Uncertainty Velocity Obstacle Avoidance for sUAS Trajectory Planning in a 2D Plane

Jaron Ellingson
Mechanical Engineering
Brigham Young Univerity
Provo, UT 84602
jaronce@byu.edu

Emily Pitts
Electrical and Computer Eningeering
Brigham Young Univerity
Provo, UT 84602
evpitts@byu.edu

Cameron K. Peterson
Electrical and Computer Eningeering
Brigham Young Univerity
Provo, UT 84602
cammy.peterson@byu.edu

Karl Warnick
Electrical and Computer Eningeering
Brigham Young Univerity
Provo, UT 84602
warnick@byu.edu

Tim McLain
Mechanical Engineering
Brigham Young Univerity
Provo, UT 84602
mclain@byu.edu

***Abstract*—To guarantee safe autonomous navigation of multiple small unmanned aircraft systems (sUAS) operating within a limited airspace, collision avoidance algorithms must be robust to uncertainties in aircraft states. The velocity obstacle (VO) concept is a popular avoidance algorithm which uses a collision cone to effectively determine if two objects will collide in the near future. The VO method is a reactive algorithm that allows for avoidance of dynamic obstacles. This paper proposes a novel approach, the uncertainty velocity obstacle (UVO) method, to enhance the decentralized VO collision avoidance method by addressing uncertainties in the position and velocity of moving obstacles. A scenario is presented to illustrate the utility of this method for an sUAS encountering other cooperating vehicles. In this scenario the vehicles use global positioning system (GPS) messages to communicate position and velocity between cooperative vehicles. Each vehicle uses these states in a constant-jerk Kalman filter to estimate other vehicles' positions and velocities. Numerical simulations show that UVO enhances a vehicle's ability to avoid collisions when operating in uncertain environments.**

NOMENCLATURE

a	Ellipse x axis
b	Ellipse y axis
c	Intesecting or tangent line offset
m	Intesecting or tangentLine slope
A	Vehicle A
σ	Standard deviation
v_i'	Velocity point i from admissible velocities
v_d	Desired velocity incorporating buffer zone velocity
$\tilde{v}_d$	Desired velocity given no collision cones
v_{buf}	Buffer velocity
g	Buffer gain
b_r	Radius of buffer zone
c_r	Collision radius of vehicle
$d(A, I)$	Euclidean distance between vehicle A and I
f_r	Repulsive force
$v^{\max}$	Max velocity of a vehicle
$a^{\max}$	Max acceleration of a vehicle
(x, y)	Coordinates with respect to ellipse center
A_x	x position of vehicle A
A_y	y position of vehicle A
T_{line}	Line T
N	Number of Vehicles
Δt	Time step
$\mathbf{v}'$	Set of velocities satisfying $v^{\max}$ and $a^{\max}$

978-1-7281-2734-7/20/$31.00 ©2020 IEEE

TABLE OF CONTENTS

1. INTRODUCTION

Increasing the access of small unmanned aircraft systems (sUAS) to the National Airspace System (NAS) will benefit society by enabling sUAS use for package delivery [1], security and defense [2], medicine delivery [3], and recreation [4]. The Federal Aviation Administration (FAA) predicted a large increase in the numbers of commercial sUAS in the US [4]. To enable safe operation in dense traffic enviroments, sUAS need to be capable of autonomous collision avoidance despite uncertainty in the positions of other vehicles. This paper assists in the safe merging of sUAS into the NAS by increasing sUAS's capability to avoid collisions when operating in dense environments with imperfect information. This is accomplished by augmenting the velocity obstacle method [5] to account for uncertainty.

The main contributions of this paper include (1) augmenting the VOs collision cone to handle position and velocity uncertainty, (2) addressing a limitation of current VO methods when the collision cone is undefined, and (3) recommending the appropriate transmission range to promote safe operations.

Current VO methods assume perfect and instantaneous knowledge of the opposing vehicles' positions and velocities. In real-world operations this assumption is unrealistic. To prevent collisions, the sUAS must augment the current VO method to account for uncertainty. We accomplish this by modifying the collision cone to include position and velocity uncertainty. The results provide a dynamic collision cone that changes to account for varying vehicle uncertainty. The novelty of the UVO algorithm resides in how the uncertainty is accounted for while maintaining the original VO assumptions.

In traditional VO methods, the collision cone between two vehicles has the potential to become undefined. This occurs at a critical time when the vehicles are close together. To augment this deficiency and ensure safe operations, we enhance the VO algorithm with an artificial potential field that prevents vehicles from reaching this degenerate state.

In environments with many sUAS, estimating the positions and velocities of other vehicles is costly. Furthermore, the sUAS might only care about vehicles within a certain range. Therefore, we provide an analysis for finding the appropriate range for communicating GPS messages at which the sUAS should fly to prevent collisions.

This paper validates the new UVO method and its effectiveness in preventing collisions through Monte Carlo simulations. The paper proceeds as follows. In Section 2 we present previous and related work to UVO. In Section 3 we provide the mathematical derivation of UVO. Finally, Sections 4 and 5 go over our results and conclusions.

2. Related Works

This section summarizes related research that has been accomplished in VO avoidance methods and its extension of the reciprocal velocity obstacle (RVO) method. Key concepts to both of these methods including collision cones and velocity choosing will be explained. We also present prior work that incorporates different algorithms and technologies to enhance collision avoidance. They include artificial potential fields and global positioning system (GPS) sensors.

Velocity Obstacle Method

The VO method was first proposed by [6] to enable robot motion planning in dynamic environments. Using the robot's and obstacle's current position and velocity, avoidance maneuvers can be selected that place the robot's velocity on a collision-free path. The method was further developed in [7], which creates an area of collision with a dynamic obstacle called a collision cone. If the robots choose velocities outside of the collision cone they will avoid all obstacles. The algorithm handles dynamic obstacles of arbitrary size.

A shortcoming in VO is that approaching agents that both execute the method oscillate between avoiding each other and continuing on their desired trajectory. Oscillating paths are undesirable and [5, 8] redefine VO to mitigate this behavior. In [8], oscillations are reduced on sUAS by creating right-of-way or visual-flight rules. These rules mandate that only the aircraft who is not in the right-of-way move, thus solving most of the oscillation issues. In contrast, [5] redefines the collision cone by assuming that each agent will take half of the responsibility for moving out of the way. They also account for uncertainty in the robots' position, velocity, size, and dynamics to reduce oscillations. Their method, called reciprocal velocity obstacles (RVO), is summarized in Figure 1 and provides a starting point for this research. We first review the way in which RVO creates the collision cone to choose the appropriate velocities.

