

Decentralized Reactive Collision Avoidance for Multiple Unicycle-Type Vehicles

Emmett Lalish, Kristi A. Morgansen and Takashi Tsukamaki

Abstract—This paper addresses a novel approach to the n -vehicle collision avoidance problem. The vehicle model used is a planar constant-speed unicycle, chosen for its wide applicability to ground, sea, and air vehicles. An algorithm is developed which guarantees all vehicles remain free of collisions while attempting to attain their trajectory goals (given certain restrictions on their initial conditions). This controller 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.

In order to capture the essential dynamics of a wide range of vehicles, the unicycle model was chosen for this work. That choice allows this collision avoidance algorithm to function on surface vehicles, underwater vehicles, ships and aircraft. Aircraft are the primary goal of this research, and so a constant speed model was chosen as many aircraft (especially UAVs) have such a restricted flight envelope that their range of acceptable speeds is quite small. Additionally, because turning does not change the vehicle's kinetic energy, this type of control does not tend to require as much effort as changing speed, even for surface vehicles.

The work presented here is planar for applicability to the widest range of systems. Even aircraft are often restricted to a fixed altitude for air traffic control. In addition altitude changes, like speed changes, require a change in energy which places more load on the actuation system. Finally, adding a third dimension actually makes the deconfliction problem easier because it adds another degree of freedom for re-routing. Therefore this work can be fairly easily extended to three dimensions.

The algorithm presented here makes use of the collision cone concept, which is common in the deconfliction literature [1], [2], [3]. This method involves a first order look ahead for detecting conflicts. Therefore, 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 a constant heading.

A useful overview of the papers on aircraft deconfliction can be found in [4]. The authors divide autonomous conflict resolution methods into three categories: prescribed, optimized, and force field.

Prescribed maneuvers include approaches like [5], [6], [7], where all vehicles follow a set protocol, not unlike the rules of the road. While this approach can lead to easy proofs, it also tends to be less flexible with respect to changing conditions.

Optimization schemes are also quite common [3], [8], [9], 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.

The approach presented here fits most closely into the force field category, though it differs widely from most other algorithms which use force fields or potential functions. 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) [10], [11], [12]. The force field defined in the work here differs in that it relies primarily on velocity information, and is defined on a different metric. Therefore this approach can take the restrictions of a unicycle model into account directly.

The authors of [4] also make a distinction between pairwise and global conflict resolution maneuvers. While the collision cone is fundamentally a pairwise conflict detection scheme, the algorithm presented here takes into account all of the other vehicles in order to compute the control, making it a global approach.

The algorithm presented here 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) as from communication. If communication is the chosen route, 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.

This paper is organized as follows. Section II gives the problem statement. Definitions related to the collision cone and conflict detection are given in Section III. The collision avoidance algorithm is described in Section IV. Section V demonstrates the algorithm in simulation. Conclusions and future work are in Section VI.

This work supported in part by NSF grant CMS-0234861 and in part by a grant from the Boeing Company.

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

T. Tsukamaki is with Boeing Phantom Works, Seattle, WA. takashi.tsukamaki@boeing.com.

II. PROBLEM STATEMENT

This paper presents a method for deconflicting a homogeneous group of n constant-speed unicycle vehicles. The dynamics of the i^{th} vehicle are:

$$\frac{d}{dt} \begin{bmatrix} x_i \\ y_i \\ \psi_i \end{bmatrix} = \begin{bmatrix} v \cos(\psi_i) \\ v \sin(\psi_i) \\ u_i \end{bmatrix}, \quad (1)$$

where v is the constant speed (identical for all vehicles), and the only input is u , the heading rate. The input is restricted as follows:

$$-u_{max} \leq u_i \leq u_{max}. \quad (2)$$

Each vehicle also has a desired control input, $u_d(t)$, which comes from an arbitrary outer-loop controller. This controller causes the vehicle to perform a desired task, which could be anything, e.g. target tracking, waypoint navigation, area searching, etc. The goal of the algorithm developed in this paper is to adjust the control input on each vehicle to guarantee collision avoidance while simultaneously staying as close to the desired control input as possible (keeping in mind that this desired control will change with time, as it also comes from a feedback control law).

