Reactive Trajectory Generation for Multiple Vehic le s in Unknown Environments With Wind Disturbances

Kenan Cole ♦ and Adam M. Wickenheiser ♦, Member, IEEE

Abstract—Unmanned aerial vehicle use continues to increase, including operating beyond line of sight in unknown environments where the vehicle must autonomously generate a trajectory to safely navigate. In this article, we develop a trajectory generation algorithm for vehicles with second-order dynamics in unknown environments with bounded wind disturbances in which the vehicle only relies on its on-board distance sensors and communication with other vehicles to navigate. The proposed algorithm generates smooth trajectories and can be used with high-level planners and low-level motion controllers. The algorithm computes a maximum safe cruise velocity for the vehicle in the environment and guarantees that the trajectory does not violate the vehicle’s thrust limitation, sensor constraints, or user-defined clearance radius around other vehicles and obstacles. Additionally, the trajectories are guaranteed to reach a stationary goal position in finite time given a finite number of bounded obstacles. Simulation results demonstrate the algorithm properties through two scenarios. First, a quadrotor navigating through a moving obstacle field to a goal position and, second, multiple quadrotors navigating into a building to different goal positions.

Index Terms—Autonomous agents, collision avoidance, motion control, path planning for multiple mobile robot systems, robot navigation.

I. INTRODUCTION

Unmanned aerial vehicles continue to become more prolific, with a new focus on enabling autonomous navigation. The push for beyond-line-of-sight operation is becoming more of a reality with improved sensors such as miniature radars weighing as little as 120 g with ranges on the order of hundreds of meters [1], [2]. Additionally, laser range finders weighing as little as 120 g provide 360° coverage and with ranges up to 40 m [3]. Associated with using this technology are the challenges of autonomous sense and avoid, how to operate in unknown and potentially harsh environments, and how to compensate for hardware constraints such as maneuverability and sensor limitations. These constraints are particularly important for vehicles with second-order dynamics where the vehicle cannot turn instantaneously, so the trajectory generation algorithm must compensate. Collision-free trajectory generation to a goal position for each vehicle under hardware limitations is the focus of this article.

There are several approaches to trajectory generation in the presence of obstacles and/or other vehicles. Hoy et al. [4] provide a good summary article of various approaches and desirable algorithm features. The most popular approaches include global planners, local and reactive planners, and formation controllers. In the trajectory generation literature, global optimization techniques are prevalent [5]–[7] because for a known environment, they can ensure convergence to the goal position. Global optimization is not possible for our application where the environment is unknown and dynamic.

Local planners are similar to global planners but examine a shorter time window to reduce the computational expense. They can also address obstacles that may not be known a priori. For example, Alonso-Mora et al. [8] take the trajectory from a global planner and locally modify it to address any additional constraints based on other vehicle motion. Shiller et al. [9] take a similar approach by optimizing the trajectory around immediate obstacles. One of the main drawbacks to local planners is the lack of an overall safety or convergence guarantee since the optimization is occurring for short time windows for only the closest obstacles.

Reactive controllers are a type of local planner that generate the trajectory directly as the environment is sensed. These approaches utilize distance sensors to determine course changes [10]–[12]. While these solutions are generally not optimal, they are typically computationally faster than the optimized solutions and do not require convergence of an optimization algorithm to generate a viable solution. Their drawback however is that they do not address the smoothness of the trajectory. This can be problematic if the desired navigation requires more thrust than that the vehicle can produce, and/or if the higher derivatives of the trajectory are not bounded, which may violate vehicle controller requirements.

Formation controllers typically govern the motion of multiple vehicles using a reduced set of parameters or states and also provide solutions for collision avoidance. Examples of collision avoidance methods include potential fields [13], decentralized cooperation through sharing possible trajectory sets [14], and navigation of the formation as a rigid body [8], [15], [16]. In the scenario we consider, the behavior is more similar to swarms, which may change composition and formation, and are defined
by only a few parameters. Groups such as [17] consider swarm behavior in obstacle-free environments, whereas [18] relies on a distributed optimization between the vehicles to avoid obstacles and maintain the formation. Our algorithm assures the safety of vehicles in the formation and also smoothly and safely navigates retasked vehicles out of the formation. In addition, our algorithm can be applied to clusters of formations of vehicles with their own clearance radii.

The physical limitations of the vehicle, such as maneuverability, sensing, and control input constraints, must also be considered to ensure the generated trajectory is feasible. In the literature there are various works that consider limitations, such as sensor range (see [10], [11], [19]–[22]), maximum velocity (see [11], [12], [14], [21], [22]), clearance radius (see [8], [10], [12], [19], [21], [22]), and turning rate (see [10], [19], [21], [22]). Setting bounds on only a subset of these parameters may be reasonable for certain environments; however, all parameters are important in potentially harsh and unknown environments to ensure that the trajectory is not too aggressive. Of the works reviewed, only a few consider all of these constraints simultaneously, but none consider environmental disturbances as input to the trajectory generation. Examination of disturbances is much more prevalent in vehicle controller literature to show ultimate bounded or asymptotic stability [23]–[26]. To achieve these stability guarantees, the controllers require the desired trajectory higher derivatives to exist and be bounded. In order to meet these criteria, the control authority to overcome the disturbance must also be considered when generating the trajectory.

To address each of these areas, we build upon [27], which describes trajectory generation for groups of quadrotors in unknown environments that bounds the maximum cruise velocity and respects thrust limitations and sensor constraints. In this article, we establish guarantees with respect to the obstacle spacing, which was not explicitly considered in the prior work. Additionally, we look at the maneuverability of vehicles separate from the obstacles to enable more aggressive maneuvering. Finally, we account for the goal position more explicitly when determining course and velocity changes to reduce the trajectory length/time.

We organize the rest of the paper by first defining the problem, algorithm properties, and operating assumptions in Section II. The trajectory generation is defined in Section III, which describes how to smoothly adjust vehicle course and/or velocity to safely clear obstacles and other vehicles. Section IV provides the analysis for bounding the trajectory acceleration to respect thrust limitations and bounding the maximum cruise velocity to safely navigate the environment. The vehicle dynamics and controller for the simulation case study are given in Section V. Two simulation case studies in Section VI demonstrate the trajectory generation algorithm’s features. Finally, Section VII provides concluding remarks.

II. PROBLEM DEFINITION

We define an algorithm that generates a trajectory for each vehicle that satisfies Properties 1 and 2 for an environment similar to Fig. 1. These properties are rigorously achieved, as shown in Section IV, under the following assumptions, some of which may be relaxed as discussed in Section VII.

A. Algorithm Properties

Property 1: Generation of a piecewise-smooth (with isolated bounded discontinuities) desired trajectory \( p_d : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \) where the derivatives \( p_d^{(i)} : \mathbb{R}^3, \forall i = 0, 1, \ldots, n \) exist, are bounded, and respect the vehicle’s maximum thrust \( f_{\text{max}} \) for a translational wind velocity of unknown direction and bounded magnitude, \( ||v_{\text{air}}|| \leq v_{\text{air, max}} \).

Property 2: Clearance of all obstacles and other vehicles by a user-defined clearance radius \( r_c \) which takes into account the vehicle’s size as well as measurement, estimation, and tracking errors.

B. Algorithm Assumptions

Assumption 1: Vehicle desired trajectories and obstacle motions are planar, but vehicle dynamics are not restricted to be planar.

Assumption 2: Vehicles are finite in number and heterogeneous in physical parameters (mass, max thrust, etc.) and importance (e.g., higher valued asset).

Assumption 3: Vehicles share current position and course information when in range via wireless communication.

Assumption 4: Vehicles’ sensor and communication sample periods and ranges are equal and given by \( \Delta T \) and \( r_s > r_c \), respectively. Within these limitations, the sensor and intervehicle communications provide perfect distance and velocity information.

Assumption 5: The clearance radius \( r_c \) ensures that there are no aerodynamic interactions between one vehicle and another or with obstacles.

Assumption 6: Wind disturbances are bounded, time varying, and planar. Updraft effects near obstacles are assumed to be limited to a distance less than \( r_c \).

Assumption 7: There are a finite number of obstacles, and each obstacle is finite size and moves with constant velocity (less than minimum vehicle cruise velocity) and constant course.
Minimum obstacle separation does not prevent the vehicles from moving between them.

Assumption 8: Goal positions are not too close to obstacles or each other to violate vehicle clearance radii and are not infinitely far from the coordinate origin.

III. TRAJECTORY GENERATION

The trajectory generation algorithm takes each vehicle from its starting position and velocity, and guides it on a collision-free trajectory to the goal position. To achieve this, the vehicle first determines which vehicles within a distance \( r_c \) it is responsible for maneuvering around. Next, if the vehicle has thrust availability for maneuvering, it compiles all sensor/communication inputs to identify the most imminent obstacle/vehicle safety threat. The vehicle then computes a circumnavigation direction to traverse the obstacle/vehicle, a course change angle, and a velocity change to maintain the desired clearance radius, \( r_c \). Finally, the vehicle uses sigmoid functions to smoothly transition to the desired course and velocity. These steps are discussed in detail in Sections III-A to III-H.

A. Ranking Vehicles’ Maneuverability

To determine which vehicles maneuver and which vehicles stay on course, vehicles exchange their maximum cruise velocity \( v_c \), current velocity \( p_k \), clearance radius \( r_c \), and a preassigned ID value when they come within communication range of each other. To satisfy Assumption 7, vehicles with larger \( v_c \) must maneuver around vehicles with smaller \( v_c \). If the vehicles have equal \( v_c \) values, then the vehicles with lower ID values maneuver around vehicles with higher ID values, forming the set \( I_{mnvr} \subseteq I_{at} \), where \( I_{at} \) is the set of all vehicles within \( r_c \) of the vehicle’s current position. Hovering or loitering vehicles are considered to have \( v_c = 0 \).