Collision Cones— All of the VO methods, RVO included, define a collision cone which depicts the area that, if a vehicle's velocity is inside the cone, will results in an imminent collision. For example, in Figure 1 the velocity of vehicle A, A_v, lies within the collision cone, shown by the solid triangle, and therefore is on a collision course with vehicle B.

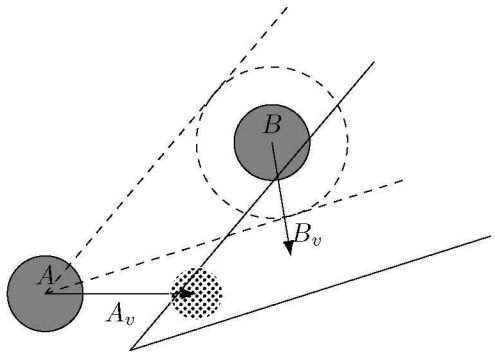


Figure 1: The RVO method. The dotted circle around B is the combined radii of the two vehicles and represents the Minkowski sum. The dotted triangle is the collision cone which is translated to the solid cone. Finally, the dots around A_v represent the admissible velocities.

The VO methods create a collision cone which originates from vehicle A and encompasses the Minkowski sum of vehicle A and B located at vehicle B (seen as dashed lines and circle in Figure 1). When both vehicles are circular, the Minkowski sum is a circle with radius equal to the sum of both vehicles' radii.

The collision cone is translated using a combination of the velocity of and A and B. The original VO method translates the cone by the velocity of B while RVO (shown here in Figure 1) is translated using an average of the vehicles. When more than two vehicles are present, all vehicles induce a separate collision cone on vehicle A. The VO methods then seek to find an achievable velocity which is outside all of the cones.

Choosing Appropriate Velocity—To select a velocity outside of a collision cone, the RVO method generates a cloud of admissible velocities (AV) that it can reach within a max acceleration. The approach in [5] for selecting a velocity is summarized next.

The admissible velocities,

$$AV(A_v) = \{v_i' \text{ s.t. } ||v_i'|| < v^{\max} \text{ and } ||A_v - v_i'|| < a^{\max}\Delta t\},\ \forall v_i' \in \mathbf{v}', \quad (1)$$

represents the set of velocities that vehicle A can reach within the next time step Δt, where $v^{\max}$ is the max velocity the sUAS can fly, $a^{\max}$ is it maximum possible acceleration, and $\mathbf{v}$ is all velocities that satisfy the $v^{\max}$ and $a^{\max}$ constraints.

The vehicle's velocity is chosen using

$$\text{penalty}(v_i') = \underbrace{||v_d - v_i'||}_{p_{\text{dist}}(v_i')} + \underbrace{w_{\text{time}}\frac{1}{t_c(v_i')}}_{p_{\text{col}}(v_i')}, \quad (2)$$

where w_{time} is a weight which specifies a vehicle's aggressiveness. This both minimizes the time to collision (t_c) and is the closest to the desired velocity. Refer to work in [5] for a more detailed explanation.

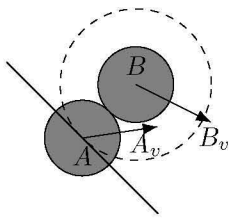


Figure 2: The degenerate case in VO happens when vehicle A and B touch. The tangent collision cone becomes a straight line allowing for vehicle A to choose a velocity which places it on a collision course with vehicle B.

Probabilistic VO—Finally, [9] presents a probabilistic variation to RVO through modeling velocity level chance constraints. Called PRVO, this method approaches the problem through Bayesian decomposition to calculate the individual effect of position and velocity. PRVO is similar to this research in that it models both positional and velocity uncertainty. In this paper, however, we use a direct geometric approach for incorporating the uncertainty that applies specifically to sUAS scenarios by calculating the estimated position and velocity of incoming vehicles in real time using a Kalman filter.

Potential Fields

Potential fields are a common collision avoidance technique where an artificially derived field is superimposed onto obstacles to create a repulsive force [10]. The vehicle's velocity is guided by the negative gradient of all the potential fields. This method provides a real-time decentralized avoidance scheme that drives a vehicle towards its goal location while avoiding multiple moving obstacles.

However, when using artificial potential fields, vehicles may get trapped in local minima with no guarantees of reaching its destination. Much of the literature regarding potential fields aim to overcome this shortcoming. For example, in [11] the authors reduced the effect of local minima by applying potential fields across the entire path of the vehicle. However, this method assumes the obstacles are static and known a-priori. In [12] a velocity-dipole method was developed that incorporates the velocity of the moving obstacles, the resulting potential fields are elliptical causing the avoiding vehicle to traverse around and behind moving obstacles. This method works well, but is not robust to multiple vehicles whose paths may cross.

In this paper, an artificial potential field is implemented to mitigate a known VO and RVO edge case. This edge case is illustrated in Figure 2, where vehicle A has calculated the Minkowski sum of vehicle B, denoted by the dashed circle, and is attempting to create the collision cone, denoted by the solid line. However, because the edge of the Minkowski sum lies at the center of vehicle A, there is only one tangent point to the Minkowski sum causing the tangent lines of the collision cone to become colinear. The vehicle is then unable to compute a collision-free velocity.

To mitigate this shortcoming, we augment RVO with guidance from potential fields. At close range, the potential field induces a strong repulsive force between the vehicles providing additional separation. The strength of the repulsive force is dependent on the distance between the vehicles and drops off sharply when they are outside this edge case. At distant ranges the potential field has a minimal effect on the vehicle's velocity and RVO exclusively guides the vehicle's path. In this way we utilize the benefits of artificial potential fields while avoiding their shortcomings.

Collision Avoidance Sensors

The numerical simulation for this work focused on a cooperative scenario for the sUAS. The scenario has each sUAS share GPS position and velocity while using a constant jerk Kalman filter as described in [13], to approximate the position and velocity of the other aircraft.

GPS Sharing—The sharing of GPS among commercial vehicles is implemented through the FAA's mandate to have all aircraft flying A, B, C and some E class airspace to be equipped with an Automatic Dependent Surveillance-Broadcast (ADS-B) system by January 2020 [14]. ADS-B is a technology that allows aircraft to periodically broadcast its GPS position as well as other information such as altitude, speed, flight number, type of aircraft, and maneuvering state (turning, climbing, or descending) [15]. This information allows other aircraft and air traffic control stations to aid in safe separation of the aircraft. Aircraft below 10,000 ft (3,048 m) are intended to use ADS-B on 978 MHz while all other aircraft can use 1090 MHz frequency. Incorporating ADS-B into sUAS is still a question of research and this work provides information about how ADS-B should be implemented by understanding collision avoidance ranges.

