unless that input would cause the vehicle to come into conflict with another vehicle.

In order to smoothly transition from the desired control to the avoidance control, each vehicle needs a way to measure how close its velocity vector is to causing a conflict. The first step is to construct a unit vector, c, defining the near side of the collision cone:

$$c = R(\text{sgn}(\beta)\alpha)\frac{\tilde{r}}{\|\tilde{r}\|},$$

where $\text{sgn}(\gamma)$ is the sign function, taking on the value 1 for $\gamma \geq 0$ and -1 for $\gamma < 0$, and $R(\gamma)$ is the planar rotation matrix:

$$R(\gamma) \equiv \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix}.$$

Next, construct a normal vector, e, from the collision cone to the relative velocity vector, $\tilde{v}$ (see Fig. 2). If $c^T\tilde{v} > 0$, e

can be found from the following two geometric relations:

$$e + kc = \tilde{v}$$
$$c^T e = 0,$$

where k is an appropriate scalar. These equations can be rewritten as

$$\begin{bmatrix} I & c \\ c^T & 0 \end{bmatrix} \begin{bmatrix} e \\ k \end{bmatrix} = \begin{bmatrix} \tilde{v} \\ 0 \end{bmatrix}. \tag{9}$$

If $c^T \tilde{v} \le 0$ (the vehicles are headed away from each other), then no normal exists and the nearest point on the collision cone is the tip, so $e = \tilde{v}$. Combining this result with the solution of (9) gives

$$e = \begin{cases} \tilde{v}, & c^T \tilde{v} \le 0 \\ R\left(\frac{\pi}{2}\right) c c^T R^T \left(\frac{\pi}{2}\right) \tilde{v}, & c^T \tilde{v} > 0, \end{cases}$$

where $R\left(\frac{\pi}{2}\right) c$ is a vector orthogonal to c.

Because the control inputs u_t and u_n produce changes in velocity along the orthogonal vectors t and n respectively (2), they are decoupled and can be analyzed separately. For u_t, a signed distance to the collision cone in the direction of t_i is needed in order to determine how much change in speed would cause a conflict. This signed distance is defined as follows (valid only when no conflict is present):

$$p_{t,ij} = \frac{\|e_{ij}\|^2}{e_{ij}^T t_i}.$$

The definition of the signed distance in the normal direction is similar (see Fig. 2), but must be augmented by the vehicle speed, s_i, to account for the relationship between u_n (heading rate) and lateral acceleration:

$$p_{n,ij} = \frac{\|e_{ij}\|^2}{s_i e_{ij}^T n_i}.$$

Note that when $c^T \tilde{v} < 0$, these signed distances do not measure strictly to the collision cone, but rather to a half-space that encloses it. Therefore these signed distances are conservative.

Define $\epsilon_t, \epsilon_n > 0$ as thresholds such that when $\|p_t\| > \epsilon_t$, the conflict is far enough away that it can be ignored (and likewise for p_n). The n-vehicle DRCA algorithm running on vehicle i computes p_t and p_n to each of the other vehicles and then finds the closest conflict in each direction, i.e.

$$\begin{aligned} p_{t_i}^+ &= \min_j \{p_{t,ij} > 0, \epsilon_{t_i}\} \\ p_{t_i}^- &= -\max_j \{p_{t,ij} < 0, -\epsilon_{t_i}\}, \end{aligned} \tag{10}$$

and likewise for p_n. Note that by definition $0 < p^\pm \le \epsilon$. To simplify notation, in any case where a relation holds for both p_t and p_n, the subscript will be suppressed.