B. Compiling Sensor Inputs

The vehicle uses distance and angle measurements to obstacles and other vehicles to determine the most imminent collisions, if any. We assume that the sensing is isotropic (i.e., has the same range and rate in all directions). The sensor output is a data array of relative positions of sensed points on obstacles. By finding discontinuities in range and angle, the sensor scan information is used to distinguish different obstacles, each of which is given a unique local identifier, id. The id values of all obstacles within the sensor scan comprise the set \( I_{obs} = \{id_1, \ldots, id_m\} \), where \( m \) is the number of distinct obstacles within range. The inertial positions of the sensed points are given by \( p_{id,i} \), where \( i = 1, \ldots, n_{id} \), and \( n_{id} \) is the number of sensed points for that particular obstacle.

The intervehicle communication provides inertial positions, \( p_{id} \), in addition to the data described in Section III-A. The data for the vehicles in \( I_{mnvr} \) is combined with the data for the obstacles in \( I_{obs} \) to form a data array of distinct vehicles and obstacles that is used to determine appropriate course and/or velocity changes for collision-free navigation in the environment.

C. Critical Obstacle and Vehicle Identification

Now that the vehicle has compiled its sensor and communication inputs, it identifies critical and noncritical obstacles/vehicles in the environment. Critical obstacles/vehicles are within the minimum reaction distance \( \Delta T_{tot} \) (defined in (2)), require immediate action from the vehicle to avoid collisions and violations of \( r_c \), and if there are multiple critical obstacles/vehicles then all contribute to the course change. Noncritical obstacles are outside the minimum reaction distance and contribute to course changes when there are no critical obstacles or the critical obstacles do not prohibit the vehicle from navigating to the goal position. This section describes the process to determine the sets \( I_{co}, I_{cv}, I_{mnvr}, I_{at} \), which are the sets of critical obstacles and vehicles, and noncritical obstacles and vehicles, respectively.

The vehicle first determines the closest sensed point for the \( k \)th obstacle/vehicle in \( I_{obs} \cap I_{at} \)

\[
p_{k_{\text{min}}} = \underset{i}{\text{argmin}} \left( ||p_{k,i} - p_d|| \right)
\]

where \( i = 1, \ldots, n_k \) are the indices of the sensed points for obstacle/vehicle \( k \).

Since the other vehicles can change course and speed whereas the obstacles have constant course and speed, the minimum reaction distance to maintain \( r_c \) and avoid collisions is different for obstacles and vehicles. We define the minimum reaction distance to avoid obstacle/vehicle \( k \) as

\[
r_{c,k} = \left\{ \begin{array}{ll}
\text{obs}, & k \in I_{obs} \\
\max(r_{c,k}, r_c) + r_{180} + ||\dot{p}_k||\Delta T_{tot}, & k \in I_{at}
\end{array} \right.
\]

where \( r_{obs} \) is the minimum distance between obstacles, \( r_c \) is the clearance radius of the current vehicle, \( r_{c,k} \) is the clearance radius of sensed vehicle \( k \), \( \Delta T_{tot} = \tau_f + \Delta T_c \), \( \dot{p}_k \) is the velocity vector of vehicle \( k \), \( \tau_f = v_c \int_0^{r_{180}} \sin(\phi(t)) dt \) and \( r_{180} = \frac{c_3}{a_{\text{max}}} \tau_f v_c \) are the distance and time span required for the vehicle to make a 180° turn (\( v_c \) is defined in Theorem 2 below, \( \phi(t), c_3, a_{\text{max}} \) are defined in Section III-H), respectively, and \( \Delta T_c < \Delta T_{tot} \) is the maximum on-board algorithm computation time. Appendix C of [28] discusses the development of the definition of \( r_{c,k} \).

To accurately determine which obstacle/vehicle poses the most imminent threat, we normalize the distance from the vehicle to \( p_{k_{\text{min}}} \) by the corresponding minimum reaction distance, \( r_{c,k} \). The obstacle/vehicle that minimizes (4) is the most imminent threat

\[
p_{\text{min}} = p_{k_{\text{min}},\text{min}}
\]

where

\[
k_{\text{min}} = \underset{k}{\text{argmin}} \left( \frac{||p_{k_{\text{min}}} - p_d|| - r_{c,k}}{r_{c,k}} \right)
\]

Since \( r_c < r_{c,k}^{*} \) for all cases, \( ||p_{k_{\text{min}}} - p_d|| - r_{c,k}^{*} \) can be negative and likely is negative while traversing an obstacle/vehicle. When it is negative, the obstacle/vehicle that the vehicle is traversing is defined as critical. The sets of critical
obstacles and vehicles are, respectively, defined as

\[
\mathcal{I}_{\text{co}} = \{ k \mid ||p_{k,\text{min}} - p_{d}|| < r_{e,k}, k \in \mathcal{I}_{\text{obs}} \} \\
\mathcal{I}_{\text{cv}} = \{ k \mid ||p_{k,\text{min}} - p_{d}|| < r_{e,k}^{*}, k \in \mathcal{I}_{\text{m}} \}.
\]

The noncritical sets of obstacles and vehicles are, \(\mathcal{I}_{\text{no}} = \mathcal{I}_{\text{obs}} - \mathcal{I}_{\text{co}}\) and \(\mathcal{I}_{\text{mv}} = \mathcal{I}_{\text{m}} - \mathcal{I}_{\text{cv}}\), respectively. These sets are used for determining appropriate course changes as discussed in Section III-F.

### D. Course Change Definition for an Obstacle

To safely navigate the environment and avoid collisions, the vehicle can change course and/or velocity. In this study, we assume that the vehicle travels at its maximum safe cruise velocity and makes course changes as the default behavior to try to minimize the time required to reach the goal position. The process for determining an appropriate course change applies to both critical and noncritical obstacles. This section describes the process to determine a course change angle \(\Delta \phi_{k}\) and the set of all feasible course angles \(\mathcal{O}_{k}\) for obstacle \(k\). There are a few differences in the process for obstacles compared to vehicles, so vehicles are discussed separately in Section III-E.

To start the process of determining a candidate course change, the vehicle takes the sensed points for each obstacle and determines the bounding extent points (i.e., the left and right most extent points), \(p_{k,e,1}\) and \(p_{k,e,2}\), and their corresponding projected extent points, \(p_{k,e,1}^{*}\) and \(p_{k,e,2}^{*}\), which take into account \(r_{e}\). This process is illustrated in Fig. 2 where the vehicle first computes the angle to the projected sensed points as follows:

\[
\phi_{e,i} = \phi_{1,i} + \phi_{2,i}
\]

where

\[
\phi_{1,i} = \angle(r_{k,i}, r_{k,i,1})
\]

\[
\phi_{2,i} = k_{\phi,e,1} \left( \phi_{1,i} \sin^{-1} \left( \frac{r_{e}}{||r_{k,i}||} \right) \right)
\]

\[
r_{k,i} = p_{k,i} - p_{d}
\]

\[
k_{\phi,e,i} = \begin{cases} 
  \text{sgn}(\phi_{1,i}), & |\phi_{1,i}| > 0 \\
  \text{sgn}(\angle(r_{k,i,\text{min}}, \hat{p}_{d})), & \phi_{1,i} = 0
\end{cases}
\]

where \(i = 1, \ldots, n_{k}\) is the index of sensed points for obstacle \(k\).

Note that (9) only produces a real result when \(||r_{k,i}|| \geq r_{e}\); however, Section IV guarantees this condition.

The bounding extent points are the points that produce the maximum and minimum \(\phi_{e,i}\) as shown in Fig. 2(b) and defined as

\[
p_{k,e,1} = \arg \max_{i} (\phi_{e,i})
\]

\[
p_{k,e,2} = \arg \min_{i} (\phi_{e,i}).
\]

The final point the vehicle calculates is the projected minimum point as shown in Fig. 2(b) and is defined as

\[
p_{k,\text{min}} = p_{k,\text{min}} + r_{e} ( -\hat{r}_{k,\text{min}} ).
\]

If there is only one sensed point for an obstacle, then the extent points and projected extent points are equal as shown in Fig. 2(c), and are defined as

\[
p_{k,e,1} = p_{k,e,1}^{*} = p_{k,1} + r_{e} \hat{r}_{k,1} + \phi_{k} ( -\hat{r}_{k,1} )
\]

\[
p_{k,e,2} = p_{k,e,2}^{*} = p_{k,1} + r_{e} \hat{r}_{k,1} - \phi_{k} ( -\hat{r}_{k,1} )
\]

where

\[
\phi_{k} = \cos^{-1} \left( \frac{r_{e}}{||r_{k,1}||} \right).
\]

Next, the vehicle uses the projected extent points to determine four candidate tangent directions per obstacle. The candidate tangent directions are shown in Fig. 3(a) and are summarized as

\[
p_{k,s,1} = p_{k,e,1}^{*} - p_{k,\text{min}}
\]

\[
p_{k,s,2} = p_{k,e,1}^{*} - p_{d}
\]

\[
p_{k,s,3} = p_{k,e,2}^{*} - p_{k,\text{min}}
\]

\[
p_{k,s,4} = p_{k,e,2}^{*} - p_{d}.
\]

The \(p_{k,s,1}\) and \(p_{k,s,3}\) tangent directions are “conservative” because they define the slope based on the projected minimum point, thus keeping the vehicle parallel with the estimated obstacle “face”. The \(p_{k,s,2}\) and \(p_{k,s,4}\) tangent directions are “aggressive” because they allow the vehicle to get closer to the obstacle by heading toward the tangent point on the \(r_{e}\) circle.

We use the tangent directions to determine the obstacle velocity components parallel to the tangent directions (i.e., along the obstacle “face”) \(p_{k,\perp,i}\) and perpendicular to the tangent directions (i.e., normal to the obstacle “face”), \(p_{k,\perp,i}\), for \(i = 1, \ldots, 4\). The vehicle must match the component of velocity in the \(p_{k,\perp,i}\) direction to avoid collisions, and then use any remaining velocity to traverse the obstacle in the \(p_{k,i}\) direction. We define these quantities in the following paragraphs, where (23)–(33) are evaluated for all four candidate tangent directions, but for brevity, only the \(p_{k,s,1}\) equations are presented.