ADS-B is an attractive sensor for detect and avoid (DAA) systems and has been analyzed in a variety of studies. Initial research [16] suggests that the use of ADS-B would greatly enhance DAA operations for sUAS because of its superior range, resolution, accuracy, and update rate to any other sensor. This study also suggests that ADS-B would be greatly augmented by optical, radar, or other sensors that offer detection for non-cooperative vehicles. In [17] a DAA system is proposed that uses ADS-B to track intruders, detect collisions, avoid collisions, and update waypoints to achieve safe operations. It uses a Kalman filter for the estimates and characterizes the error of the states using the ADS-B uncertainty parameters and errors associated with latency, resolution, and message success rate. Our work is different in that we use a modified VO method as our avoidance algorithm.

3. METHODS

In this section we will explain the methods used to create the UVO in a 2D formulation. We first briefly highlight the geometry equations found in the derivation of UVO. We then show the formulation for position and velocity uncertainty collision cones and explain how to calculate their associated standard deviations. Finally, we show how to choose the appropriate velocity and how we implemented the potential field technique.

Ellipse Geometry

The UVO method assumes that the vehicles are operating in a plane in 2D, therefore the derivations of the UVO method start with the geometric equations for an intersection between a line $y = mx + c$ and an ellipse,

$$b^2x^2 + a^2y^2 = a^2b^2. \tag{3}$$

Combining the two yields

$$(a^2m^2 + b^2)x^2 + 2a^2mcx + a^2(c^2 - b^2) = 0. \quad (4)$$

Next we solve Equation (4) for x and y using the quadratic formula to obtain

$$x_{1,2} = \frac{-a^2mc \pm ab\sqrt{a^2m^2 + b^2 - c^2}}{a^2m^2 + b^2}, \quad (5)$$

and

$$y_{1,2} = \frac{b^2c \pm abm\sqrt{a^2m^2 + b^2 - c^2}}{a^2m^2 + b^2}. \quad (6)$$

The value of discriminate $D = a^2m^2 + b^2 - c^2$, in Equations (5) and (6), define the intersection points of the line and ellipse. We will call D the tangency condition. From this condition we can determine the intersection characteristics

i) if $D < 0$, then we have two intersections.

ii) if $D = 0$, then the line is tangent to the ellipse.

iii) if $D > 0$, then the line does not intersect the ellipse.

This derivation gives us a way to calculate whether or not a line is tangent to an ellipse base on the discriminant. This will be important as we seek to find the ellipse which is tangent to every admissible velocity.

Position Uncertainty

The Kalman filter estimation provides an error covariance in the x and y directions for both position and velocity. This error covariance provides an error ellipse at a certain standard deviation. We will use these error ellipses in the following derivations.

The positional uncertainty can be summarized through Figure 3 where a collision cone for vehicle A is induced by vehicle B. The standard deviation of B's position uncertainty is calculated through the Kalman filter update. The position uncertainty then augments the collision cone. For example, the dotted region in Figure 3 may represents one standard deviation of uncertainty. More standard deviations would mean a bigger ellipse, a bigger collision cone, and a greater area to avoid but higher probability of avoiding the other aircraft. The Minkowski sum is therefore represented as an ellipse and the collision cone is redefined to encapsulate the whole Minkowski sum.

To construct this new collision cone we solve Equation (3) for y and take its derivative with respect to x

$$\frac{dy}{dx}_{\text{ellipse}} = \frac{-b^2x}{a^2y}. \quad (7)$$

Equation (7) represents the slope of the tangent line of any point (x,y) on the ellipse. We can combine this equation with the slope of the line originating from vehicle A and extending to a point tangent to the ellipse. The slope of the line is defined as

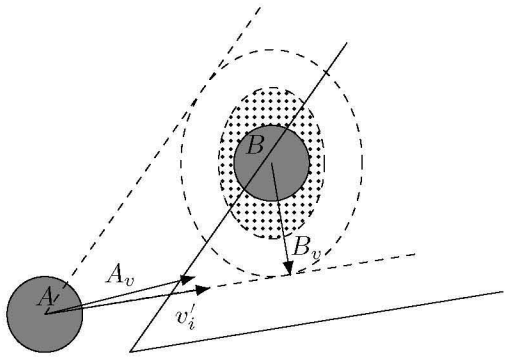


Figure 3: Position uncertainty method where the dashed ellipse is the Minkowski sum, the dotted ellipse is the uncertainty, the dashed cone is the collision cone, and the solid cone is the final translated collision cone. The velocities of both vehicles are represented as well as an admissible velocity of A called v_i'.

$$\frac{dy}{dx}_{\text{line}} = \frac{A_y - y}{A_x - x}, \quad (8)$$

where A_x and A_y are the x and y position of vehicle A. This equation determines the slopes of the dashed lines in Figure 3.

We combine Equations (7) and (8) to give

$$a^2y^2 + b^2x^2 = a^2A_yy + b^2A_xx.$$

Finally we substitute in Equation 3 and and solve for y to get

$$y = \frac{b^2}{A_y} - \frac{b^2A_x}{a^2A_y}x. \quad (9)$$

Choosing $c = \frac{b^2}{A_y}$ and $m = -\frac{b^2A_x}{a^2A_y}$ provides the equation for the intersecting line. The x positions are found using Equation (4).

Velocity Uncertainty

Similar to position uncertainty, our velocity uncertainty also affects the construction of the collision cone (Figure 4). The top of the translated collision cone represents the mean of the uncertainty, while the dotted ellipse once again represents the standard deviation chosen to encompass a desired percentage of the error covariance. To avoid this uncertainty we can draw a cone that has the same shape as the original RVO but is expanded to engulf the entire uncertainty ellipse.

To construct the velocity uncertainty collision cone we start with Equation (3), solve for y, and take its derivative with respect to x

$$\frac{dy}{dx} = \frac{-b}{a^2}\sqrt{\frac{x^2a^2}{1 - x^2}}. \quad (10)$$

This is the tangent slope of any x value in the ellipse.