Additional definitions used in this paper are as follows. The position vector of vehicle i is

$$\vec{r}_i \equiv \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \quad (3)$$

and similarly the velocity vector is

$$\vec{v}_i \equiv \frac{d\vec{r}_i}{dt} = \begin{bmatrix} v \cos(\psi_i) \\ v \sin(\psi_i) \end{bmatrix}. \quad (4)$$

The relative position vector from vehicle i to vehicle j is defined as

$$\vec{r}_{ij} \equiv \vec{r}_j - \vec{r}_i, \quad (5)$$

while the relative velocity vector is defined in the opposite sense:

$$\vec{v}_{ij} \equiv \vec{v}_i - \vec{v}_j. \quad (6)$$

Note that these definitions imply the following:

$$\frac{d\vec{r}_{ij}}{dt} = -\vec{v}_{ij}. \quad (7)$$

For constant \vec{v}_{ij} (i.e. $u_i = u_j = 0$), $\vec{r}_{ij}(t)$ evolves as

$$\vec{r}_{ij}(t) = - \int_0^t \vec{v}_{ij} dt = \vec{r}_{ij}(0) - \vec{v}_{ij}t, \quad (8)$$

where $t = 0$ means the current time. To simplify the notation in the rest of this paper, $t = 0$ will be assumed and the ij subscripts will be dropped (for example, $\vec{r}_{ij}(0)$ will be written as \vec{r}).

III. CONFLICTS & COLLISIONS

In order to avoid collisions, first a strict definition of collision is necessary. These vehicles are modeled as non-holonomic point masses, however real vehicles have finite size. Therefore in order to collide, the vehicles do not have to attain the same position in space at the same time, but rather 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 two vehicles when

$$\|\vec{r}\| < d_{sep}, \quad (9)$$

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

For two vehicles not 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 two vehicles (i and j) if they are not currently in a collision, but with zero control input ($u_i = u_j = 0$), will at some future point in time enter a collision:

$$\min_{t>0} \|\vec{r}(t)\| < d_{sep}. \quad (10)$$

Lemma 1: A necessary and sufficient condition for there to be no conflict is

$$|\beta| \geq \alpha, \quad (11)$$

where

$$\beta = \angle \vec{v} - \angle \vec{r}, \quad (12)$$

and

$$\alpha = \arcsin \left(\frac{d_{sep}}{\|\vec{r}\|} \right). \quad (13)$$

Proof: The minimum distance between the vehicles occurs at time t_c , when the derivative of that distance is zero:

$$\begin{aligned} \frac{d}{dt} \|\vec{r}(t)\|^2 &= -2\vec{v}^T \vec{r}(t) \\ 0 &= \vec{v}^T \vec{r}(t_c). \end{aligned} \quad (14)$$

To find t_c , multiply \vec{v}^T to both sides of (8) with $t = t_c$:

$$\begin{aligned} \vec{v}^T \vec{r}(t_c) &= \vec{v}^T \vec{r} - \vec{v}^T \vec{v} t_c \\ 0 &= \|\vec{v}\| \|\vec{r}\| \cos \beta - \|\vec{v}\|^2 t_c \\ t_c &= \frac{\|\vec{r}\|}{\|\vec{v}\|} \cos \beta. \end{aligned} \quad (15)$$

From (8), the minimum distance between the vehicles is

$$\|\vec{r}(t_c)\| = \sqrt{(\vec{r} - \vec{v}t_c)^T \vec{r}(t_c)}. \quad (16)$$

Using (14) and again substituting (8) results in

$$\|\vec{r}(t_c)\| = \sqrt{\vec{r}^T (\vec{r} - \vec{v}t_c)} \quad (17)$$

Substituting (15) and reducing algebraically yields

$$\begin{aligned}\|\vec{r}(t_c)\| &= \sqrt{\|\vec{r}\|^2 - \|\vec{v}\| \|\vec{r}\| \cos \beta \left(\frac{\|\vec{r}\|}{\|\vec{v}\|} \cos \beta \right)} \\ &= \sqrt{\|\vec{r}\|^2 (1 - \cos^2 \beta)} \\ &= \|\vec{r}\| |\sin \beta|.\end{aligned}\quad (18)$$

If there is to be no conflict, then (10) must be reversed:

$$\begin{aligned}d_{sep} &\leq \min_{t>0} \|\vec{r}(t)\| \\ &= \|\vec{r}(t_c)\| \\ &= \|\vec{r}\| |\sin \beta|.\end{aligned}\quad (19)$$

Therefore, to remain free of conflict:

$$|\beta| \geq \arcsin \left(\frac{d_{sep}}{\|\vec{r}\|} \right) = \alpha, \quad (20)$$

which is depicted in Fig. 1. The angle α represents the half-width of the collision cone ([1], [2], [3]).