The unit vectors parallel and perpendicular to the tangent direction and the corresponding obstacle velocity components are defined as

\[
p_{k,s,1}^{*} = p_{d} + \sqrt{||r_{k,i}||^2 - r_{e}^2} R_{\phi_{e,1},i} \hat{r}_{k,\text{min}}
\]

where \(i = 1, \ldots, n_{k}\) is the index of sensed points for obstacle \(k\).
are, respectively, given by

$$\begin{align*}
\mathbf{p}_{k,s} = &\mathbf{p}_{k,s}^{1}, \\
\mathbf{p}_{k,\bot} = &R_{\theta_{k}^{1}}\mathbf{p}_{k,s}^{1}, \\
v_{k,s} = &\mathbf{p}_{k,s}^{1}, \\
v_{k,\bot} = &\mathbf{p}_{k,\bot}^{1},
\end{align*}$$

where

$$\theta_{k,s} = \text{sgn}(\langle \mathbf{p}_{k,s}^{1} \times (\mathbf{p}_{d} - \mathbf{p}_{k,\text{min}}) \rangle \cdot \mathbf{z}_{I}) \frac{\pi}{2}.$$ 

To avoid collisions, the navigating vehicle matches at minimum, the obstacle velocity component in the $\mathbf{p}_{k,\bot}^{1}$ direction, so that (27) is satisfied. The remaining velocity magnitude, defined in (28), is available to traverse the obstacle

$$\mathbf{p}_{d} \cdot \mathbf{p}_{k,\bot}^{1} \geq v_{k,\bot}^{1},$$

$$v_{k,\text{rem},1} = \sqrt{||\mathbf{p}_{d}||^{2} - v_{k,\bot}^{1}^{2}},$$

where $v_{k,\text{rem},1} > 0$ since Assumption 7 guarantees

$$||\mathbf{p}_{d}|| > ||\mathbf{p}_{k,s}^{1}|| \geq v_{k,\bot}^{1}.$$ 

Next, we define the desired velocity vector for the vehicle that matches the perpendicular component of the obstacle velocity and applies the remaining velocity to traverse the tangent direction as follows:

$$\mathbf{v}_{k,s}^{1} = v_{k,\bot}^{1} \mathbf{p}_{k,\bot}^{1} + v_{k,\text{rem},1} \mathbf{p}_{k,s}^{1}.$$ 

The corresponding course change to reach the desired velocity vector is

$$\Delta \phi_{k,s}^{1} = \text{angle}(\mathbf{p}_{d}, \mathbf{v}_{k,s}^{1}).$$

The circumnavigation direction corresponding to this tangent direction is defined by

$$z_{k,s}^{1} = -\text{sgn}(z_{k,s}^{1}) \mathbf{z}_{I},$$

where

$$z_{k,s}^{1} = ((\mathbf{p}_{k,\text{min}} - \mathbf{p}_{d}) \times \mathbf{p}_{k,s}^{1}) \cdot \mathbf{z}_{I}.$$ 

The circumnavigation direction is in the $+\mathbf{z}_{I}$ direction if the vehicle traverses counterclockwise around the obstacle and is in the $-\mathbf{z}_{I}$ direction otherwise. The circumnavigation direction in (32) and the candidate course change in (31) define the minimum absolute course angle, but this is not the only feasible course angle. The feasible course angles for each tangent direction $\mathbf{p}_{k,s}^{1}$ include any angle between $\Delta \phi_{k,s}^{1}$ and the angle to the opposite extent point $\Delta \phi_{k,s}^{2}$ as shown in Fig. 3. We define this generically as

$$\mathbf{O}_{k,s}^{\prime} = \left\{ \begin{array}{ll}
\{ \Delta \phi | \Delta \phi_{k,s}^{1} \leq \Delta \phi < \Delta \phi_{k,s}^{2} \}, & z_{k,s}^{1} = \mathbf{z}_{I} \\
\{ \Delta \phi | \Delta \phi_{k,s}^{1} < \Delta \phi \leq \Delta \phi_{k,sj}^{1} \}, & z_{k,s}^{1} = -\mathbf{z}_{I}
\end{array} \right\} \tag{34}$$

where

$$\Delta \phi_{k,sj}^{1} = \left\{ \begin{array}{ll}
\Delta \phi_{k,s4}, & z_{k,s}^{1} \Delta \phi_{k,s4} > 0, i = 1, 2, \\
\text{sgn}\left(\Delta \phi_{k,s4}^{1}\right)(2\pi - |\Delta \phi_{k,s4}^{1}|), & z_{k,s}^{1} \Delta \phi_{k,s4} < 0, i = 1, 2, \\
\text{sgn}\left(\Delta \phi_{k,s4}^{1}\right)(2\pi - |\Delta \phi_{k,s2}^{1}|), & z_{k,s}^{1} \Delta \phi_{k,s2}^{1} > 0, i = 3, 4, \\
\text{sgn}\left(\Delta \phi_{k,s2}^{1}\right)(2\pi - |\Delta \phi_{k,s2}^{1}|), & z_{k,s}^{1} \Delta \phi_{k,s2} < 0, i = 3, 4.
\end{array} \right\}$$

The set defined in (34) is further refined in (41) once a circumnavigation direction is chosen.

At this point, the vehicle has four candidate final velocity vectors defined in (30) and must choose among these four. The vehicle can traverse the obstacle toward $\mathbf{p}_{k,\text{e}1}$ or $\mathbf{p}_{k,\text{e}2}$, where the $\mathbf{v}_{k,s}^{1}$ and $\mathbf{v}_{k,s}^{2}$ candidates are associated with $\mathbf{p}_{k,\text{e}1}$, and the $\mathbf{v}_{k,s}^{1}$ and $\mathbf{v}_{k,s}^{2}$ candidates are associated with $\mathbf{p}_{k,\text{e}2}$.

The vehicle’s objective is to reach the goal position where the course change to the goal position is given by

$$\Delta \phi_{g} = \text{angle}(\mathbf{p}_{d}, \mathbf{p}_{g} - \mathbf{p}_{s}).$$

To determine if changing course to the goal position is feasible, the vehicle first considers if the obstacle’s circumnavigation direction has been established. If it has, the vehicle must choose the least extent point consistent with the established direction to keep the vehicle circling the obstacle in the same direction. The circumnavigation direction is established if the obstacle was previously the active obstacle, where the active obstacle is the closest obstacle that requires the vehicle to make a heading or velocity change. This is formally defined in Section III-F.

If the circumnavigation direction has not been established and $\Delta \phi_{g} \in \mathbf{O}_{k,s}^{\prime}$ for $i = 1, \ldots, 4$, then the vehicle has sufficiently traversed obstacle $k$ such that the goal position is feasible for one of the tangent directions. The extent point associated with this tangent direction is chosen and $\Delta \phi_{k} = \Delta \phi_{g}$.

If the circumnavigation direction has not been chosen and $\Delta \phi_{g} \notin \mathbf{O}_{k,s}^{\prime}$ for $i = 1, \ldots, 4$, then the vehicle uses
(36) and (37), based on current sensor information, to estimate how long it would take to traverse the obstacle toward the extent points

\[ t_{k,e1} = \frac{||\mathbf{p}_{k,e1} - \mathbf{p}_{k,min}||}{v_{k,rem,s1} - v_{k,s1}} \quad (36) \]

\[ t_{k,e2} = \frac{||\mathbf{p}_{k,e2} - \mathbf{p}_{k,min}||}{v_{k,rem,s3} - v_{k,s3}} \quad (37) \]

where \( t_{k,e1} \) corresponds with tangent directions \( \mathbf{p}_{k,s1} \) and \( \mathbf{p}_{k,s2} \), and \( t_{k,e2} \) corresponds with tangent directions \( \mathbf{p}_{k,s3} \) and \( \mathbf{p}_{k,s4} \). If the circumnavigating direction has not been established, then \( z_{k,t} = 0 \).

Now that \( \mathbf{p}_k.E \) and the circumnavigation direction have been established, two of the four candidate final velocity vectors have been eliminated. Next, the conservative or aggressive tangent direction must be selected. Since the vehicle cannot change course instantaneously, if the aggressive tangent direction solution is in the opposite direction as the circumnavigation direction (i.e., the vehicle maneuvering takes the vehicle closer to the obstacle before achieving the desired course change), then the aggressive tangent direction is not suitable because it will violate \( r_c \). If instead the tangent direction solution is in the same direction as the circumnavigation direction, then the aggressive tangent direction is suitable. It is summarized as follows:

\[
\mathbf{p}_k.E = \begin{cases} 
\mathbf{p}_{k,e1}, & z_{k,s1} = z_{k,t}, z_{k,t} \neq 0 \\
\mathbf{p}_{k,e2}, & z_{k,s3} = z_{k,t}, z_{k,t} \neq 0 \\
\mathbf{p}_{k,e1}, & \Delta \phi_g \in O'_{k,s1}, i = 1, 2 \\
\mathbf{p}_{k,e2}, & \Delta \phi_g \in O'_{k,s1}, i = 3, 4 \\
\mathbf{p}_{k,e1}, & t_{k,e1} = t_{k,e2}, \frac{||\Delta \phi_{s1} - \Delta \phi_g||}{||\Delta \phi_{s3} - \Delta \phi_g||} \leq 1 \\
\mathbf{p}_{k,e2}, & t_{k,e1} = t_{k,e2}, \frac{||\Delta \phi_{s1} - \Delta \phi_g||}{||\Delta \phi_{s3} - \Delta \phi_g||} > 1 \\
\mathbf{p}_{k,e1}, & t_{k,e1} < t_{k,e2} \\
\mathbf{p}_{k,e2}, & \text{otherwise}
\end{cases} \quad (38)
\]

where \( z_{k,t} \) is the established circumnavigation direction (32) from when obstacle \( k \) was the active obstacle. If the circumnavigation direction has not been established, then \( z_{k,t} = 0 \).

\[
S = \begin{cases} 
\{s2\}, & \text{sgn}(\text{angle} (\hat{\mathbf{p}}_d, \mathbf{v}_{k,s2})) = \text{sgn}(z_{k,s1}), \mathbf{p}_k.E = \mathbf{p}_{k,e1} \\
\{s1\}, & \text{sgn}(\text{angle} (\hat{\mathbf{p}}_d, \mathbf{v}_{k,s2})) \neq \text{sgn}(z_{k,s1}), \mathbf{p}_k.E = \mathbf{p}_{k,e1} \\
\{s4\}, & \text{sgn}(\text{angle} (\hat{\mathbf{p}}_d, \mathbf{v}_{k,s4})) = \text{sgn}(z_{k,s3}), \mathbf{p}_k.E = \mathbf{p}_{k,e2} \\
\{s3\}, & \text{sgn}(\text{angle} (\hat{\mathbf{p}}_d, \mathbf{v}_{k,s4})) \neq \text{sgn}(z_{k,s3}), \mathbf{p}_k.E = \mathbf{p}_{k,e2} 
\end{cases} \quad (39)
\]

Now that a tangent direction has been chosen, we update \( O'_{k,S} \) to \( O_k \) to take into account further restrictions if the vehicle is traversing a nonconvex obstacle. In this case, if the full obstacle is not within the sensor range, \( O'_{k,S} \) may not restrict the vehicle maneuvering enough, and the vehicle may incorrectly conclude that a course change to the goal position is feasible.