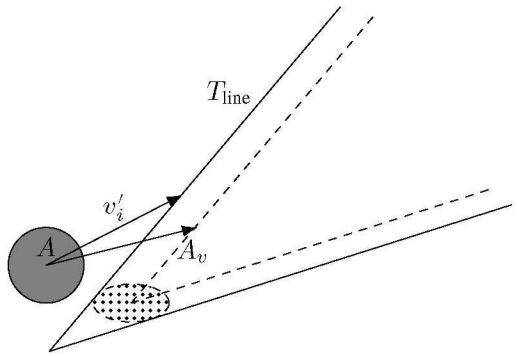


Figure 4: Velocity uncertainty compensation method where the dashed line is the translated collision cone from the RVO method and the dotted ellipse is the velocity uncertainty. The collision cone is further augmented to engulf the entire velocity uncertainty. Vehicle A's velocity is A_v and one admissible velocity is v_i'.

Next, we substitute the slope m of T_{line}, as shown in Figure 4, for $\frac{dy}{dx}$ and solve for x to get

$$x = \sqrt{\frac{m^2 a^4}{b^2 + m^2 a^2}}. \tag{11}$$

To obtain the corresponding y values, we substitute Equation (11) into the ellipse Equation (3).

Determining Standard Deviation

Thus far, we assumed that we wanted to stay outside the position and velocity uncertainty ellipses by one or two standard deviations. In this section, we invert the problem to solve for what standard deviation we are avoiding given a velocity v_i'. For example, in Figures 3 and 4 the point v_i' represents an admissible velocity whose point lies tangent to the depicted standard deviation in the figure. If we were to pick a different admissible velocity, we could either enlarge or reduce a scalar multiple of the standard deviation to satisfy the tangency condition $a^2m^2+b^2 = c^2$. To solve for the standard deviation we are avoiding, we first substitute in known values in for a and b at one standard deviation, $a = \sigma a_\sigma$ and $b = \sigma b_\sigma$. Fortunately, a_σ and b_σ can be directly calculated from the Kalman filter's covariance.

Second, we solve the tangency condition for σ to get

$$\sigma = \sqrt{\frac{c^2}{a_\sigma^2 m^2 + b_\sigma^2}}. \tag{12}$$

Equation (12) allows us to quantify how far a point is from the mean relative to the standard deviation and helps us associate a cost to every admissible velocity.

For both positional and velocity uncertainty σ becomes a function of each admissible velocity v_i' because m and c (Equation (12)) are the tangent lines created in Figures 3 and 4. For the position tangent line, we use Equations (9) and (3) and for the velocity tangent line we use Equations (11) and (3). Therefore Equation (12) becomes $\sigma_{\text{pos}}(v_i')$ and $\sigma_{\text{vel}}(v_i')$ for position and velocity.

Choosing Appropriate Velocity

Once we associate a standard deviation with uncertainty, we choose the best commanded velocity subject to the maneuvering capabilities of the vehicle. We start by sub-sampling the cloud of admissible velocities (see Equation (1)) and assigning penalties to each sample.

First, the penalties for distance and time-to-collision have been defined in Equation (2). Next, we assign position and velocity uncertainty penalties based on how many standard deviations the corresponding point lies away from the mean. The penalties for position and velocity become

$$p_{\text{pos}}(v_i') = w_{\text{pos}} \frac{1}{\sigma_{\text{pos}}(v_i')}, \tag{13}$$

and

$$p_{\text{vel}}(v_i') = w_{\text{vel}} \frac{1}{\sigma_{\text{vel}}(v_i')}, \tag{14}$$

We then combine the four penalties,

$$p(v_i') = p_{\text{dist}}(v_i') + p_{\text{col}}(v_i') + p_{\text{pos}}(v_i') + p_{\text{vel}}(v_i'), \tag{15}$$

to calculate a final penalty for each sampled admissible velocity v_i'.

Once all samples have an assigned final penalty, we use least squares to fit them to an order n polynomial surface S,

$$S = c_0 + c_{10}x + c_{11}y + c_{20}x^2 + c_{21}xy + c_{22}y^2 + \cdots + c_{n0}x^n + \cdots + c_{nn}y^n, \tag{16}$$

where $c_0, c_{10}, \ldots, c_{nn}$ are the polynomial coefficients.

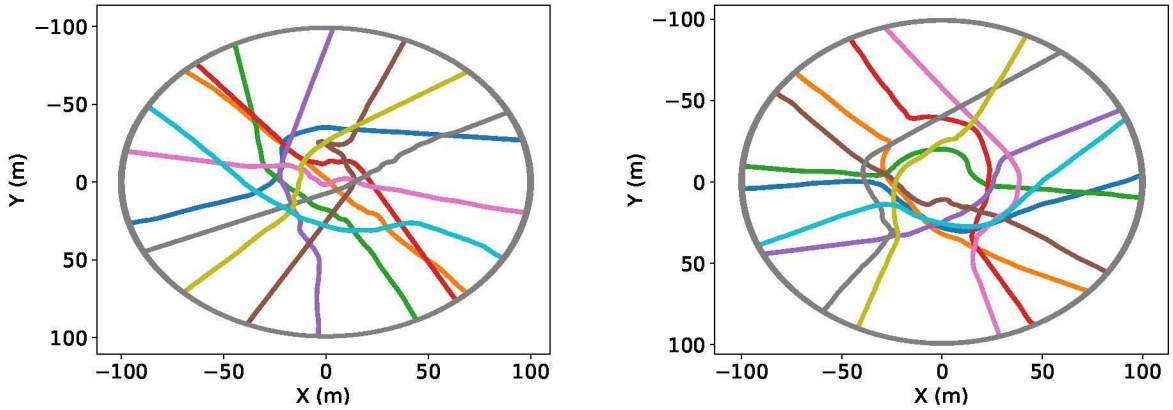
Finally, we use this surface equation to solve for a minimum location. First, we find the smallest set, or convex hull, of penalties which contains all the points. These points and edges define our boundaries and our constraints to solve for the minimum. We then use the Ceres Solver [18], a C++ optimization library, to find a minimum on the surface. This minimum corresponds to the best velocity given the penalties and is commanded in the next time step. However this velocity does not guarantee that we are outside the collision cone, but instead implies that we are headed outside the cone.