Likewise, if $|\beta| \geq \alpha$, then (20) and (19) still hold, thus implying that there is no conflict. ■

The collision cone becomes more useful to this algorithm when it is defined in a different reference frame. Rather than determining allowed directions for the relative velocity vector, a more useful tool is the set of conflict-free headings for vehicle i relative to vehicle j . Fig. 2 shows graphically how to convert from one collision cone frame to the other. The new collision cone is defined by the angles of its two edges, ψ^+ and ψ^- , which are:

$$\psi^\pm = \pi - \psi_j + 2(\angle \vec{r} \pm \alpha), \quad (21)$$

for

$$|\angle \vec{r} \pm \alpha - \psi_j| < \frac{\pi}{2}. \quad (22)$$

This condition is equivalent to the circle intersecting that side of the collision cone (see Fig. 2). If that condition is not met then (21) takes on its limiting value corresponding to (22) attaining equality:

$$\psi^\pm = \psi_j. \quad (23)$$

If both ψ^+ and ψ^- fail condition (22), then no collision cone exists (i.e. no matter which way vehicle i turns it can never collide with vehicle j). In this case ψ^+ and ψ^- do not exist.

The following definition provides a useful metric to determine the nearness to a conflict.

$$\gamma_{ij}^\pm \equiv \pm \psi_i \mp \psi^\pm. \quad (24)$$

These angle metrics are depicted in Fig. 2. To make the signs meaningful, γ^+ and γ^- are wrapped to the domain $(-2\pi, 2\pi)$ such that they are positive when the vehicles are not in conflict and negative otherwise.

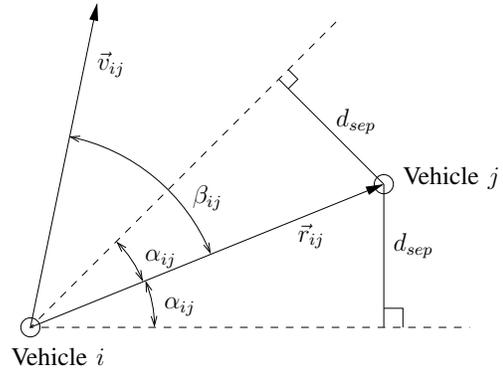


Fig. 1. Geometric definition of the collision cone. The area between the two dotted lines is the collision cone; a conflict occurs when the relative velocity vector, \vec{v}_{ij} , lies within this area.

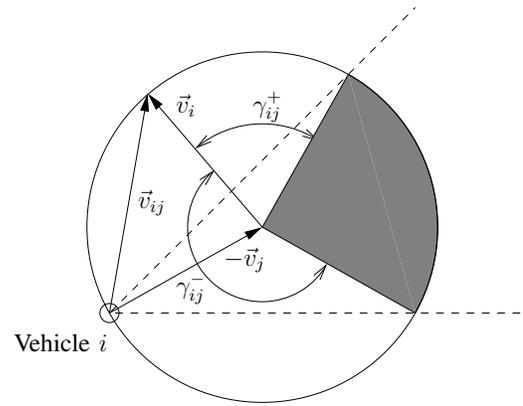


Fig. 2. Collision cone in different coordinates. The dotted lines represent the collision cone from Fig. 1. The circle denotes the entire range of headings for vehicle i . The shaded area represents the collision cone relative to the heading of vehicle i . γ_{ij}^+ denotes the most vehicle can turn to the right and γ_{ij}^- denotes the most it can turn to the left.

IV. AVOIDANCE ALGORITHM

The concept behind this algorithm is for each vehicle to adjust its heading until there are no conflicts between any of the n vehicles. At that point, each vehicle can use its desired control input unless that input would cause it to come into conflict with another vehicle. Transition regions are created around the boundaries of the collision cone so that the control can transition smoothly (finite gain).