Instead, the vehicle stores the most constraining extent point, \( \mathbf{p}_{k.ep} \), which is the more restrictive of either the previous most constraining extent point \( \mathbf{p}_{k.ep}^- \) or the nonchosen extent point from most recent sensor information as follows:

\[
\mathbf{p}_{k.ep} = \begin{cases} 
\mathbf{p}_{k,rj}, & z_{k,t} = 0 \\
\mathbf{p}_{k,rj}, & z_{k,t} = \mathbf{z}_I, \Delta \phi_{k,S} > \Delta \phi_{k,ep} \\
\mathbf{p}_{k.ep}^-, & z_{k,t} = -\mathbf{z}_I, \Delta \phi_{k,S} < \Delta \phi_{k,ep} \\
\mathbf{p}_{k.ep}^-, & z_{k,t} = \mathbf{z}_I, \Delta \phi_{k,.ep} \geq \Delta \phi_{k,S} \\
\mathbf{p}_{k.ep}^-, & z_{k,t} = -\mathbf{z}_I, \Delta \phi_{k.ep} \leq \Delta \phi_{k,S}
\end{cases} \quad (40)
\]

where

\[
\Delta \phi_{k.ep} = \text{angle} (\hat{\mathbf{p}}_d, \mathbf{p}_{k.ep}^* - \hat{\mathbf{p}}_d) \\
\mathbf{p}_{k.ep}^* = \begin{cases} 
\mathbf{p}_{k.ep}, & j = 1, S = s3, s4 \\
\mathbf{p}_{k.ep}, & j = 2, S = s1, s2 \quad S' = \{s4, S = s1, s2\} 
\end{cases} \quad (41)
\]

and \( \mathbf{p}_{k.ep}^- \) is calculated from (12).

The heading change associated with the most constraining extent point is compared to the feasible course change angles in \( O_k.S \) according to

\[
O_k = \begin{cases} 
\{\Delta \phi|\max(\Delta \phi_{k.ep}, \min(O'_{k,S})) \leq \Delta \phi < \Delta \phi_{k,S}\}, \\
\{\Delta \phi|\Delta \phi_{k,S} < \Delta \phi \leq \min(\Delta \phi_{k,.ep}, \max(O'_{k,S}))\}, \\
z_{k,S} = -\mathbf{z}_I.
\end{cases} \quad (42)
\]

The final step is to define the course change angle for obstacle \( k \) as

\[
\Delta \phi_k = \begin{cases} 
\Delta \phi_g, & \Delta \phi_g \in O_k \\
\Delta \phi_{k,S}, & \text{otherwise}
\end{cases} \quad (42)
\]

The course change angle and set of all feasible course angles for obstacle \( k \) are used to compare with other obstacles and vehicles to determine a final course change in Section III-F. Fig. 4 shows this process for a circular obstacle.

E. Course Change Definition for a Vehicle

The process to determine a course change angle and the set of all feasible course angles for a vehicle is very similar to the process described in Section III-D for an obstacle. We distinguish critical and noncritical vehicles and also if there are critical obstacles present as there are some slight differences in the calculations.

For noncritical vehicles with critical, noncritical, or no obstacles, and critical vehicles with no critical obstacles, the process is the same as Section III-D, except for three modifications: 1) the vehicle uses \( r_{c,k}^* \) instead of \( r_c \); 2) the vehicle uses (44)–(47) to define the tangent directions; and 3) since the vehicles are convex, \( O_{k,s,i} = O'_{k,s,i} \) for \( i = 1, \ldots, 4 \). Using these modifications, the course change angle \( \Delta \phi_k \) is solved from (42),
and the set of all feasible course changes $O_k$ is solved from (34) for $S$ defined by (39).

For critical vehicles with critical obstacles present, we include two additional modifications: 1) the vehicle does not fix its circumnavigation direction since the other vehicle can maneuver differently and 2) because the circumnavigation is not fixed, it retains the desired course angles and the set of all feasible course angles for both extent points so there are two possible circumnavigation directions. This section describes the process to determine the course angles, $\Delta \phi_k, z_i$ and $\Delta \phi_k, -z_i$, and the corresponding sets of all feasible course angles, $O_{k, z_i}$ and $O_{k, -z_i}$, associated with each circumnavigation direction, $z_i$ and $-z_i$, for a critical vehicle $k$.

Similar to obstacles, the vehicle first determines the bounding extents and tangent directions, $p_{k \min}$ and $p_{k \max}$, for each extent point so there are two possible circumnavigation directions as defined in (23)–(33).

The course change angles, $\Delta \phi_{k, z_i}$ and $\Delta \phi_{k, -z_i}$, and set of all feasible course angles, $O_{k, z_i}$ and $O_{k, -z_i}$, for vehicle $k$ are used to compare to other obstacles and vehicles to determine a final course change in Section III-F.

F. Course Change Definition for Multiple Obstacles/Vehicles

To determine the overall course change to safely navigate in the environment, the vehicle uses the course change angles at minimum for the critical obstacles and vehicles.
(\(k \in I_{\mathcal{co}} \cup I_{\mathcal{cv}}\)) and may also consider noncritical obstacles. This section describes the process for evaluating and combining the course change angles and sets of all feasible course angles for each obstacle/vehicle to determine a final course change \(\Delta \phi\) and identify the active obstacle.

Starting with the critical obstacles, we take the course change definitions \(\Delta \phi_k\) from (42) and the feasible sets of course angles for obstacles \(O_k\) from (41) and combine them for all obstacles \(k \in I_{\mathcal{co}}\) to form the set of candidate course changes \(O_{\Delta \phi, o}\) and the feasible set of all course changes \(O_o\), as defined in the following:

\[
O_{\Delta \phi, o} = \{ \Delta \phi_k \mid k \in I_{\mathcal{co}} \} \tag{48}
\]

\[
O_o = \left\{ \begin{array}{ll}
\bigcap_{m=1}^{j} O_{k_m}, & \bigcap_{m=1}^{j} O_{k_m} \neq \emptyset, \\
\bigcap_{k \in I_{\mathcal{co}}} O_k, & \text{otherwise}
\end{array} \right. \tag{49}
\]

where the obstacles are evaluated from most to least critical where the most critical obstacle minimizes \((4)\). The \(k_{j+1}\) obstacle is the first obstacle that produces an empty set of feasible course changes when \(O_{k_j}\) is intersected with all previous sets. By constantly navigating around the closest \(j\) obstacles, the vehicle will eventually clear all obstacles as shown in Theorems 2 and 3.

Similarly, if there are only critical vehicles (no critical obstacles), then we get \(O_{\Delta \phi, v}\) and \(O_o\) from (48) and (49), respectively, for \(k \in I_{\mathcal{cv}}\).

Recall that when there are both critical obstacles and vehicles we retain both circumnavigation directions, so the process is a little different to simplify the course changes and feasible sets. We start by bounding the course changes for each circumnavigation direction as

\[
\Delta \phi_{v,z} = \min_{k \in I_{\mathcal{cv}}} (\Delta \phi_{k,z}) \tag{50}
\]

\[
\Delta \phi_{v,-z} = \max_{k \in I_{\mathcal{cv}}} (\Delta \phi_{k,-z}) \tag{51}
\]

While the bounding angles provide a course change that navigates around all vehicles, the goal position may still be feasible depending on the vehicle locations. To determine if the goal position is feasible, we define sets of all feasible course change angles for each circumnavigation direction as

\[
O_{\phi,v, \pm z} = \left\{ \begin{array}{ll}
\bigcap_{m=1}^{j} O_{k_m, \pm z}, & \bigcap_{m=1}^{j} O_{k_m, \pm z} \neq \emptyset, \\
\bigcap_{k \in I_{\mathcal{cv}}} O_{\phi,v, \pm z}, & \text{otherwise}
\end{array} \right. \tag{52}
\]

We use the sets for each circumnavigation direction and the bounding course change angles to define the set of vehicle course change angles as

\[
O_{\Delta \phi, v} = \left\{ \begin{array}{ll}
\{ \Delta \phi_g \}, & \Delta \phi_g \in O_{v,z} \cap O_{v,-z}, \\
\{ \Delta \phi_g, \Delta \phi_{v,-z} \}, & \Delta \phi_g \in O_{v,z}, \Delta \phi_g \notin O_{v,-z}, \\
\{ \Delta \phi_g, \Delta \phi_{v,z} \}, & \Delta \phi_g \in O_{v,-z}, \Delta \phi_g \notin O_{v,z}, \\
\{ \Delta \phi_{v,z}, \Delta \phi_{v,-z} \}, & \text{otherwise}
\end{array} \right. \tag{53}
\]

Now that the critical obstacles and vehicles have been combined independently, a final course change must be determined. As the vehicle evaluates the candidate course changes and sets of all feasible course changes, it does so according to the following principles. The main objective is to reach the goal position, so the ability to navigate toward the goal position is checked first, as given by condition 1 in (54).