Potential Field

When two (or more) vehicles are in close proximity the evading vehicle (A) can enter a degenerate state where the collision cone is undefined. In this section we present a modification to RVO that mitigates this edge case by incorporating a localized artificial potential field (PF). The PF forces a buffer zone between two vehicles so that the distance of the Minkowski sum of the incoming (invading) vehicle never overlaps with the location of the evading vehicle, ensuring that the collision cone is well-defined. The PF uses distances between the vehicles and a gain method to create a new

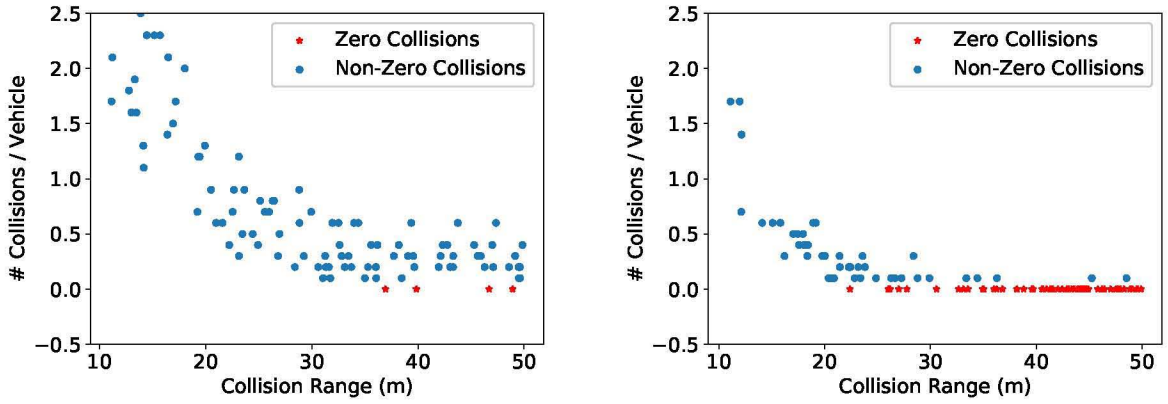
Table 1: GPS Sharing Simulations.

Case	Kalman Filter	Potential Field	UVO	Communication Drop %	Position σ (m)	Velocity σ (m/s)	Number of Vehicles	Max Velocity (m/s)
RVO Worst Case	-	-	-	50	3.0	1.0	10	5
RVO Best Case	-	-	-	0	0.0	0.0	10	5
KF-RVO	X	-	-	50	3.0	1.0	10	5
PF	X	X	-	50	3.0	1.0	10	5
UVO	X	-	X	50	3.0	1.0	10	5
UVO-PF	X	X	X	50	3.0	1.0	10	5



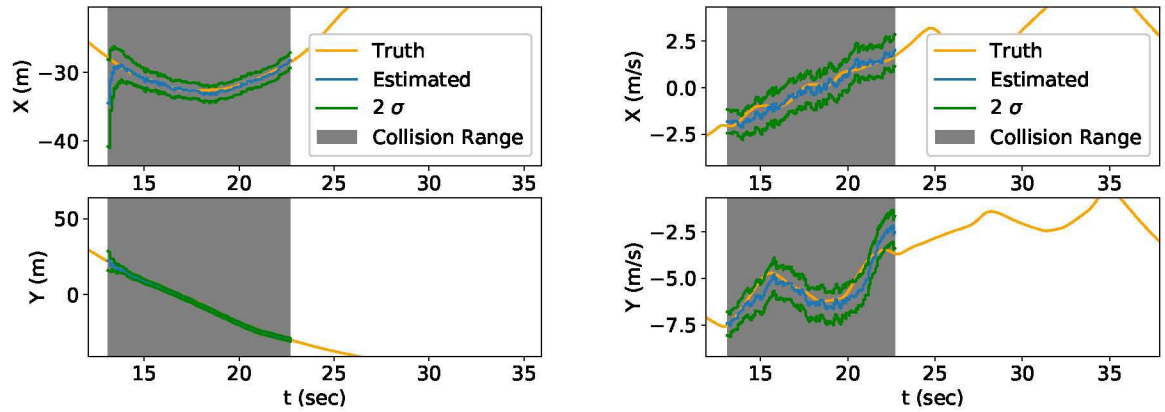
(a) RVO (b) UVO-PF

Figure 5: The routes of the sUAS are shown for agents at a random starting positions on a circle with a 100 meter radius and with a collision range of 50 meters. The agents are given a waypoint goal directly across the circle from their starting point. (a) shows the RVO method with noise characteristics defined in Table 1 and low deviation from a straight path. On the other hand, (b) shows UVO with with noise characteristics defined in Table 1 and it performing more conservatively to better guarantee collision avoidance. The UVO weights for time to collision and position and velocity uncertainty were respectively 100, 1, and 1 (see Table 2).



(a) Worst case results. (b) Best case results.

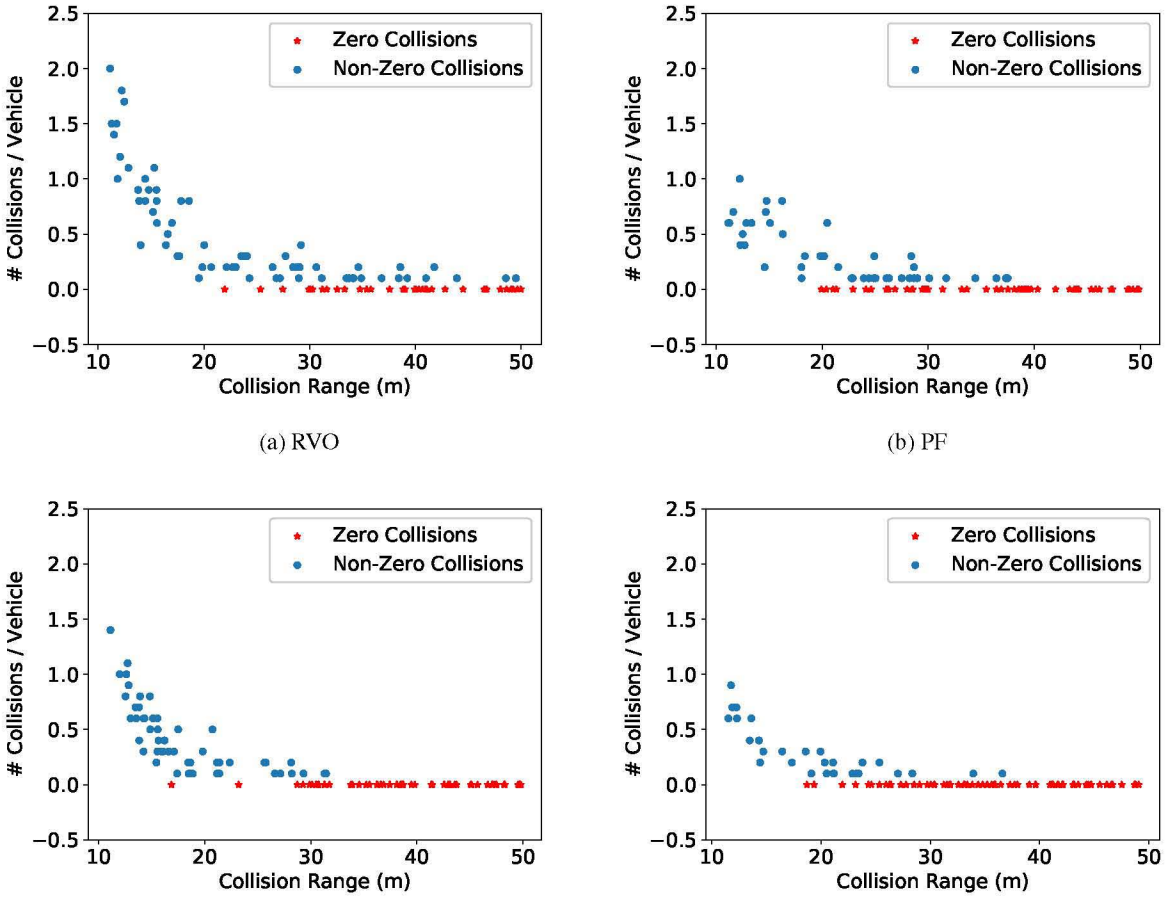
Figure 6: These tests highlight the upper and lower baseline for all our simulations. The worst case always has collisions while the best case significantly reduces the number of collision which occur. This is most likely due to the reaction time of the sUAS. The range axis indicates a particular run's collision range.