First, the distance to the nearest conflict on the right (γ_i^+) and to the nearest conflict on the left (γ_i^-) are defined by

$$\begin{aligned}\gamma_i^+ &= \min_j \left\{ \gamma_{ij}^+ \mid \beta_{ij} \geq 0 \right\} \\ \gamma_i^- &= \min_j \left\{ \gamma_{ij}^- \mid \beta_{ij} < 0 \right\}.\end{aligned}\quad (25)$$

The control input is found using the control function, F :

$$u_i = F(\gamma_i^+, \gamma_i^-). \quad (26)$$

The control function chosen for this algorithm is piecewise-linear, defined by the following ordered triples of

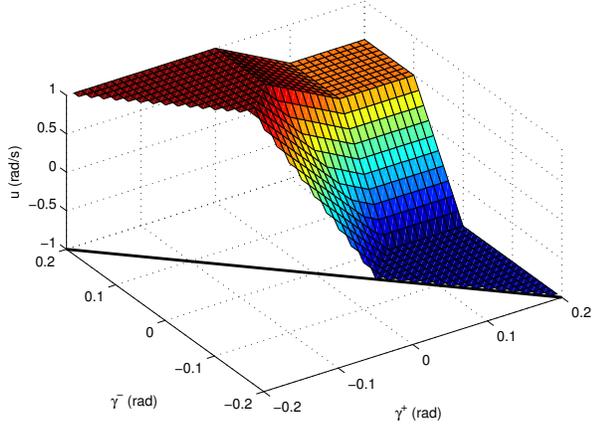


Fig. 3. Example of the control function, F . In this case, $u_d = 0.5$ rad/s, $u_{max} = 1$ rad/s, and $\epsilon = 0.1$ rad.

the form $(\gamma_i^+, \gamma_i^-, u_i)$:

$$\begin{pmatrix} \epsilon, \epsilon, u_d \\ 2\pi, \epsilon, u_d \\ \epsilon, 2\pi, u_d \end{pmatrix} \begin{pmatrix} 0, \epsilon, u_{max} \\ 0, 2\pi, u_{max} \\ -\frac{\epsilon}{2}, \frac{\epsilon}{2}, u_{max} \end{pmatrix} \begin{pmatrix} \epsilon, 0, -u_{max} \\ 2\pi, 0, -u_{max} \\ \frac{\epsilon}{2}, -\frac{\epsilon}{2}, -u_{max} \end{pmatrix} \quad (27)$$

Because F is a function of the desired control, u_d must be saturated such that

$$|u_d| \leq u_{max}. \quad (28)$$

An example of this control function is shown in Fig. 3.

The intuition behind the choice of control function is as follows. First, if the vehicle is in conflict with another vehicle (i.e. $\gamma_i^\pm < 0$), then that vehicle should exert maximum control authority to get out of the conflict. If the vehicle is sufficiently far from a conflict ($\gamma_i^\pm > \epsilon$), it should be allowed to use its desired control, u_d . The parameter ϵ defines the size of the transition region, or effectively the gain of the controller. The rest of the function must connect these areas in a continuous fashion so that the control input will also be continuous.

Any γ_i^\pm which does not exist can be replaced by π (any value greater than ϵ can be used because any value greater than ϵ is effectively ignored by the control function).

When vehicle i starts in conflict with more than one other vehicle the collision cones must all be considered at once, rather than individually. Therefore all overlapping cones must be united into a single collision cone. This step ensures that each vehicle can determine the nearest conflict-free heading in a global sense. γ_i^\pm is still found from (25), but now using only the γ_{ij}^\pm s which define the edges of the united cone. Because any other collision cones that are left must be disjoint from the united cone, γ_i^\pm must satisfy:

$$\gamma_i^+ + \gamma_i^- \geq 0. \quad (29)$$

Therefore the control function need only be defined over the domain defined by (29). The edge of this domain is represented in Fig. 3 by the black line.

A. Staying Out of Conflict

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

Proof: For the case when (22) holds, γ_i^\pm can be found from (25) and (21):

$$\gamma_i^\pm = \pm\psi_i \pm \psi_j \mp 2\angle\vec{r} - 2\alpha, \quad (30)$$

which has derivative,

$$\frac{d\gamma_i^\pm}{dt} = \pm u_i \pm u_j \mp 2 \frac{d\angle\vec{r}}{dt} - 2 \frac{d\alpha}{dt}. \quad (31)$$

For the case when (22) is not satisfied, the derivative is

$$\frac{d\gamma_i^\pm}{dt} = \pm u_i \mp u_j. \quad (32)$$

From the geometry,

$$\frac{d\angle\vec{r}}{dt} = -\frac{\|\vec{v}\|}{\|\vec{r}\|} \sin \beta. \quad (33)$$