Each control input is constructed using a function, F, such that $u_t = F(p_t^+, p_t^-)$ and $u_n = F(p_n^+, p_n^-)$. The control function chosen for this implementation of the DRCA algorithm is piecewise-linear, defined by the following ordered triples of the form (p^+, p^-, u):

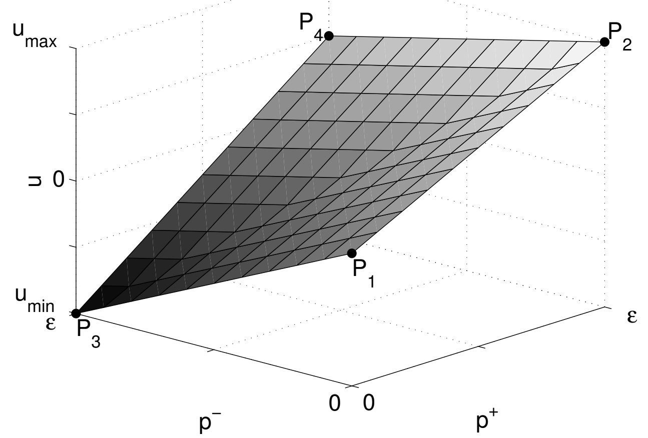

Fig. 3. Example of the control function, F. Note that P_4 moves up and down with changing u_d.

$$\begin{aligned} P_1 &= (0, 0, 0) \qquad P_2 = (\epsilon, 0, u_{max}) \\ P_3 &= (0, \epsilon, u_{min}) \qquad P_4 = (\epsilon, \epsilon, u_d). \end{aligned} \tag{11}$$

An example of this control function is shown in Fig. 3. Because F is a function of the desired control, u_d must be saturated such that

$$u_{min} \le u_d \le u_{max}. \tag{12}$$

A more intuitive way to choose values for ϵ_t and ϵ_n is to relate them to gain-like parameters,

$$k_t = \frac{u_{t,max} - u_{t,min}}{\epsilon_t}$$

(and similarly for k_n). Note that both k_t and k_n have units of inverse seconds. Also, one can see that the magnitude of the gradient of the control function will always be less than or equal to k, regardless of the desired control.

Theorem 2: The DRCA algorithm described above, when implemented on n vehicles with dynamics (1) and input constrained by (4), will keep the system collision free for all time if the system starts conflict-free.

Proof: To measure the distance to a collision, define m as a signed version of $\|e\|$ (in terms of $\tilde{v}$ from the geometry in Fig. 2):

$$m = \begin{cases} \|\tilde{v}\|, & c^T \tilde{v} \le 0 \\ \|\tilde{v}\| \sin(|\beta| - \alpha), & c^T \tilde{v} > 0. \end{cases} \tag{13}$$

Note that m is negative during conflict and positive during no conflict.

To ensure that a conflicted state is never reached, it is sufficient to show that

$$\lim_{m \to 0^+} \dot{m} \ge 0,$$

for every pair of vehicles. This condition implies that as the boundary of a collision cone approaches, it will either stop approaching or recede before a conflict is formed.

For $c^T\tilde{v} \leq 0$ (using the ij notation again briefly for clarity),

$$\dot{m}_{ij} = \frac{e_{ij}^T \dot{\tilde{v}}_{ij}}{m_{ij}}. \tag{14}$$

Substituting (2) yields

$$\begin{aligned}\frac{e_{ij}^T \dot{\tilde{v}}_{ij}}{m_{ij}} =& \frac{1}{m_{ij}} \big(u_{t_i} e_{ij}^T t_i + u_{n_i} s_i e_{ij}^T n_i \\ &- u_{t_j} e_{ij}^T t_j - u_{n_j} s_j e_{ij}^T n_j\big).\end{aligned}$$

Because of the symmetry of the problem $e_{ji} = -e_{ij}$, so

$$\begin{aligned}\frac{e_{ij}^T \dot{\tilde{v}}_{ij}}{m_{ij}} =& \frac{1}{m_{ij}} \big(u_{t_i} e_{ij}^T t_i + u_{n_i} s_i e_{ij}^T n_i \\ &+ u_{t_j} e_{ji}^T t_j + u_{n_j} s_j e_{ji}^T n_j\big) \\ =& \frac{\|e_{ij}\|^2}{m_{ij}} \left(\frac{u_{t_i}}{p_{t,ij}} + \frac{u_{n_i}}{p_{n,ij}} + \frac{u_{t_j}}{p_{t,ji}} + \frac{u_{n_j}}{p_{n,ji}}\right) \\ =& m_{ij} \left(\frac{u_{t_i}}{p_{t,ij}} + \frac{u_{n_i}}{p_{n,ij}} + \frac{u_{t_j}}{p_{t,ji}} + \frac{u_{n_j}}{p_{n,ji}}\right). \end{aligned} \tag{15}$$