(a) GPS Position Kalman Filter

(b) GPS Velocity Kalman Filter

Figure 7: The constant-jerk Kalman filter examples with truth, estimates, and 95% confidence bounds. This example has 10 sUAS with a collision range of 40 meters and zero communication dropout with noise characteristics of 3.0 meters and 1.0 meter/sec for position and velocity. The 40 meter collision range, indicated by the gray shading, is why the estimate is only calculated for a short period of time (roughly for 15 seconds).



(a) RVO

(b) PF

(c) UVO

(d) UVO-PF

Figure 8: GPS sharing scenario test results. The progression from RVO to using both the potential field and UVO is shown. There is a visual improvement between RVO, UVO, and finally both using the potential fields and UVO. While the potential fields do not seem to radically improve the avoidance, combining the potential fields with UVO appears to slightly improve avoidance.

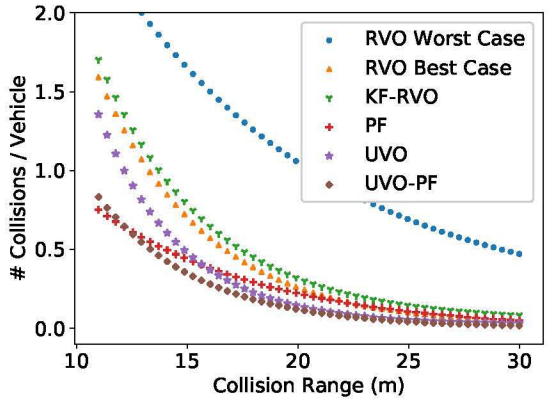


Figure 9: GPS Best Fits

desired velocity for the evading vehicle. The magnitude of the repulsive force between vehicle A and an invading vehicle I is given by

$$f_r = \left| \left(\frac{b_r}{d(A,I) - b_r} \right)^g \right| \tag{17}$$

where $d(A,I)$ is the Euclidean distance between vehicle A and I, b_r is the radius of the buffer zone, and g is a constant gain. This force drops off sharply for distances outside the buffer zone radius, while inside distances will generate a large repulsive force.

The force is used to determine an average velocity induced by the PF as

$$v_{\text{buf}} = \frac{1}{N-1} \sum_{j=0}^{N-1} f_r \frac{-e(A,I_j)}{\|e(A,I_j)\|}, \tag{18}$$

where N is the total number of vehicles, $e(A,I_j) = [A_x - I_{j,x}, A_y - I_{j,y}]^T$ is vector pointing from the position of vehicle A to the position of vehicle j^{th} invading vehicle, and $\|\cdot\|$ is the 2-norm.

Each vehicle's modified desired velocity is computed as

$$v_d = \frac{2\tilde{v}_d + v_{\text{buf}}}{2}. \tag{19}$$

The penalties placed on admissible velocities are then calculated using v_d.

4. Results

The following section highlights results from performing the UVO algorithm in realistic scenarios and compares this method to RVO. More specifically, this section will overview performance differences between UVO and RVO, highlight Monte Carlo results from the two scenarios, and finally, recommend a safe collision range separation for sUAS using this method.

All of these results are based on Monte Carlo simulations where the sUAS have real-world sUAS dynamics and control. The vehicles are quadrotors with a mass of 5 kg, mass moment of inertia of 0.6271 kg m^2, 0.6271 kg m^2, 1.25 kg m^2, linear drag coeficients of 0.1, 0.1, 0.001, and angular drag coefficients of 0.001, 0.001, 0.001. Each quadrotor is also running a PID controller on roll, pitch, and yaw rate. The vehicles are randomly placed on a radius of 100 meters, are randomly assigned a collision range, and are commanded to achieve a waypoint exactly opposite from the other agent (see Figure 5).

The collision range is defined as the range at which a sUAS takes into consideration another vehicle by tracking its movement and performing collision avoidance maneuvers. For example, if a vehicle is located at a distance 30 m away but the sUAS has a collision range of 20 m, then the vehicle will not induce a collision cone. If one or more collisions occurs among any vehicles for that run, then it is indicated with a blue dot. A collision happens if any two vehicles get within $2c_r$ m of each other, where c_r is the radius that represents the vehicles' physical size.

All of the scenarios use the following max acceleration and weights shown in Table 2. RVO and UVO both use the same max acceleration and time to collision weight. Only UVO uses the weights for both position and velocity uncertainty. Additionally the buffer zone radius b_r is define as well as the collision radius c_r. Other variables such as the maximum velocity, the number of vehicles, and amount of noise introduced into the system are defined for each scenario (see Table 1).