The derivative of α is somewhat less straight-forward:

$$\begin{aligned} \frac{d\alpha}{dt} &= \frac{d}{dt} \left(\arcsin \left(\frac{d_{sep}}{\|\vec{r}\|} \right) \right) \\ &= \frac{d}{dt} \left(\frac{d_{sep}}{\|\vec{r}\|} \right) \left(1 - \left(\frac{d_{sep}}{\|\vec{r}\|} \right)^2 \right)^{-1/2} \\ &= \frac{d_{sep} \|\vec{v}\| \cos \beta}{\|\vec{r}\|^2} \left(\frac{\|\vec{r}\|}{\sqrt{\|\vec{r}\|^2 - d_{sep}^2}} \right) \\ &= \frac{\|\vec{v}\|}{\|\vec{r}\|} \cos \beta \tan \alpha. \end{aligned} \quad (34)$$

Now (31) becomes

$$\frac{d\gamma_i^\pm}{dt} = \pm u_i \pm u_j + 2 \frac{\|\vec{v}\|}{\|\vec{r}\|} (\pm \sin \beta - \cos \beta \tan \alpha). \quad (35)$$

The definition of γ_i^\pm constrains the sign of β (25), so (35) is equivalently

$$\frac{d\gamma_i^\pm}{dt} = \pm u_i \pm u_j + 2 \frac{\|\vec{v}\|}{\|\vec{r}\|} (|\sin \beta| - \cos \beta \tan \alpha). \quad (36)$$

Recalling Lemma 1, for the system to be conflict-free, $|\beta| \geq \alpha$. Therefore $|\tan \beta| \geq \tan \alpha$, and so

$$|\sin \beta| - \cos \beta \tan \alpha \geq 0. \quad (37)$$

Any continuous control function that ensures

$$\begin{aligned} \gamma_i^+ = 0 &\implies u_i \geq 0 \\ \gamma_i^- = 0 &\implies u_i \leq 0, \end{aligned} \quad (38)$$

also ensures that

$$\frac{d\gamma_i^\pm}{dt} \geq 0 \quad \forall \quad \gamma_i^\pm = 0, \quad (39)$$

which means the system cannot enter a conflicted state. The reason u_j cannot change this fact is because every vehicle was assumed either to use this algorithm, or at worst have

zero control input. Because a conflict is symmetric (the vehicles always agree on whether a conflict exists between them), the sign of u_j must be such that both $\frac{d\gamma_j^\pm}{dt} \geq 0$ and $\frac{d\gamma_i^\pm}{dt} \geq 0$.

The control function used in this algorithm (27) satisfies (38) and so this 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 and u_{max} , so long as $u_{max} \geq 0$ and $|u_d| \leq u_{max}$ at every instant. ■

B. Conflict Resolution

The previous proof uses the assumption that the system starts conflict-free, which might seem to defeat the purpose of deconfliction. After all, why bother with a deconfliction algorithm if the vehicles are not in conflict in the first place? The first answer is that this algorithm allows changing desires to be integrated into the solution while maintaining collision avoidance (desires which might otherwise cause collisions). The second answer is that this algorithm also attempts to resolve vehicles from an initially conflicted state. Its performance in this task is good, but not proven. However, if at any point in time this algorithm manages to bring the system to a conflict-free state, then Theorem 1 applies and the system will stay free of collisions from that time forward.

The reason actual deconfliction is difficult to prove is that there exists a space of initial conditions for which deconfliction is impossible. A trivial example is when vehicles start less than d_{sep} apart, but even if intervehicle separation is larger than this, high speed (v) and low turning rate limit (u_{max}) can make it impossible to turn fast enough to avoid a collision. For a two vehicle system, a reasonable bound can be found for the acceptable initial conditions. However, bounding the n -vehicle problem is much more difficult and the feasibility of finding such a bound is still in question.

Theorem 2: The algorithm described above, when implemented on two vehicles with dynamics (1) and input constrained by (2), will ensure those vehicles never collide, provided they are initially separated by at least:

$$d_{min} = d_{sep} \sqrt{\frac{4v}{u_{max}d_{sep}} + 1}. \quad (40)$$

Proof: Taking the worst case scenario of the two vehicles heading directly toward each other (with $\beta = 0$), (25) shows that γ_i^+ will be negative and γ_i^- will not exist (and will be set to π by the algorithm) for both vehicles. The control function (27) therefore will choose $u_1 = u_2 = u_{max}$.