As long as the controller ensures that u_{t_i} has the same sign as $p_{t,ij}$ and u_{n_i} has the same sign as its $p_{n,ij}$, then $\dot{m} \geq 0$ for that pair of vehicles. Note that each vehicle need only calculate its control from its own point of view, and this rule will automatically cause the vehicles to cooperate in avoiding conflicts.

For $c^T\tilde{v} > 0$, the derivative of (13) becomes

$$\dot{m} = \sin(|\beta| - \alpha) \frac{d\|\tilde{v}\|}{dt} + \|\tilde{v}\| \cos(|\beta| - \alpha) \frac{d}{dt}(|\beta| - \alpha). \tag{16}$$

From the geometry,

$$\begin{aligned}\frac{d|\beta|}{dt} &= \mathrm{sgn}(\beta) \left(\frac{d\angle\tilde{v}}{dt} - \frac{d\angle\tilde{r}}{dt}\right) \\ &= \mathrm{sgn}(\beta) \frac{d\angle\tilde{v}}{dt} + \frac{\|\tilde{v}\|}{\|\tilde{r}\|} |\sin\beta|\end{aligned}$$

The derivative of α is somewhat less straight-forward:

$$\begin{aligned}\frac{d\alpha}{dt} &= \frac{d}{dt}\left(\arcsin\left(\frac{d_{sep}}{\|\tilde{r}\|}\right)\right) \\ &= \frac{d}{dt}\left(\frac{d_{sep}}{\|\tilde{r}\|}\right)\left(1 - \left(\frac{d_{sep}}{\|\tilde{r}\|}\right)^2\right)^{-1/2} \\ &= \frac{d_{sep}\|\tilde{v}\|\cos\beta}{\|\tilde{r}\|^2}\left(\frac{\|\tilde{r}\|}{\sqrt{\|\tilde{r}\|^2 - d_{sep}^2}}\right) \\ &= \frac{\|\tilde{v}\|}{\|\tilde{r}\|}\cos\beta\tan\alpha.\end{aligned}$$

Combining the above two terms gives

$$\frac{d}{dt}(|\beta| - \alpha) = \frac{\|\tilde{v}\|}{\|\tilde{r}\|}(|\sin\beta| - \cos\beta\tan\alpha) + \mathrm{sgn}(\beta)\frac{d\angle\tilde{v}}{dt},$$

which can be substituted into (16) to get

$$\begin{aligned}\dot{m} =& \sin(|\beta| - \alpha)\frac{d\|\tilde{v}\|}{dt} + \cos(|\beta| - \alpha)\,\mathrm{sgn}(\beta)\,\|\tilde{v}\|\,\frac{d\angle\tilde{v}}{dt} \\ &+ \cos(|\beta| - \alpha)\frac{\|\tilde{v}\|^2}{\|\tilde{r}\|}(|\sin\beta| - \cos\beta\tan\alpha). \end{aligned} \tag{17}$$

To simplify the above, note that in the case of $c^T\tilde{v} > 0$, the geometry of the vectors (Fig. 2) gives

$$\begin{aligned}e^T\dot{\tilde{v}} = m\Bigg(&\sin(|\beta| - \alpha)\frac{d\|\tilde{v}\|}{dt} \\ &+ \cos(|\beta| - \alpha)\,\mathrm{sgn}(\beta)\,\|\tilde{v}\|\,\frac{d\angle\tilde{v}}{dt}\Bigg).\end{aligned}$$