Table 2: Results Variables and Weights

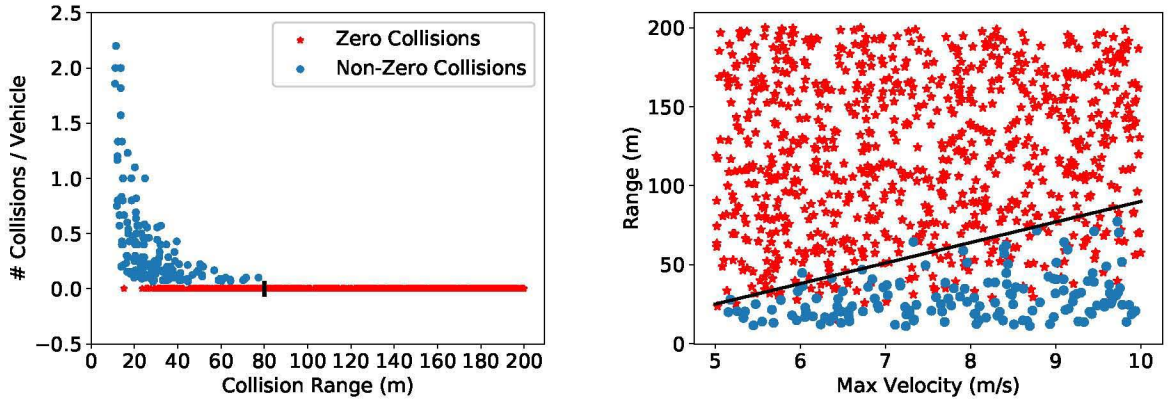
Weight/Variable	Value
$a^{\max}$	100 m/sec^2
w_{time}	100
w_{pos}	1
w_{vel}	1
b_r	5 m
c_r	1 m

Performance

The difference in performance of RVO and UVO is highlighted in Figure 5 where the Figure 5a is RVO and Figure 5b is UVO. They both have a collision range of 50 meters and avoid all other agents. The gray circles are where the vehicles start and solid lines represent the paths of the vehicles.

One major difference between RVO and UVO is that RVO allows the agents to select a path that is close to the straight line path from the starting point to their ending point. UVO, on the other hand, directs the vehicles to a more diverging route to avoid regions of uncertainty where other vehicles may be located. This difference results from selecting high weights for position and velocity uncertainty (Equations (13) and (14)) and the agents are diverging from all other agents that have high uncertainty.

With this behavior in mind, UVO was tuned in the following scenarios so that the agents would have moderate deviation from a straight-line path. Large weights on the uncertainty



(a) (b)

Figure 10: GPS scenario test with realistic characteristics. This scenario had GPS noise of 1 meter and 0.1 meter/sec for position and velocity. It varied the number of vehicles from 5 to 15 and the max velocity from 5 to 10 m/s. (a) shows 1000 runs as well as the recommended collision range for operating in velocities ranging from 5 to 10 m/s. We choose to set this point at 80 meters because no collisions happen greater than this range. (b) shows max velocity verses range for each simulation, with red stars representing zero collisions and blue dots as one or more collisions. This plot also shows an upper boundary for collisions. Almost every blue dot which represents a collision is below this line and represents the relationship between range and max velocity.

penalties tend to push the agents away from an optimal path but also decreases the chance for collisions.

Additionally, as a starting point for the scenarios, we ran both a worst case and a best case scenario for vehicles using the RVO method. (Figure 6). Both cases do not use a Kalman filter (i.e. RVO uses the unfiltered GPS measurements) and the noise characteristics defined in Table 1. Notice that in the RVO Best Case, the method still has collisions despite having perfect knowledge of another vehicle. We postulate that this is due to a combination of two reasons. First, RVO seems to be less conservative as explained above and shown in Figure 5. Second, RVO fails occasionally when vehicles are too close as described and then resolved in Section 3.

GPS Sharing

For the real-world GPS sharing scenario, we constrained the sUAS to a max velocity of 5 m/sec and the number of vehicles to 10. The vehicles share both their position and velocity with standard deviation noise characteristics of 3.0 meters and 1.0 meters/sec.

Figure 7 shows how the constant-jerk Kalman filter performs in this scenario. Notice that the estimation only occurs when the agents are within the specified collision range (shown with the gray background). Additionally, we modeled the GPS simulation to have random GPS message dropout. This dropout simulates bandwidth issues that might occur if too many sUAS are in the airspace. For the GPS simulations, we use a random dropout of 50%.

The GPS sharing scenarios are defined in Table 1 with the combination of a Kalman filter and RVO being KF-RVO and the combination of UVO and PF being UVO-PF. The corresponding results are shown in Figures 8 while Figure 9 shows the exponential best fit of each of the scenarios. The KF-RVO method shows how pairing the RVO method with a Kalman filter allows the RVO method to approach the RVO Best Case. Additionally, both the PF and the UVO simulations give better results than the RVO Best Case suggesting that both high uncertainty and degenerate RVO situations caused RVO to fail. Finally, the UVO and the UVO-PF scenarios are very similar but overall the UVO-PF simulation does slightly better. This is because the PF was designed to only mitigate occasional situations where the vehicles enter an undesirable configuration. Overall UVO-PF performs the best.

Recommendations

To recommend an appropriate collision range for sUAS we ran 1000 UVO-PF Monte Carlo simulations where each was assigned a random collision range. Additionally, we varied the number of vehicles between 5-15 and the max velocity between 5-10 m/sec for each simulation. The results of the simulations are shown in Figure 10.

Figure 10a shows the collision data similar to the plots in Figure 8. We would recommend a collision avoidance range roughly 80 meters because no collisions happen after this point. Selecting a collision range above 80 meters will achieve an appropriate balance of collision avoidance and broadcast range. Note that this recommendation applies specifically to sUAS with similar characteristics to the sUAS in these scenarios.

Figure 10b shows the maximum velocity verses range for each simulation. This plot highlights zeros collision with red starts and more than one collision with blue dots. We added the line to visually highlight the trend between range and max velocity. This trend supports the hypothesis that faster sUAS will need larger collision ranges. We expect that this trend can be extrapolated to faster sUAS.

5. Conclusion

This paper has proposed an extension to the VO method by incorporating positional and velocity uncertainty in the

formulation of the collision cone. We derived collision cone equations as well as introduced a potential field to provide a way to counteract degenerate states when vehicles are too close together. Finally, we have presented results which show that the combination of the uncertainty cones and the potential field perform better in high-uncertainty environments.