Note that β will be the same for both vehicles, and hence whatever the sign of β , both vehicles will turn in the same direction to avoid the conflict. Also note from (36) that having like signs on the control inputs is the best way to maximize $\frac{d\gamma_i^\pm}{dt}$, thus exiting from a state of conflict as quickly as possible.

The radius of curvature for each vehicle's path is $\frac{v}{u_{max}}$. The minimum distance between two arcs is measured along a line connecting their centers. Fig. 4 shows the geometry.

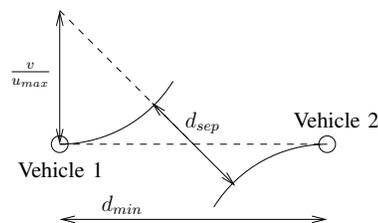


Fig. 4. Geometry of the worst case scenario for two vehicles.

The minimum allowed separation distance can be calculated with the Pythagorean Theorem:

$$d_{min}^2 + \left(\frac{2v}{u_{max}}\right)^2 = \left(\frac{2v}{u_{max}} + d_{sep}\right)^2 \quad (41)$$

which is equivalent to (40). ■

C. Performance

The collision avoidance algorithm described above will allow each vehicle to follow its desired control (u_d) whenever this action does not cause a conflict. Due to the construction of (27), so long as a vehicle is at least ϵ away from a conflict on either side then it is allowed to follow its desire exactly.

Most often, γ_i^\pm satisfies (22), which from (36) implies that the collision cone will recede from the vehicle, eventually freeing it to follow its desired control. Likewise, once two vehicles pass and start heading away from each other, their collision cones will cease to exist. However, if γ_i^\pm does not satisfy (22), then (32) shows that the collision cone will not recede. This situation happens when two vehicles are moving nearly parallel to each other and their desired controls would turn them toward each other. The algorithm will not allow this to happen, but rather will cause both vehicles to turn at the average of their two desired control inputs. This maneuver will eventually cause the collision cone to vanish for one vehicle and to satisfy the condition (22) for the other vehicle, which will allow them to pass and then use their desired control inputs.

However, a problem can occur when the situation just described happens but the vehicles have equal and opposite desired controls (generally implying they are both saturated). Then the average turning rate is zero, and the vehicles will simply move parallel to each other forever, never attaining their desired control inputs. A simple change to the saturation function can be used to break this symmetry. Adjusting (28) to

$$\frac{-u_{max}}{2} \leq u_d \leq u_{max} \quad (42)$$

removes the symmetry, and thus vehicles in the situation just described will favor turning to the left. This adjustment comes at some expense of performance, since now the vehicles cannot use their full control authority in turning to the right (unless they are in conflict). The ratio of the asymmetry is arbitrary and hence can be used to tradeoff between these two aspects of performance.

While this algorithm generally causes the vehicles to eventually return to their desired controls, this behavior

cannot be assured. For instance, in the case of conflicting desired controls (e.g. two vehicles heading for the same waypoint), collision avoidance takes precedence and the vehicles may never reach their goals. Likewise, it is possible to design a feedback desired controller that acts to stabilize an otherwise unstable equilibrium (such as that discussed in the previous paragraph), thus making the vehicles block each other from their goals. These issues should be considered when designing a desired controller for this algorithm.

Additionally, because this algorithm couples the vehicles there is a possibility for instabilities to develop, especially in the presence of delays. A full stability analysis is beyond the scope of this paper, but the idea is that the coupling gain can be lowered by increasing ϵ , thus slowing the response and increasing the robustness to delays and other disturbances.

V. SIMULATIONS

For the sake of demonstrating how this algorithm causes the n -vehicle system to avoid conflicts, remain collision-free and still follow desired control input, another controller must be employed to create a desired control input. For these simulations, the vehicles are performing path following. Each vehicle's desired path is defined as a straight line emanating from its initial position, oriented along its initial heading, $\psi(0)$. A simple path-following algorithm is used to compute a desired control:

$$u_d = k_2 \left(\psi - \psi(0) - \arctan \left(\frac{d}{k_1} \right) \right), \quad (43)$$

where d is the perpendicular distance between the vehicle and the path. The parameters used for these simulations are: $v = 1$ unit/s, $d_{sep} = 2$ units, $u_{max} = 1$ rad/s, $\epsilon = 0.1$ rad, $k_1 = 1$ unit, $k_2 = 5$ Hz.