Therefore (17) reduces to

$$\dot{m} = \frac{e^T\dot{\tilde{v}}}{m} + \cos(|\beta| - \alpha)\frac{\|\tilde{v}\|^2}{\|\tilde{r}\|}(|\sin\beta| - \cos\beta\tan\alpha).$$

Assuming the system is conflict-free, Lemma 1 dictates $|\beta| \geq \alpha$. Therefore $|\tan\beta| \geq \tan\alpha$, and so

$$|\sin\beta| - \cos\beta\tan\alpha \geq 0.$$

Also, $c^T\tilde{v} > 0$ implies that $\cos(|\beta| - \alpha) > 0$. Therefore, combining this result with (14) implies that

$$\dot{m} \geq \frac{e^T\dot{\tilde{v}}}{m}$$

for any value of $c^T\tilde{v}$. Recalling (15),

$$\dot{m}_{ij} \geq m_{ij}\left(\frac{u_{t_i}}{p_{t,ij}} + \frac{u_{n_i}}{p_{n,ij}} + \frac{u_{t_j}}{p_{t,ji}} + \frac{u_{n_j}}{p_{n,ji}}\right).$$

Combining this result with the definitions (10), any continuous control function that satisfies

$$\begin{aligned}\lim_{p_t^+ \to 0} u_t \geq 0, \quad \lim_{p_t^- \to 0} u_t \leq 0, \\ \lim_{p_n^+ \to 0} u_n \geq 0, \quad \lim_{p_n^- \to 0} u_n \leq 0,\end{aligned} \tag{18}$$

also ensures that

$$\lim_{m \to 0^+} \dot{m} \geq 0,$$

guaranteeing the system cannot enter a conflicted state.

The control function used in this implementation (11) satisfies (18), so the DRCA algorithm will cause the n-vehicle system to remain conflict-free for all time, assuming it started that way. ■

Note this result holds for arbitrary (even time varying) u_d, u_{min} and u_{max}, so long as they satisfy (12) and $\mathcal{C}$ contains the origin at every instant. In fact, further saturation may be applied to u (for instance to accommodate a non-rectangular constraint set) without affecting the safety guarantee, so long as the sign of each input is preserved.