Next, we consider three cases where navigation to the goal position is not feasible and there are critical obstacles and/or critical vehicles. If there are critical obstacles and vehicles, and there is no viable course change, which occurs when \(O_{\Delta \phi, v} \cap O_o = \emptyset\), then reprioritization is necessary. In this case, the vehicle chooses a course change in \(O_o\) that is closest to \(\Delta \phi_g\) and assigns itself a higher priority (relative to the other vehicles in \(I_{\mathcal{cv}}\)) until it clears \(r^{e,k}\). This is condition 2 in (54).

If there are critical obstacles and vehicles but \(O_{\Delta \phi, v} \cap O_o \neq \emptyset\), then the vehicle chooses the desired course change from \(O_{\Delta \phi, v}\) that is in \(O_o\). This is condition 3 of (54).

If there are only critical obstacles, then the vehicle chooses the desired course change from \(O_{\Delta \phi, o}\) that is in \(O_o\) or the feasible course change from \(O_o\) that is closest to one of the desired changes in \(O_{\Delta \phi, o}\). This is condition 4 of (54).

If there are only critical vehicles, then the choice is the same as the case where there are only critical obstacles, except we use \(O_{\Delta \phi, v}\) and \(O_o\). This is condition 5 of (54).

In the event that all the critical obstacles allow the goal position or there are no critical obstacles, then the noncritical obstacles are evaluated in increasing order starting with the most imminent [i.e., the one that minimizes (4)]. The first noncritical obstacle/vehicle \(k_j\) where \(\Delta \phi_g \notin \bigcup_{m=1}^{j} O_{k_m}\) for \(k_i \in I_{\mathcal{mo}} \cup I_{\mathcal{cv}}\) defines the desired course change \(\Delta \phi_{k_j}\). The desired course change must respect the sets of all feasible course angles for all preceding obstacles/vehicles. This is condition 6 in (54).

If none of the obstacles/vehicles prohibit navigation to the goal position, then \(\Delta \phi = \Delta \phi_{k_j}\) which is condition 7 of (54). These seven conditions, summarized in the following, are evaluated in sequence until a condition is met:

\[
\Delta \phi = \left\{ \begin{array}{ll}
\Delta \phi_g, & \|p_g - p_d\| < \|p_{\min} - p_d\| \\
\arg\min_{\Delta \phi \in O_o} (|\Delta \phi - \Delta \phi_g|), & O_{c,z} = 0, I_{\mathcal{co}} \neq 0, I_{\mathcal{cv}} \neq 0 \\
\arg\min_{\Delta \phi \in O_o} (|\Delta \phi - \Delta \phi_{v}|), & \Delta \phi_g \notin O_{c,z}, I_{\mathcal{co}} \neq 0, I_{\mathcal{cv}} \neq 0 \\
\arg\min_{\Delta \phi \in O_o} (|\Delta \phi - \Delta \phi_{v}|), & \Delta \phi_g \notin O_v, I_{\mathcal{co}} \neq 0, I_{\mathcal{cv}} = 0 \\
\arg\min_{\Delta \phi \in O_o} (|\Delta \phi - \Delta \phi_{v}|), & \Delta \phi_g \notin O_v, I_{\mathcal{co}} = 0, I_{\mathcal{cv}} \neq 0 \\
\arg\min_{\Delta \phi \in O_o} (|\Delta \phi - \Delta \phi_{v}|), & \Delta \phi_g \notin O_{c,v} \\
\arg\min_{\Delta \phi \in O_o} (|\Delta \phi - \Delta \phi_{v}|), & \Delta \phi_g \notin O_v, I_{\mathcal{co}} = 0, I_{\mathcal{cv}} = 0 \\
\Delta \phi_g, & \Delta \phi_g \in O_{c,v} \\
\Delta \phi_g, & \Delta \phi_g \in O_{c,e} \\
\Delta \phi_g, & \Delta \phi_g \in O_{e,e} \tag{54}
\end{array} \right.
\]
where

\[
O_{c,e} = \begin{cases}
O_o \cap \bigcap_{i=1}^j O_{k_i}, & I_{co} \neq \emptyset, I_{cv} = \emptyset \\
O_v \cap \bigcap_{i=1}^j O_{k_i}, & I_{co} = \emptyset, I_{cv} \neq \emptyset \\
O_{c,e} \cap \bigcap_{i=1}^j O_{k_i}, & I_{co} \neq \emptyset, I_{cv} \neq \emptyset \\
\{O_i\}, & I_{co} = \emptyset, I_{cv} = \emptyset 
\end{cases}
\]

(55)

where \( z \) is either \( \pm z_{f} \) depending on which circumnavigation direction has been chosen, \( O_{c,e} \cap \bigcap_{i=1}^j O_{k_i} \) for \( z = \pm z_{f} \), and \( O_{c,e} \) is the same as \( O_{c,e} \), but evaluated up to noncritical obstacle \( k_{j-1} \) (i.e., the last noncritical obstacle/vehicle that still allows navigation to the goal position).

The active obstacle is the obstacle \( k_{a} \) that minimizes (4) and satisfies \( \phi_k \neq \emptyset \). The circumnavigation direction is only set for obstacles, and remains fixed until \( \|\dot{p}_{k_{a}}\| - \|\dot{p}_{d}\| > r_{s} \).

The course change defined by (54) is used to generate smooth trajectories in Section III-H.

G. Velocity Change Definition

The vehicle can also make velocity changes to avoid collisions and safely navigate the environment. This section examines the cases where velocity change is appropriate and the process to determine the velocity change \( \Delta \dot{v} \).

The only adjustments in velocity are in the three cases described in this section: 1) slowing to reach the goal position; 2) when the vehicle cannot safely pass another vehicle (due to the presence of obstacles); and 3) when the vehicle has come within \( r_{c,k} \) of another vehicle. We briefly examine these cases.

First, when there is a clear path to the goal position and the vehicle is within some critical distance, \( r_{goal} \), of it, the vehicle computes a trajectory to come to a stop at the goal position (or in the case of a fixed wing aircraft, to come to a loiter). The critical distance may be the minimum stopping distance, or a user-defined distance that is greater than the minimum stopping distance. This is the first condition in (56).

Second, if a vehicle detects another vehicle within its sensor range and there are also obstacles within sensor range of one or both vehicles, the vehicle slows to match the velocity of the slowest vehicle within sensor range, or connected to a vehicle within sensor range. Since the obstacle spacing \( r_{obs} \geq 2r_{c} + r_{180} \) (from Assumption 7), there is no guarantee that multiple vehicles can fit between obstacles, or that the more capable vehicle is able to safely complete a passing maneuver. The safe solution is then to match velocity so that all vehicles can maneuver safely. This is the second condition in (56).

Finally, we consider the case where the vehicle has already matched the velocity of a slower vehicle, but the slower vehicle has maneuvered so that \( r_{c,k} \) is violated. The maneuvering vehicle can either change course (as already established in Section III-F) or it can slow down temporarily. To determine if maintaining course and decreasing velocity is appropriate, we use \( v_{k,s} \), which is computed in (30) for the conservative tangent direction. If the course change associated with \( v_{k,s} \) and a decrease in velocity with \( \Delta \dot{v} = 0 \), both produce relative velocity vectors in the same direction (e.g., \( v_{rel} = -\dot{p}_{d,k} + v_{k,s} \)), then the vehicle decreases velocity according to condition 3 of (56) instead of performing a lengthy course change maneuver. Since the vehicle is slowing down (instead of maneuvering), it considers course changes for the next closest obstacle [as defined in Section III-F, (54)].

Once the vehicle has reestablished a distance \( > r_{s,k}^{*} \) from the other vehicle, then it resumes its previous velocity. This is the final condition in (56). These four conditions, summarized in the following, are evaluated in sequence until a condition is met:

\[
\Delta \dot{v} = \begin{cases}
-\|\dot{p}_{d}\|, & \|\dot{p}_{g} - \dot{p}_{d}\| \leq r_{goal} \\
\min \Delta \dot{v}_{k}, & I_{obs,k} \neq \emptyset, I_{obs} \neq \emptyset \\
(K_{\Delta v} - 1)\|\dot{p}_{d}\|, & r_{k} < r_{c,k}^{*}, \min \|\dot{p}_{e}\| \leq K_{\Delta v} < 1, 0 \in O_o \\
\dot{v}_{c} - \|\dot{p}_{d}\|, & \text{otherwise}
\end{cases}
\]

(56)

where \( \Delta \dot{v}_{k} = \|\dot{p}_{f}\| - \|\dot{p}_{d}\| \), \( \|\dot{p}_{g} - \dot{p}_{d}\| > r_{goal} \), \( r_{k} = \|\dot{p}_{k} - \dot{p}_{d}\| \), and \( K_{\Delta v} \) is a component of the solution to \( \dot{p}_{d}, -v_{rel}^{n}K = \dot{p}_{d,k} \), where \( K = [K_{\Delta v}, K_{\phi_{n}}]^{T} \), and \( K_{\Delta v} \) and \( K_{\phi_{n}} \) are scaling constants for the \( \dot{p}_{d} \) and \( v_{rel} \) vectors, respectively, so that \( -\dot{p}_{d,k} + K_{\Delta v} \dot{p}_{d} = K_{\phi_{n}}v_{rel}^{n} \) is satisfied. If the scaling constant \( K_{\Delta v} \) causes the vehicle to go below its minimum velocity \( v_{min} \) (e.g., a fixed wing aircraft), then the vehicle maintains velocity and makes a course change instead. The velocity change defined by (56) is used to generate smooth trajectories in Section III-H.

H. Smooth Course and Velocity Transitions

The trajectory generation algorithm utilizes sigmoid functions to transition from the previous course \( \phi_{n-1} \) and velocity \( v_{n-1} \) to a new course \( \phi_{n} \) and velocity \( v_{n} \) as determined by the course and velocity changes from (54) and (56). This section describes the process to make the course and velocity transitions by defining the desired trajectory, \( \dot{p}_{d}(t) \) and \( \dot{p}_{d}(t) \), for \( t_{o,n} \leq t \leq t_{o,n} + \tau_{f,n} \), where \( t_{o,n} \) is the start of the nth sigmoid curve and \( \tau_{f,n} \) is the nth sigmoid curve time span.