The first simulation included here is of a symmetric conflict scenario. Six vehicles are arranged evenly around a circle of radius 8 units, all pointed directly toward the center. The results of running this algorithm on each of the vehicles is shown in Fig. 5 and corresponding control inputs are shown in Fig. 6. Each of the vehicles is initially in conflict with every other vehicle, so they begin by turning at maximum rate. After only a quarter-second all of the vehicles have attained a conflict-free state. At this point the guarantees of Theorem 1 apply, and indeed the vehicles remain conflict and collision-free for the rest of the simulation. One can see the effect of the receding collision cones in Fig. 6 at about 8 seconds when the control is allowed to rise. Once the desired control changes sign, it is no longer pushing the vehicles toward a conflict, so they are allowed to follow it.

The second simulation (shown in Fig. 7) is similar to the first, but with randomized initial conditions. The control inputs of all the vehicles are shown in Fig. 8. The vehicles again start in conflict with all the other vehicles, but within one second all conflicts have been resolved. From that point on the vehicles are attempting to return to their desired paths, but can only do so as they pass other vehicles. Figure 9 shows how the collision cones change with time from the point of view of the cyan vehicle. Once a conflict-free state is

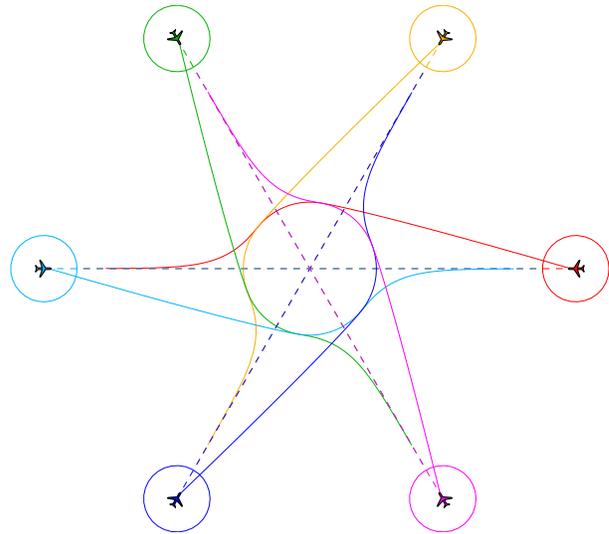


Fig. 5. Symmetric 6-vehicle collision avoidance simulation. Vehicles start 8 units from the center and have a size of radius 1 unit (depicted by the circles), representing a minimum approach distance, d_{sep} , of 2 units. The dotted lines represent the desired paths.

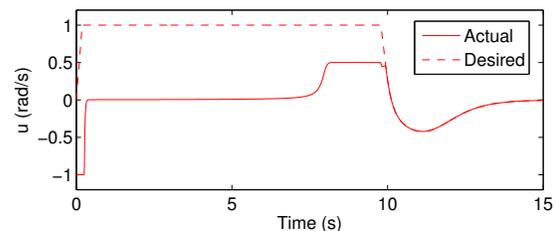


Fig. 6. Control input, u , and desired control, u_d , for the simulation shown in Fig. 5. For this symmetric case all vehicles have identical control inputs.

achieved the controller causes a quasi-steady-state to develop that keeps $\gamma_i^\pm \approx \frac{\epsilon}{2}$ until the collision cones recede and eventually vanish.

VI. CONCLUSION

This work has developed a decentralized control algorithm for deconflicting n unicycle-type vehicles. This 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 was found for the initial separation distance between two vehicles such that collision avoidance is assured even when starting in conflict. Finally, this algorithm allows the vehicles to follow changing desired controls so long as safety is not sacrificed.