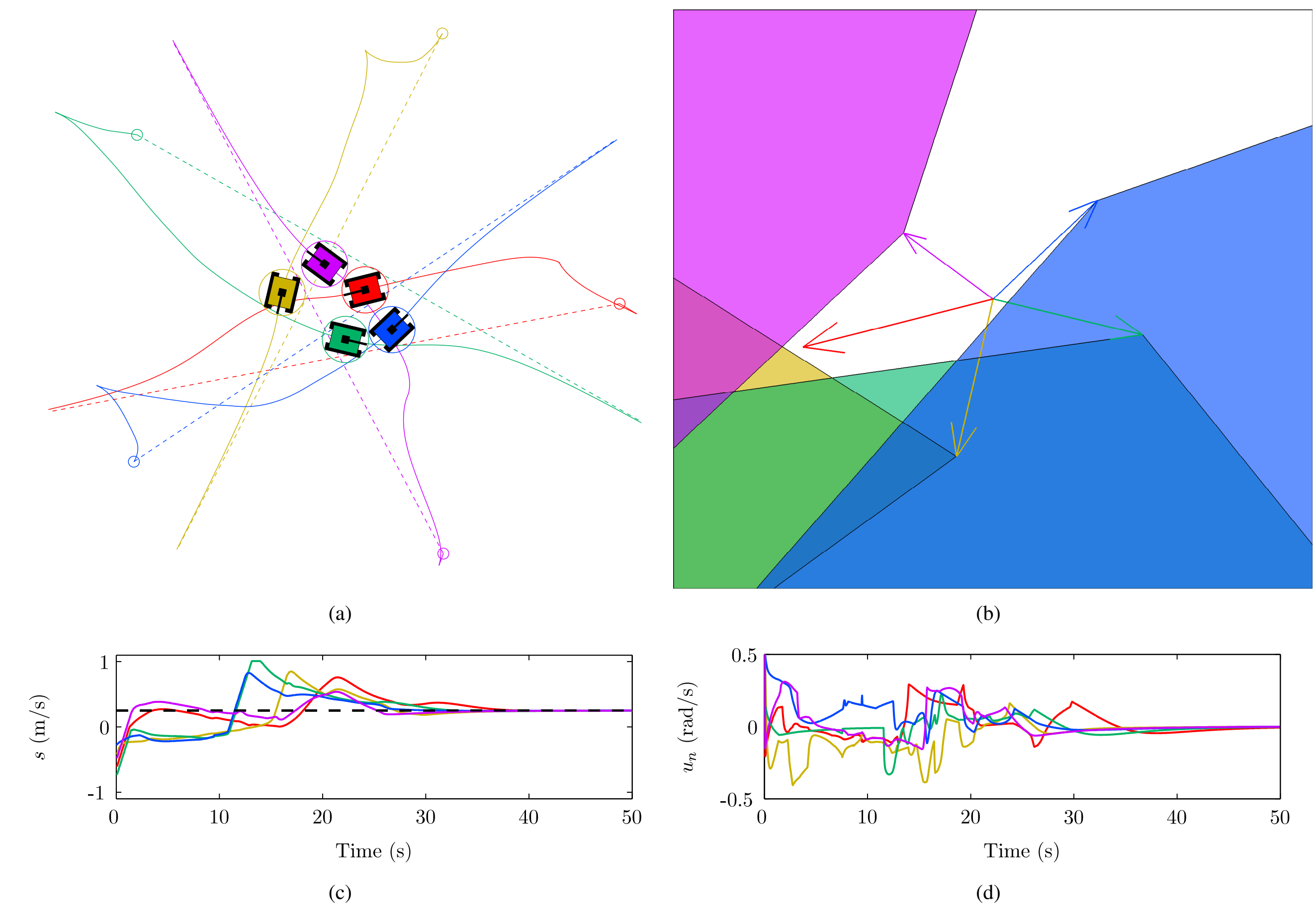

Fig. 4. Five vehicle simulation: (a) shows the vehicle trajectories. Target paths are shown as dotted lines. Vehicles are shown at 25 seconds, while their initial positions are denoted by small circles. All vehicles start with negative speeds (c). (b) shows a snapshot of the collision cones from the perspective of the red vehicle at 25 seconds. The arrows denote each vehicle's velocity. This illustration effectively overlaps several diagrams like Fig. 2. The yellow vehicle is closest to conflict with the purple and yellow vehicles. (c) and (d) show the corresponding speeds and turning rates of the vehicles, respectively.

V. SIMULATIONS

A. Variable Speed

The first simulation (Fig. 4) is of a homogeneous group of five vehicles that are capable of moving both forward and backward. Each vehicle has a target that moves at constant velocity, whose position and orientation the vehicle is attempting to attain (using a standard target following controller). The vehicles are all initialized with random headings and negative initial speeds (see Fig. 4c), so the corners in the paths denote points where the vehicle speed crosses zero. The initial positions of the vehicles form a circle of radius six meters, with their targets heading toward the center (with some randomization in heading and position). Circles of diameter $d_{sep} = 1$ m are denoted around each vehicle in Fig. 4a. A collision occurs if the circles overlap. In this example $k_n = 3$ s^{-1} and $k_t = 10$ s^{-1}, which causes the vehicles to follow their desired acceleration more than their desired turning. $s_{max} = -s_{min} = 1$ m/s and $u_{t,max} = -u_{t,min} = 0.5$ m/s^2 and $u_{n,max} = -u_{n,min} = 0.5$ rad/s.