The hyperbolic tangent function \( \tan h \) is chosen because of its widespread use in generating smooth motion transitions [29]. We define the course and velocity functions, and their first derivatives as

\[
\phi = c_{1} \tanh(c_{2} \tau - c_{3}) + c_{4}
\]

(57)

\[
\dot{\phi} = c_{1}c_{2} \left( 1 - \tanh^{2}(c_{2} \tau - c_{3}) \right)
\]

(58)

\[
v = d_{1} \tanh(d_{2} \tau - d_{3}) + d_{4}
\]

(59)

\[
\dot{v} = d_{1}d_{2} \left( 1 - \tanh^{2}(d_{2} \tau - d_{3}) \right)
\]

(60)

where \( c_{i} \) and \( d_{i} \) are coefficients to be determined and \( \tau_{n} = t - t_{o,n} \) is the sigmoid curve time, where the sigmoid curve start time, \( t_{o,n} \), is defined in (68). The desired velocity vector is

\[
\dot{p}_{d} = \begin{bmatrix}
v \cos \phi \\
v \sin \phi
\end{bmatrix}
\]

(61)

The coefficients are solved analytically by considering the following assumptions: 1) each sigmoid function occurs over the time interval \( \tau = 0 \) to \( \tau = \tau_{f} \) and 2) since \( \tanh \) asymptot-
Fig. 6. Example of summed sigmoid curves. At $t_{o,n}$, the vehicle determines a course change of $\Delta \phi_{n-1}$ (blue curve) and starts the sigmoid at time $t_{o,n-1}$. With new sensor information, an additional course change of $\Delta \phi_n$ (red curve) is necessary. This curve cannot start immediately because it would violate the maximum thrust. Instead, the vehicle computes $t_{o,n}$ from (68), which is after the $n-1$ curve peak acceleration at $K_{o,n-1} \tau_{f,n-1}$. The two curves are summed to produce a smooth course change (green curve) that does not violate $f_{\text{max}}$.

Physically approaches $-1$ and $1$, the bounds of the function are approximated by $\pm (1 - \varepsilon)$, (where we use $\varepsilon = 10^{-3}$ to reduce the error of this approximation to $<1\%$). The coefficient solutions are written as

$$
c_3 = d_3 = \tanh^{-1}(1 - \varepsilon) = 3.8 \tag{62}
$$

$$
c_2 = d_2 = 2c_3 / \tau_f = 7.6 / \tau_f \tag{63}
$$

$$
c_1 = c_4 = 0.5 \Delta \phi \tag{64}
$$

$$
d_1 = d_4 = 0.5 \Delta v \tag{65}
$$

where the sigmoid curve time span $\tau_f$ is defined later in Theorem 1 to respect the vehicle thrust limitations.

As the vehicle navigates the environment, it receives new sensor information every $\Delta T_s$. If the sensor update rate is very fast, the vehicle likely does not complete the desired course and velocity changes before new sensor data is available. As a result, the vehicle must wait until there is available thrust and then compute course and velocity changes from the most recent sensor data.

The maximum delay in starting the next maneuver in the cruise velocity constraints is defined in Section IV-B. This delay allows subsequent maneuvers to begin before previous ones are finished without violating the maximum thrust constraint.

To make use of the new sensor information, the vehicle sums successive sigmoid curves as shown in Fig. 6. Since the sigmoid function and its first four derivatives asymptotically approach 0 (effectively are 0) at $\tau = 0$ and $\tau = \tau_f$, the summed sigmoid curve provides the smoothness guarantee of Property 1.

The sigmoid curves cannot be summed arbitrarily without violating the vehicle’s maximum thrust, $f_{\text{max}}$. Instead, we scale and shift subsequent maneuvers such that the summation of the maneuvers (i.e., sigmoid curves) is bounded to respect $f_{\text{max}}$. The curve scaling is achieved by varying the sigmoid curve time span $\tau_f$ and we introduce an offset time $t_{o,n}$ to shift the curve start. The summed sigmoid functions are defined as

$$
\phi(t) = \sum_{i=1}^{n} (c_{1,i} \tanh(c_{2,i}(t_n - t_{o,i}) - c_{3,i}) + c_{4,i}) \tag{66}
$$

$$
v(t) = \sum_{i=1}^{n} (d_{1,i} \tanh(d_{2,i}(t_n - t_{o,i}) - d_{3,i}) + d_{4,i}) \tag{67}
$$

where

$$
t_{o,n} = \begin{cases} 
  t_n, & n = 1 \\
  \max(t_n + \Delta T_s, t_{o,n-1} + t_{\text{int},n}), & n > 1 
\end{cases} \tag{68}
$$

where $t_n \geq 0$ is the current time and $t_{\text{int},n}$ is the point on the previous sigmoid curve where thrust becomes available for the next maneuver.

The definition for $t_{\text{int},n}$ is the intersection point between the sigmoid acceleration curve ($\dot{\phi}_d(t)$), and a linear approximation of the sigmoid curve acceleration as shown in Fig. 7(a) and defined as

$$
K_{n-1} \tau_{f,n-1} < t_{\text{int},n} < \tau_{f,n-1} \tag{69}
$$

$$
-a_{\text{max}} - \frac{1}{1 - K_{n-1}} t_{\text{int},n} \tag{70}
$$

We use the linear approximation because it provides simple upper bounds on the sum of two successive $||\dot{\phi}_d(t)||$ curves when the second curve starts at $t_{o,n}$ (as proven in [28, Appendix A]).

The linear approximation slope for the rising $h_n^+$ and falling $h_n^-$ sides of the sigmoid curve acceleration are defined by

$$
h_n^+ = a_{\text{max}} / K_n \tau_{f,n} \tag{71}
$$

$$
h_n^- = a_{\text{max}} \frac{1}{(1 - K_n) \tau_{f,n}} \tag{72}
$$
where \( a_{\text{max}} \) is the remaining acceleration available for tracking the trajectory (after lift and drag forces have been accounted for) and defined in Theorem 1 by (86), and \( K_n \tau_{f,n} \) is the sigmoid curve time where the acceleration is maximum as defined by

\[
\tau_{n, \text{max}} = K_n \tau_{f,n}, \quad \text{where} \quad 0 \leq K_n \leq 1
\]  
(73)

\[
K_n = \frac{1}{2} \left( \frac{\tanh^{-1}(H_n)}{c_3} + 1 \right)
\]  
(74)

\[
H_n = \tanh(c_2 n \tau_{n, \text{max}} - c_3 n)
\]  
(75)

The computation of \( H_n \) is described in Theorem 1 and its proof. We use the linear approximation slope terms when solving for \( \tau_{f,n} \) and \( t_{o,n} \) to match slopes and respect \( a_{\text{max}} \) as shown in Fig. 7(b).

Once the sigmoid curve time span, \( \tau_{f,n} \), is solved according to Theorem 1, the trajectory is fully defined by (61) which uses the definitions for \( \phi(t) \) and \( v(t) \) from (66) and (67). The trajectory is defined over \( t_{o,n} \leq t \leq t_{o,n} + \tau_{f,n} \) where \( t_{o,n} \) is defined by (68). A vehicle controller, such as the one described in Section V, follows the trajectory to navigate the vehicle.

IV. TRAJECTORY GUARANTEES

To guarantee that the vehicle can navigate safely in the environment, we present several theorems that define the sigmoid curve time bound, the maximum velocity, guarantee that the vehicle clears obstacles and other vehicles by \( r_c \), and guarantee that the vehicle reaches the goal position in finite time. The proofs for each of these theorems are given as Appendices A–C, respectively.

To aid in the theorems and proofs, we define several quantities. First, the maximum available planar force \( f_{p, \text{max}} \) and drag force \( f_w \) are defined by

\[
f_{p, \text{max}} = \sqrt{f_{\text{max}} - (mg)^2}
\]  
(76)

\[
f_w = K_d \|v_w\|^2 (-x_W)
\]  
(77)

\[
K_d = \frac{1}{2} \rho C_D A_{xW}
\]  
(78)

where \( m \) is the vehicle mass, \( g \) is gravity, \( v_w = \dot{\tilde{h}} - v_{\text{air}} \) is the relative wind velocity between the vehicle and the air, \( x_W \) is the wind frame axis aligned with \( v_w \), \( \rho \) is the air density, \( C_D \) is the coefficient of drag, and \( A_{xW} \) is the cross-sectional area normal to \( v_w \).

Additionally, we define the planar force vector as the sum of the trajectory and drag forces

\[
f_{\text{planar}} = m\ddot{\tilde{h}} + f_w
\]  
(79)

where the maximum planar force magnitude \( f_{p, \text{max}} \) occurs when the vectors are aligned. Considering the two components independently, the maximum drag is \( K_d v_{w, \text{max}}^2 \) where \( v_{w, \text{max}} \) is the maximum wind speed defined in the theorems, and the desired acceleration magnitude is obtained from (61) as

\[
||\ddot{\tilde{h}}|| = \sqrt{\dot{v}^2 \dot{\phi}^2 + \dot{\phi}^2}
\]  
(80)

We can further manipulate (80) to define its maximum value by substituting the definitions for \( K \) and \( H \) from (74) and (75) into the sigmoid functions from (57) and (59). If we also utilize (63), we can isolate the dependency on \( \tau_f \) as

\[
||\ddot{\tilde{h}}||^2 = \left( \frac{2c_3}{\tau_f} \right)^2 S_{\text{traj}}
\]  
(81)

where

\[
S_{\text{traj}} = (c_1 (d_1 H + d_2 (1 - H^2))^2 + (d_1 (1 - H^2))^2
\]  
(82)

and the solution for \( H \) is derived in [28, Appendix A]. Each of these relationships is referenced in the following theorems.

A. Sigmoid Curve Time span

The sigmoid curve time span is bound by Theorem 1 to ensure that the sigmoid curve does not violate \( f_{\text{max}} \). Theorem 1 also defines the offset time for the sigmoid curve start.