Many extensions are possible for this work. First is to make the vehicle group heterogeneous (different speeds and turning rate limits). Next will be to allow speed variations and use this as another degree of freedom for avoiding collisions (especially important for mobile robot applications). For applications to aircraft and submarines, extending this concept to three dimensions will again add a degree of

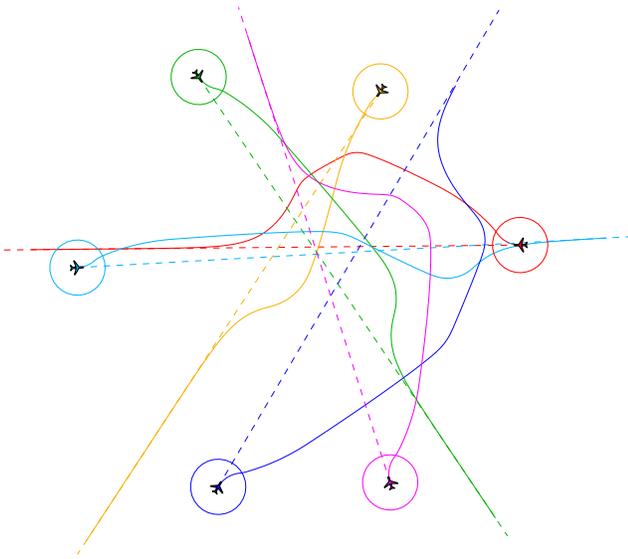


Fig. 7. Random 6-vehicle collision avoidance simulation. The dotted lines represent the desired paths.

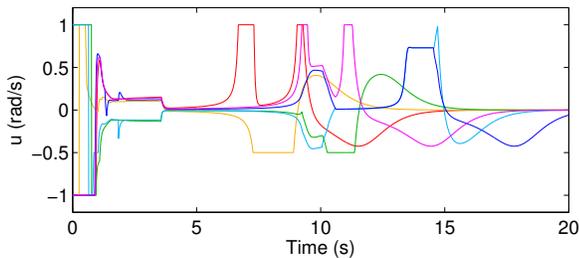


Fig. 8. Control input, u_i , for the simulation shown in Fig. 7.

freedom and hence increase the performance of the system. Finally, this algorithm shows promise for evading adversarial pursuers, and may lead to interesting results regarding pursuit/evasion games. These areas are all currently under investigation.

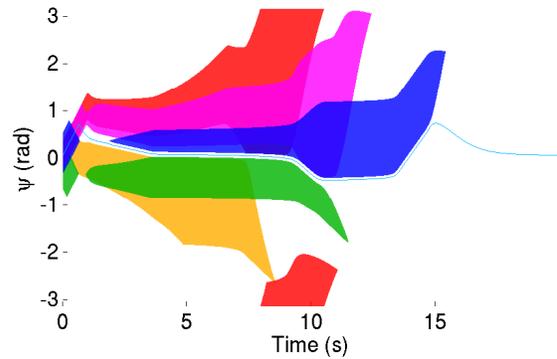


Fig. 9. Collision cones from the point of view of the cyan vehicle for the simulation shown in Fig. 7. The line is the cyan vehicle's heading, while the shaded regions represent the set of headings of the cyan vehicle that would result in conflict.

REFERENCES

- [1] C. Carbone, U. Ciniglio, F. Corrado, and L. Luongo, "A novel 3d geometric algorithm for aircraft autonomous collision avoidance," *IEEE Conference on Decision and Control*, 2006.
- [2] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: A collision cone approach," *IEEE Transactions on Systems, Man, and Cybernetics*, 1998.
- [3] 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*, 2001.
- [4] J. Kuchar and L. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Transactions on Intelligent Transportation Systems*, 2000.
- [5] I. Hwang and C. Tomlin, "Protocol-based conflict resolution for air traffic control," Stanford University, Tech. Rep. SUDAAR-762, 2002.
- [6] K. Bilimoria, B. Sridhar, and G. Chatterji, "Effects of conflict detection methods for air traffic management," *AIAA Guidance, Navigation, and Control Conference*, 1996.
- [7] FAA, "Precision runway monitor demonstration report, Tech. Rep. DOT/FAA/RD-91/5, Feb. 1991.
- [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*, 1998.
- [9] J. Andrews, "A relative motion analysis of horizontal collision avoidance," *SAFE Journal*, vol. 8, 1978.
- [10] M. Eby and W. Kelly, "Free flight separation assurance using distributed algorithms," *IEEE Aerospace Conference*, pp. 429–441, 1999.
- [11] J. Kosecka, C. Tomlin, G. Pappas, and S. Sastry, "Generation of conflict resolution maneuvers for air traffic management," *Int. Conf. Intell. Robot. Syst.*, pp. 1598–1603, 1997.
- [12] K. Zeghal, "A review of different approaches based on force fields for airborne conflict resolution," *AIAA Guidance, Navigation, and Control Conference*, pp. 818–827, 1998.