In this example, the vehicles begin conflict-free, so no deconfliction maneuver is necessary. However, the targets move in such a way that following the desired controls would cause the vehicles to collide. The DRCA algorithm successfully keeps the system conflict-free while eventually allowing the vehicles to attain their targets. The turning rate control for each vehicle is shown in Fig. 4d. To demonstrate how the controller operates, Fig. 4b shows the collision cones from the red vehicle's perspective at the same instant that the vehicles are plotted in Fig. 4a. Interesting to note is that while the blue and purple vehicles are the nearest in distance to the red vehicle, the purple and yellow vehicles are closest to conflict.

B. Heterogeneity

The second simulation is geared more toward aircraft, as the group of five vehicles in Fig. 5 are restricted to a constant speed of 1 m/s. In addition to the five vehicles is one static obstacle (which can be thought of as a zero speed vehicle). In this simulation, d_{sep} is the sum of the radii of the two vehicles involved, and the shaded regions represent the radius of each vehicle. The vehicles' initial conditions form them into a circle of radius eight meters and point them roughly toward the obstacle, so the system starts in conflict.

The vehicles perform the "all turn left" initial deconfliction maneuver (as seen in Fig. 6), which brings the system to a

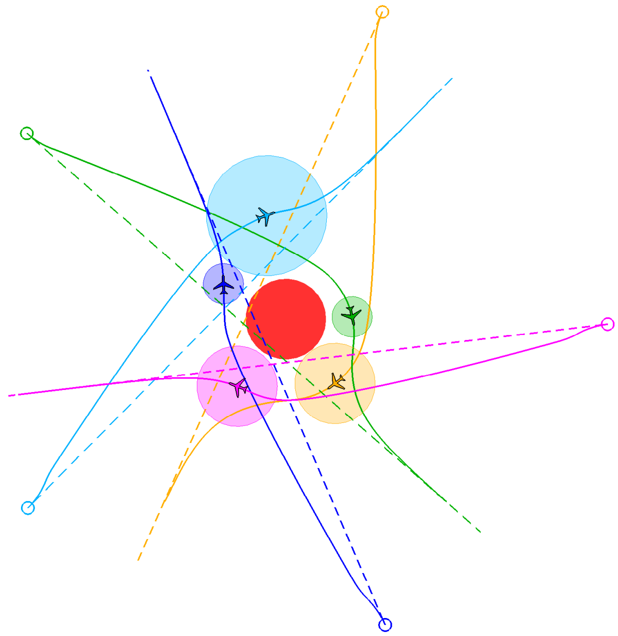

Fig. 5. Five constant-speed vehicle simulation with one obstacle (red). Desired paths are shown as dotted lines. Vehicles are shown at 9.5 seconds, while their initial positions are denoted by small circles. The shaded circles represent vehicle size, such that a collision occurs when the shaded regions overlap.

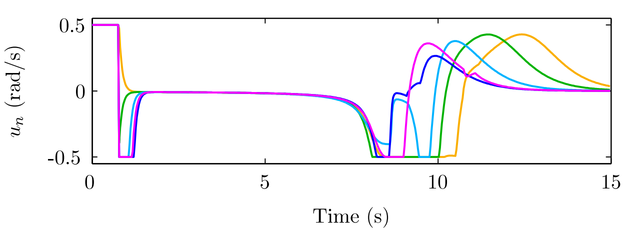

Fig. 6. Turning rate input of the vehicles in Fig. 5. $u_{n,max} = -u_{n,min} = 0.5$ rad/s and $k_n = 5$ s^{-1}.

conflict-free state in less than one second. This deconfliction maneuver naturally results in a roundabout type of behavior. Each vehicle uses a path following algorithm for its desired control, where the path is defined as a straight line emanating from the vehicle's initial conditions. All of the vehicles return to their desired paths once they have passed the obstacle and each other.

VI. Conclusion

The work presented in this paper has developed a decentralized control algorithm for deconflicting n unicycle-type vehicles. The DRCA algorithm is reactive and so can easily be implemented real time on a wide variety of vehicles, including aircraft, ships, submarines and cars. Collision avoidance is guaranteed for a general n-vehicle system once a conflict-free state is reached, even in the case of arbitrarily small control authority. A lower bound (8) was found for the initial separation of the n vehicles such that collision avoidance is assured even when starting in conflict. Finally, the DRCA algorithm allows the vehicles to follow changing desired controls so long as safety is not sacrificed.