From the work presented we can draw the following conclusions.

1) The combination of the UVO and the potential fields (UVO-PF) outperform the original RVO method when there is a high amount of uncertainty in a system.

2) Collision range and velocity of the sUAS appear to be the primary characteristics that affect whether a sUAS will be successful in avoiding another sUAS.

3) We recommend a collision avoidance range of 80 meters and above when flying the UVO-PF method with aircraft having similar capabilities to the sUAS described here.

Future work includes extending the algorithm to 3D as well as including additional variables into the Monte Carlo simulations. All of the agents have exactly the same dynamics and control scheme but variation would provide more realistic simulations. Additionally, this work could benefit from agents that are unable to go below a certain velocity (i.e. fixed-wing aircraft). Finally, future work will include hardware verification of the algorithm.

ACKNOWLEDGMENTS

This work is supported by NSF award No. 1727010 and the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation Industry/University Cooperative Research Center (I/UCRC) under NSF award No. IIP-1161036 along with significant contributions from C-UAS industry members.

REFERENCES

[1] Joshuah K Stolaroff, Constantine Samaras, Emma R ONeill, Alia Lubers, Alexandra S Mitchell, and Daniel Ceperley. Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. *Nature communications*, 9(1):409–422, 2018.

[2] Noel Sharkey. The automation and proliferation of military drones and the protection of civilians. *Law, Innovation and Technology*, 3(2):229–240, 2011.

[3] Timothy Amukele, Paul M Ness, Aaron AR Tobian, Joan Boyd, and Jeff Street. Drone transportation of blood products. *Transfusion*, 57(3):582–588, 2017.

[4] Federal Aviation Administration (FAA). FAA Aerospace Forecast: Fiscal Years 2019-2039. 2019.

[5] Jur Van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935. IEEE, 2008.

[6] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998.

[7] Animesh Chakravarthy and Debasish Ghose. Obstacle avoidance in a dynamic environment: A collision cone approach. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 28(5):562–574, 1998.

[8] Yazdi I Jenie, Erik-Jan Van Kampen, Coen C de Visser, and Q Ping Chu. Selective velocity obstacle method for cooperative autonomous collision avoidance system for unmanned aerial vehicles. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, page 4627, 2013.

[9] Bharath Gopalakrishnan, Arun Kumar Singh, Meha Kaushik, K Madhava Krishna, and Dinesh Manocha. Prvo: Probabilistic reciprocal velocity obstacle for multi robot navigation under uncertainty. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1089–1096. IEEE, 2017.

[10] Masatoshi Okutomi and Masahiro Mori. Decision of Robot Movement By Means of a Potential Field. *Advanced Robotics*, 1(2):131–141, 1986.

[11] Charles W. Warren. Global Path Planning Using Artificial Potential Fields. page 6. IEEE, 1989.

[12] Enxiu Shi, Tao Cai, Changlin He, and Junjie Guo. Study of the new method for improving artifical potential field in mobile robot obstacle avoidance. *Proceedings of the IEEE International Conference on Automation and Logistics, ICAL 2007*, (50375119):282–286, 2007.

[13] Kyle Ingersoll. Vision based multiple target tracking using recursive RANSAC. Master's thesis, Brigham Young University, 2015.

[14] Federal Aviation Administration. Automatic dependent surveillance–broadcast (ADS–B) out performance requirements to support air traffic control (ATC) service. 2010.

[15] S. B. Hottman, K. R. Hansen, and M. Berry. Literature review on detect, sense, and avoid technology for unmanned aircraft systems. *Tech. Rep.*, 2009.

[16] Brandon Stark, Brennan Stevenson, and YangQuan Chen. ADS-B for small unmanned aerial systems: Case study and regulatory practices. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 152–159. IEEE, 2013.

[17] Sahawneh. L., Duffield M., Beard R., and McLain T. Detect and avoid for small unmanned aircraft systems using ADS-B. *Air Traffic Control Quarterly*, 23(2-3):203–240, 2015.

[18] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

BIOGRAPHY

Jaron Ellingson *is a master student in mechanical engineering at Brigham Young University and currently enjoys researching in the Multiple Agent Intelligent Coordination and Control lab at BYU. He has also been a project lead in other research projects and student competitions such as Deep Remote Control, Mars Rover University Challenge, and the AUVSI Student Competition. In his free time he enjoys rock climbing and hiking.*

Emily Pitts *is an undergraduate student in electrical engineering at Brighan Young University. She will receive her BS from BYU in December of 2019, and hopes to continue on to a Masters program at BYU. Her research interests include control systems and signal processing.*

Cameron Peterson *received her Bachelor and Master degrees in Applied Physics from Brigham Young University and Johns Hopkins University respectively. She received her Ph.D. in Aerospace from the University of Maryland in 2012. She has over fifteen years of experience working as an engineer in the defense industry, most recently for Johns Hopkins University. She joined the Department of Electrical and Computer Engineering of Brigham Young University as an Assistant Professor in 2016. Her research interests include tracking, estimation, autonomy, and multi-vehicle motion coordination.*

Karl Warnick *received the B.S. degree in Electrical Engineering and Mathematics and the Ph.D. degree in Electrical Engineering from Brigham Young University (BYU), Provo, UT, in 1994 and 1997, respectively. From 1998 to 2000, he was a Postdoctoral Research Associate and Visiting Assistant Professor in the Center for Computational Electromagnetics at the University of Illinois at Urbana-Champaign. Since 2000, he has been a faculty member in the Department of Electrical and Computer Engineering at BYU, where he is currently a Professor. Dr. Warnick has published many books, scientific articles and conference papers on electromagnetic theory, numerical methods, antenna applications, and high sensitivity phased arrays for satellite communications and radio astronomy.*

Tim McLain *is a professor of mechanical engineering at Brigham Young University. He received BS and MS degrees from BYU and a PhD from Stanford University, all in mechanical engineering. He joined BYU as a professor in 1995. During 1999 and 2000, he was a visiting scientist at the Air Force Research Laboratory where he initiated research in the guidance and control ofunmanned aircraft systems. Since then, his UAS research has attracted the support of the Air Force, the Army, DARPA, NASA, NSF, ONR, NIST, and numerous companies. He is a co-author of the textbook Small Unmanned Aircraft and the author of over 140 peer-reviewed articles. From 2012 to 2019, he was the director of the Center for Unmanned Aircraft Systems sponsored by the National Science Foundation. He currently serves as the associate dean of the Fulton College of Engineering at BYU.*