Theorem 1: Let the sigmoid curve time span \( \tau_f \) for the \( n \)th sigmoid be defined as

\[
\begin{cases}
\tau_{f,\text{min},n}, & n = 1 \\
\tau_{f,n} & n > 1 \\
\tau_{f,\text{min},n}, & n = 1
\end{cases}
\]  
(83)

\[
\tau_{f,n} = \max\left( t_{o,i} + \tau_{f,i} \right)
\]  
(84)

\[
h_n = \begin{cases}
\tau_{f,\text{min},n}, & n = 1 \\
h_{n-1} & n > 1
\end{cases}
\]  
(85)

\[
\begin{align*}
\rho C_D A_{xW} & = m \tau_{f,\text{min},n} \tau_{f,n} \\
& = 1/m \left( f_{p, \text{max}} - K_d v_{w, \text{max}}^2 \right)
\end{align*}
\]  
(86)

\[
v_{w, \text{max}} = \max(v_{n-1}, v_{n-1} + \Delta v_n) + v_{\text{air}}, \quad v_{w, \text{max}} \text{ is the desired final velocity at } t_{n-1}, \quad K_{\text{min}} = \min(K_0, 1 - K_n), \quad H_n \text{ is the real solution to}
\]

\[
-3c_1^2 c_2^2 \tau_f^2 H_n^3 - 5c_1^2 d_1 d_2 H_n^2 + (-2c_1^2 c_2^2 + d_1 c_2 - 2d_2) H_n + d_2 d_1 c_2^2 = 0
\]  
(87)

that satisfies \( |H_n| < 1 - \varepsilon \). Then, if \( \tau_{f,n} \) is always chosen to satisfy (83), the vehicle trajectory does not violate \( f_{\text{max}} \) in the presence of a bounded wind disturbance velocity, \( v_{\text{air, max}} \), that satisfies \( v_{\text{air, max}} < \sqrt{f_{\text{planar}}/K_d} \).

Proof: See [28, Appendix A].

B. Cruise Velocity Bound

Theorem 2 derives an upper bound on the vehicle’s cruise velocity based on its maximum thrust, sensor update rate and range, and obstacle spacing. These criteria result in three inequalities that must be satisfied for the cruise velocity. Since
each inequality provides an upper bound on the velocity, the minimum is chosen.

**Theorem 2**: Let the vehicle's maximum cruise velocity be defined as

$$v_c \leq \min(v_c, v_{c,s}, v_{c,o})$$  \hspace{1cm} (88)

where $v_{c,f}$ is the minimum real, positive solution of

$$\left(\frac{m}{r_{\min}} + K_d\right) v_{c,f}^2 + 2K_d v_{\text{air,max}} v_{c,f} + K_d v_{\text{air,max}}^2 - f_{\text{planar}} = 0$$

$v_{c,s}$ satisfies the following inequalities:

$$v_{c,s} \leq \frac{r_s - (t_{d,max} + \tau_{f,s})v_{o,max} - r_c}{t_{d,max} + \int_{0}^{\tau_{f,s}} \sin \phi(t)dt}$$ \hspace{1cm} (90)

$$\tau_{f,s} \geq \frac{c_3}{a_{\max}} \left(\tan^{-1}\left(\frac{v_{o,max}}{v_{c,s}^2 - a_{\max}^2}\right) + \frac{\pi}{2}\right) v_{c,s}$$ \hspace{1cm} (91)

where

$$t_{d,max} = \begin{cases} 0.54c_3 \pi v_{c,s} + \Delta T_c, & 0.54\tau_{f,s} \geq 2\Delta T_c \\ 2\Delta T_s + \Delta T_c, & \text{otherwise} \end{cases}$$ \hspace{1cm} (92)

and $v_{o,max}$ is the maximum expected obstacle velocity in the environment (or $v_{o,max} = v_{c,s}$ if the maximum is unknown, producing $\tau_{f,s} \geq \frac{c_3}{a_{\max}} \pi v_{c,s}$), $\Delta T_c$ is the maximum on-board algorithm computation time, and $v_{c,o}$ satisfies the following inequalities:

$$v_{c,o} \leq \frac{r_{\text{obs}} - 2r_c}{t_{d,max} + \int_{0}^{\tau_{f,c}} \sin \phi(t)dt}$$ \hspace{1cm} (93)

$$\tau_{f,o} \geq \frac{c_3}{a_{\max}} \frac{1}{2} \pi v_{c,o}$$ \hspace{1cm} (94)

where $r_{\text{obs}}$ is the minimum distance between two obstacles. If the vehicle's maximum cruise velocity satisfies (88), then the vehicle does not violate $f_{\text{max}}$ when making a turn of radius $r_t \geq r_{\min}$ in the presence of a bounded wind velocity disturbance that satisfies $v_{\text{air,max}} < \sqrt{f_{\text{planar}}/K_d}$ and safely clears obstacles by $r_c$.

**Proof**: See Appendix B of [28].

C. Goal Position Convergence and Clearance Radius Guarantee

The convergence to the goal position and clearance radius guarantee are defined in Theorem 3 to ensure that the vehicle reaches the goal position in finite time, avoids collisions, and clears all obstacles and vehicles by $r_c$.

**Theorem 3**: Let the course change, velocity change, and circumnavigation direction be defined by (54), (56), and (32), respectively. If the vehicle generates a trajectory from these equations, and the maximum cruise velocity satisfies Theorem 2, then the vehicle reaches the goal position (provided $||\hat{p}_g|| = 0$) in finite time and clears all obstacles by $r_c$.

**Proof**: See Appendix C of [28].
various moving obstacles to a goal position and 2) multiple vehicles navigate into two entrances of a building to reach goal positions inside the building. We ran the simulations for multiple random initial conditions, and showed an example result from each scenario. For all simulations, we use (83) in Theorem 1 and (88)–(91), (93), and (94) in Theorem 2 as equality constraints. The other simulation parameters and results are discussed in the following sections.

We introduce a smoothing method to reduce the vehicle switching between the conservative and aggressive tangent directions. This variable $r_{CA}$ defined in (101), is similar to $r_c$ from (2) to account for the distance the vehicle travels when making a turn. This distance allows the vehicle to safely use the aggressive tangent direction until it is within $r_{CA}$ of the obstacle/vehicle. Once within $r_{CA}$, it has sufficient clearance to maneuver and reverts back to the criteria for conservative and aggressive tangent directions from (39)

$$r_{CA} = r_c + r_{180} + \Delta T_{tot} v_{\perp,CA}$$  \hspace{1cm} (101)

$$v_{\perp,CA} = \dot{p}_k \cdot \frac{p_d - p_{min}}{||p_d - p_{min}||}$$ \hspace{1cm} (102)

where $r_{180}$ and $\Delta T_{tot}$ are defined in (2).

A. Simulation 1

The first simulation shows a vehicle navigating around moving obstacles to a goal position. The environment has a bounded mean wind disturbance of 3 m/s and there is also gusting. The wind model uses the Von Kármán power spectral density function over a finite frequency range, and then applies that model to the method described in [36] to create a spatial wind field. The gusting profile is defined in the military specification MIL-F-8785C [37] as a “$1 - \cos^n$” model. Fig. 8(a) shows the wind experienced by the vehicle in the simulation.

The vehicle parameters for the simulation are: $m = 0.54$ g, $f_{max} = 10.17$ N, $r_c = 2$ m, $r_{obs} = 7$ m, $r_s = 10$ m, $\Delta T_s = 1$ s, $\Delta T_e = 0.1$ s, $J = \text{diag}([0.0017, 0.0017, 0.0031])$ kg/m², $C_d = 1.6$, and $A_{xv} = [0.20, 0.20, 0.45]$ m². The maximum cruise velocity is solved from Theorem 2 as $v_c = 0.89$ m/s. The controller gains are $\alpha_1 = 1$, $\alpha_2 = 3$, $k_x = 9$, and $\beta = 0.25$, where all of these values satisfy the constraints outlined in [26] except $\beta$, which produces nonsmooth behavior for large values.

There are four moving obstacles in the simulation with velocity magnitudes ranging from 0.375 to 0.75 m/s and constant courses as shown in Fig. 9(a). Note that all obstacle velocities are less than the vehicle cruise velocity. Snapshots of the vehicle maneuvering through the environment are shown in Fig. 9, the
course changes are shown in Fig. 10, and the thrust required is shown in Fig. 11. The vehicle clears all four obstacles by \( \geq r_c \), travels at \( v_c \) until the goal position, and reaches the goal position in just under 90 s.

**B. Simulation 2**

The second simulation shows five vehicles navigating into a building and around stationary obstacles to different goal positions. We use temporary goal positions to guide the vehicles inside the building; once the temporary goal positions are reached, the final goal positions are used. There is a bounded mean disturbance of 2 m/s when the vehicles are outside the building, a small transition zone where the wind enters the building, and no wind once the vehicles are fully inside. The transition zone is based on the results of a simulation of the building environment using SolidWorks 2016 Flow Simulation package. Fig. 8(b) shows the wind experienced by vehicle 5 in the simulation.

The vehicle parameters that differ for the five vehicles are summarized in Table I. The vehicles are physically the same but...
differ in maximum thrust and clearance radii to represent vehicles carrying different payloads for different missions. The other parameters are the same as in simulation 1, except \( r_{\text{obs}} = 2 \) m.

Different cruise velocities result in different sets of \( I_{\text{mnvr}} \) for each vehicle. In this simulation, vehicle 1 does not maneuver around any other vehicles and vehicle 3 must maneuver around all the other vehicles. Fig. 12 shows an overview of the vehicle trajectories overlaid on the windfield at one time instance. We examine the performance of vehicle 5 as a representative case in Figs. 13–15, showing snapshots of the vehicle navigating the environment, the course and velocity changes, and thrust required, respectively. All the vehicles clear all of the other obstacles/vehicles by their desired \( r_c \) values, and the maximum thrust is not violated for any vehicle. The computation time per sensor update was 0.15 s or less (for \( \leq 600 \) sensed points), which is well below the updated rate of 2 samples/s.

**VII. CONCLUSION**

The trajectory generator presented navigates a vehicle in an unknown environment collision free while respecting the vehicle’s physical limitations. The vehicle uses its sensor and communication inputs to compute course and velocity changes to avoid obstacles by a prescribed clearance distance. The sigmoid functions used to transition course and velocity provide piecewise smooth motion with bounded discontinuities and incorporate the course changes from each sensor update by matching the sigmoid slopes and summing the curves. In the event, the feasible course changes become an empty set, the vehicle adjusts priority temporarily to avoid collisions. Finally, the vehicle incorporates the expected wind disturbance, thrust limitations, and sensor constraints to bound the maximum safe cruise velocity.