Many extensions are possible for this work. First, for applications to aircraft and submarines, broadening this concept to three dimensions will add a degree of freedom and hence increase the performance of the system. Second, the DRCA algorithm shows promise for evading adversarial pursuers, and may lead to interesting results regarding pursuit/evasion games. Third, finding more general initial deconfliction maneuvers will make this algorithm more suitable to general conflict scenarios. Finally, integrating a finite detection horizon into this algorithm will be important to maintain the collision avoidance guarantees while operating with physical sensor models that limit the available information.

References

[1] E. Lalish, K. A. Morgansen, and T. Tsukamaki, "Decentralized reactive collision avoidance for multiple unicycle-type vehicles," in *Proc. IEEE American Control Conference*, 2008.

[2] E. Lalish and K. A. Morgansen, "Decentralized reactive collision avoidance for multivehicle systems," in *Proc. IEEE Conference on Decision and Control*, 2008, to appear.

[3] J. Kuchar and L. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Transactions on Intellingent Transportation Systems*, vol. 1, no. 4, pp. 179–189, 2000.

[4] K. Bilimoria, B. Sridhar, and G. Chatterji, "Effects of conflict detection methods for air traffic management," in *Proc. AIAA Guidance, Navigation, and Control Conference*, 1996.

[5] I. Hwang and C. Tomlin, "Protocol-based conflict resolution for air traffic control," Stanford University, Tech. Rep. SUDAAR-762, 2002.

[6] L. Pallottino, V. G. Scordio, A. Bicchi, and E. Frazzoli, "Decentralized cooperative policy for conflict resolution in multivehicle systems," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1170–1183, 2007.

[7] J. Andrews, "A relative motion analysis of horizontal collision avoidance," *SAFE Journal*, vol. 8, no. 2, pp. 7,8, 1978.

[8] C. Tomlin, G. Pappas, and S. Sastry, "Conflict resolution for air traffic management: a study in multiagent hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 509–521, April 1998.

[9] E. Frazzoli, Z. Mao, J. Oh, and E. Feron, "Resolution of conflicts involving many aircraft via semidefinite programming," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 24, no. 1, pp. 79–86, 2001.

[10] J. C. Hill, J. K. Archibald, W. Stirling, and R. L. Frost, "A multi-agent system architecture for distributed air traffic control," in *Proc. AIAA Guidance, Navigation and Control Conference*, 2005.

[11] M. Eby and W. Kelly, "Free flight separation assurance using distributed algorithms," in *Proc. IEEE Aerospace Conference*, 1999, pp. 429–441.

[12] E. Lalish, K. A. Morgansen, and T. Tsukamaki, "Formation tracking control using virtual structures and deconfliction," in *Proc. IEEE Conference on Decision and Control*, 2006.

[13] S. Mastellone, D. M. Stipanovic, C. R. Graunke, K. A. Intlekofer, and M. W. Spong, "Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments," *International Journal of Robotics Research*, vol. 27, no. 1, pp. 107–126, 2008.

[14] G. P. Roussos, D. V. Dimarogonas, and K. J. Kyriakopoulos, "3D navigation and collision avoidance for a non-holonomic vehicle," in *Proc. IEEE American Control Conference*, 2008.

[15] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: A collision cone approach," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 28, no. 5, pp. 562–574, September 1998.

[16] Z. Shiller, F. Large, and S. Sekhavat, "Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories," in *Proc. IEEE International Conference on Robotics and Automation*, 2001.

[17] C. Carbone, U. Ciniglio, F. Corraro, and L. Luongo, "A novel 3D geometric algorithm for aircraft autonomous collision avoidance," in *Proc. IEEE Conference on Decision and Control*, 2006.