There are several directions in which the algorithm presented in this article can be extended. One area is obstacles with nonconstant velocity and course. If the algorithm considered everything in the environment as if it were another vehicle, it would handle maneuvering obstacles as well. Using this approach, the algorithm would generate more conservative trajectories for obstacles with constant velocity and course than the trajectories presented in this article. Moreover, the algorithm is defined generically enough that it could be extended to three-dimensional motions by rotating the plane in which the motion occurs or generating a separate altitude adjustment that is combined with the planar trajectory. The thrust required for the altitude maneuver reduces the thrust available for planar motions, so when the two motions are combined, the vehicle’s thrust limitation is still respected. Finally, even though the simulations presented use stationary goal positions, the...
algorithm can handle moving goal positions so it can be incorporated into a higher-level motion planner.

REFERENCES

[1] Echodyne. [Online]. Available: www.echodyne.com
[2] L. Newmeyer, D. Wilde, B. Nelson, and M. Wirthlin, “Efficient processing of phased array radar in sense and avoid application using heterogeneous computing,” in Proc. 26th Int. Conf. Field Programmable Log. Appl., 2016, pp. 1–8.
[3] Sweep v1 Laser Scanner. [Online]. Available: http://www.robotshop.com/en/sweep-v1-360-laser-scanner.html
[4] M. Hoy, A. Matveev, and A. Savkin, “Algorithms for collision-free navigation of mobile robots in complex cluttered environments,” Robotics, vol. 33, pp. 463–497, 2014.
[5] M. Turpin, K. Mohta, N. Michael, and V. Kumar, “Goal assignment and trajectory planning for large teams of interchangeable robots,” Auton. Robots, vol. 37, pp. 401–415, 2014.
[6] S. Bhattacharya, R. Ghrist, and V. Kumar, “Persistent homology for path planning in uncertain environments,” IEEE Trans. Robot., vol. 31, no. 3, pp. 578–590, Jun. 2015.
[7] W. VanLoo, G. Pipeleers, M. Diehl, J. De Schutter, and J. Swevers, “Optimal path following for differentially flat robotic systems through a geometric problem formulation,” IEEE Trans. Robot., vol. 30, no. 4, pp. 854–865, Aug. 2014.
[8] J. Alonso-Mora, T. Naegele, R. Siegwart, and P. Beardsley, “Collision avoidance for aerial vehicles in multi-agent scenarios,” Auton. Robots, vol. 39, pp. 101–121, 2015.
[9] Z. Shiller, S. Sharma, I. Stern, and A. Stern, “Online obstacle avoidance at high speeds,” Int. J. Robot. Res., vol. 32, pp. 1030–1047, 2013.
[10] J. Chunyu, Z. Qu, E. Pollak, and M. Falash, “Reactive target-tracking control with obstacle avoidance of unicycle type mobile robots in a dynamic environment,” in Proc. Amer. Control Conf., Baltimore, MD, USA, 2010, pp. 1190–1195.
[11] S. H. Tang, C. K. Ang, D. Nakhaeinia, B. Karasfi, and O. Motlagh, “A reactive collision avoidance approach for mobile robot in dynamic environments,” J. Automat. Control Eng., vol. 1, pp. 16–20, 2013.
[12] A. Matveev, M. Hoy, and A. Savkin, “A globally converging algorithm for reactive robot navigation among moving and deforming obstacles,” Automatica, vol. 54, pp. 292–304, 2015.
[13] K. Chang, Y. Xia, and K. Huang, “Coordinated formation control design with obstacle avoidance in three-dimensional space,” J. Franklin Inst., vol. 552, pp. 5779–5795, 2015.
[14] K. E. Bekris, D. K. Grady, M. Moll, and L. E. Kavraki, “Safe distributed motion coordination for second-order systems with different planning cycles,” Int. J. Robot. Res., vol. 31, pp. 129–149, 2012.
[15] M. Zhang, Y. Chen, Q. Wang, and Y. Wang, “Dynamic artificial potential field based multi robot formation control,” in Proc. IEEE Intern. Conf. Meas. Technol., vol. 3, pp. 5779–5795, 2015.
[16] S. Sarkar and I. N. Kar, “Formation control of multiple group of robots,” in Proc. 52nd IEEE Conf. Decis. Control, 2013, pp. 1466–1471.
[17] C. Belva and V. Kumar, “Abstraction and control for groups of robots,” IEEE Trans. Robot., vol. 20, no. 5, pp. 865–875, Oct. 2004.
[18] R. Hedjar and M. Bounkhel, “Real-time obstacle avoidance for a swarm of autonomous mobile robots,” Int. J. Adv. Robot. Syst., vol. 11, pp. 67–78, 2014.
[19] K. Choi, Y. Kim, and I. Hwang, “Reactive collision avoidance of unmanned aerial vehicles using a single vision sensor,” J. Guid., Control, Dyn., vol. 36, pp. 1234–1240, 2013.
[20] S. Roelofsen, A. Martinoli, and D. Gillet, “Distributed deconfliction algorithm for unmanned aerial vehicles with limited range and field of view sensors,” in Proc. Amer. Control Conf., Chicago, IL, USA, 2015, pp. 4356–4361.
[21] M. Hoy, A. Matveev, and A. Savkin, “Collision free cooperative navigation of multiple wheeled robots in unknown cluttered environments,” Robot. Auton. Syst., vol. 60, pp. 1253–1266, 2012.
[22] E. Ferrera, J. Capitán, A. R. Castedo, and P. J. Marrón, “Decentralized safe conflict resolution for multiple robots in dense scenarios,” Robot. Auton. Syst., vol. 91, pp. 179–193, 2017.
[23] D. Cabecinhas, R. Cunha, and C. Silvestre, “Experimental validation of a globally stabilizing feedback controller for a quadrotor aircraft with wind disturbance rejection,” in Proc. Amer. Control Conf., Washington, DC, USA, 2013, pp. 1024–1029.
[24] S. Waslander and C. Wang, “Wind disturbance estimation and rejection for quadrotor position control,” in Proc. Amer. Inst. Aeronaut. Astronaut. Unmanned Unlimited, Seattle, WA, USA, pp. 1935–1948, 2009.
[25] N. Sydney, B. Smyth, and D. Paley, “Dynamic control of autonomous quadrotor flight in an estimated wind field,” in Proc. IEEE Conf. Decis. Control, 2013, pp. 3690–3616.
[26] N. Fischer, D. Hughes, P. Walters, E. Schwartz, and W. Dixon, “Nonlinear RISE-based control of an autonomous underwater vehicle,” IEEE Trans. Robot., vol. 30, no. 4, pp. 845–852, Aug. 2014.
[27] K. Cole and A. Wickenheiser, “Reactive trajectory generation in an unknown environment,” in Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst., Vancouver, BC, Canada, 2017, pp. 2151–2157.
[28] K. Cole and A. M. Wickenheiser, “Reactive trajectory generation for multiple vehicles in unknown environments with wind disturbances,” 2018. [Online]. Available: http://arxiv.org/abs/1708.03427
[29] N. Fischer, A. Kan, R. Kamalapurkar, and W. Dixon, “Saturated RISE feedback control for a class of second-order nonlinear systems,” IEEE Trans. Autom. Control, vol. 59, no. 4, pp. 1094–1099, Apr. 2014.
[30] G. Leishman, Principles of Helicopter Aerodynamics, Cambridge, U.K.: Cambridge Univ. Press, 2006.
[31] A. R. S. Bramwell, G. Done, and D. Balmford, Bramwell’s Helicopter Dynamics, Oxford, U.K.: AAIA and Butterworth-Heinemann, 2001.
[32] S. Newman, The Foundations of Helicopter Flight. New York, NY, USA: Halsted Press, 1994.
[33] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, “Quadrotor helicopter dynamics and control: Theory and experiment,” in Proc. Amer. Inst. Aeronaut. Astronaut. Guid., Navigation, Control Conf., Hilton Head, SC, USA, 2007.
[34] B. Bialy, J. Klotz, K. Brink, and W. E. Dixon, “Lyapunov-based robust adaptive control of a quadrotor UAV in the presence of modeling uncertainties,” in Proc. Amer. Control Conf., Washington, DC, USA, 2013, pp. 13–18.
[35] N. Cao and A. F. Lynch, “Inner-outer loop control for quadrotor UAVs with input and state constraints,” IEEE Trans. Control Syst. Technol., vol. 24, no. 5, pp. 1797–1804, Sep. 2016.
[36] K. Cole and A. Wickenheiser, “Impact of wind disturbances on vehicle station keeping and trajectory following,” in Proc. Amer. Inst. Aeronaut. Astronaut. Guid., Navigation, Control Conf., Boston, MA, USA, 2013.
[37] U.S. Military Specification, MIL-F-8785C, 1980.

Kenan Cole received the B.S. and M.S. degrees in mechanical engineering from The George Washington University (GWU), Washington, DC, USA, in 2007 and 2011, respectively. She is currently working toward the Ph.D. degree in mechanical engineering focused on multi-vehicle controls at GWU. Her research interests include vehicle controls, formation controls, and modeling environmental disturbances.

Adam M. Wickenheiser (M’08) received the B.S. degree in mechanical engineering (with a minor in applied mathematics) from Cornell University, in 2002 and the M.S. and Ph.D. degrees in aerospace engineering from Cornell University, Ithaca, NY, USA, in 2006 and 2008, respectively. From 2008 to 2009, he was a Postdoctoral Associate with the Sibley School of Mechanical & Aerospace Engineering, Cornell University. Since 2010, he has been Faculty with the Department of Mechanical and Aerospace Engineering, The George Washington University, Washington, DC, where he is currently an Assistant Professor. His current research interests include bioinspired flight, multifunctional materials and systems, and energy harvesting for autonomous systems.

Dr. Wickenheiser has served as the Chair of the Energy Harvesting Technologies (ICAST). He was the recipient of the 2011 Intelligence Community Young Investigator Award.