CEAS-GNC-2022-022

CEAS EuroGNC 2022
"Conference on Guidance, Navigation and Control"
3-5 May 2022 @ Technische Universität Berlin, Germany

Optimal Intruder Collision Avoidance for UAVs via Waypoint Tracking

Aniruddha Perumalla Ph.D. Student, Pennsylvania State University, Department of Aerospace Engineering, University Park, Pennsylvania, United States 16802. amp7226@psu.edu

Eric Johnson Professor, Pennsylvania State University, Department of Aerospace Engineering, University Park, Pennsylvania, United States 16802. eric.johnson@psu.edu

Puneet Singla Professor, Pennsylvania State University, Department of Aerospace Engineering, University Park, Pennsylvania, United States 16802. psingla@psu.edu

Anusna Chakraborthy Research and Development Scientist, UtopiaCompression Corporation, Los Angeles, California, United States 90064. anusna@utopiacompression.com

Lukas Liebischer Research and Development Software Engineer, UtopiaCompression Corporation, Los Angeles, California, United States 90064. lukas@utopiacompression.com

ABSTRACT

This paper presents a guidance strategy for UAV navigation to a destination while safely avoiding static and moving obstacles along the way, building on previous work that reduces the problem of collision avoidance to one of waypoint tracking. The strategy developed in this paper introduces a "safety ellipsoid" approach, a more general variation of the "collision cone" introduced in previous work, to assess whether obstacles are critical enough to require a maneuver, and generate temporary waypoints if so. A well-known proportional navigation optimal guidance law is utilized to reach the waypoints with minimum control effort and thereby avoid the obstacles. The strategy is validated via physically accurate, image-in-the-loop simulation, using Microsoft AirSim, of a head-on encounter between a UAV and an intruder aircraft whose position is estimated through a visual tracking system. The simulations demonstrate that this approach may be particularly useful in collision scenarios in which aircraft must meet international "remain well-clear" (RWC) standards to avoid other vehicles in the airspace.

Keywords: obstacle avoidance; UAV navigation; optimal control; collision avoidance; autonomous aircraft

Nomenclature

p	=	position
v	=	velocity/speed
a	=	acceleration
d	=	distance
$\mathbf{x}$	=	state vector
$q\|_{t=t_f}$	=	generic function or variable q evaluated at terminal time t_f
$\boldsymbol{\lambda}$	=	Lagrange multiplier (co-state) vector

J	=	cost function
Φ	=	terminal state cost function
L	=	Lagrangian
H	=	Hamiltonian
$\dot{q}$	=	time derivative of generic function or variable q
m, n	=	terminal state cost weights
t_{go}	=	"time-to-go" to obstacle or waypoint
u^*	=	optimal control input
K	=	feedback control gain
d_{safe}	=	generic safe miss distance(s) for obstacle avoidance
$d_{\text{safe,x}}$	=	safe miss distance in the x-direction
t_{safe}	=	maximum time to obstacle location, under which maneuver must begin to avoid collision

1 Introduction

Collision avoidance is an important requirement for modern unmanned aerial vehicles (UAVs). On one hand, many popular applications of UAVs, such as manufacturing facility inspection and search-and-rescue, often demand navigation in small, congested environments in which static and moving obstacles may hinder movement. On the other hand, UAVs flying in uncontrolled airspaces may encounter other aircraft travelling at high speed, potentially leading to a head-on collision in a worst-case scenario. This motivates UAVs to possess a "sense-and-avoid" capability, for which they must identify (and/or estimate) potential obstacles in their environments and proceed to safely navigate away from them while completing missions. This paper outlines an approach that, by transforming the problem of collision avoidance into one of waypoint tracking, can be utilized by UAVs operating in both the former and latter types of situation.

In particular, this work is motivated by the need for UAVs flying in uncontrolled airspaces to avoid collision while meeting international and/or government restrictions and guidelines for doing so, for instance the "remain well-clear" criteria established by international ASTM standards [1]. To do so, a "collision cone" method introduced in the context of robotics by Chakravarthy and Ghose [2] and applied to UAV collision avoidance by Han and Bang [3] and Watanabe et al. [4, 5] is expanded in this work. Photorealistic and physically accurate simulation of such a scenario is used to demonstrate the feasibility of the developed approach to meet the aforementioned goal.

Related work in this area includes a study by Smith and Harmon [6], in which collision cones formed in an obstacle-rich environment were "aggregated" to enable avoidance of collision with multiple obstacles simultaneously. Park and Kim [7] examined the use of a stereo camera onboard a quadrotor for sensing of static obstacles followed by a collision cone approach for avoidance. Additionally, Lin et al. [8] introduced a "fast geometric avoidance" algorithm that, through calculation of the optimal time to begin a maneuver, performs collision avoidance in cases of both static and dynamic obstacles with strong improvements in computation time.

In Section 2, a proportional navigation (PN) guidance law for intercept/rendezvous is derived from first principles of optimal control. Section 3 introduces the proposed approach for collision avoidance using the "safety ellipsoid" approach to assess obstacle criticality and generate waypoints for safe collision avoidance to be reached via PN guidance. Finally, in Section 4, basic MATLAB simulation of multiple obstacle avoidance scenarios, followed by image-in-the-loop simulation in Microsoft AirSim [9] of head-on collision scenarios, both support the utility of the approach.

2 Optimal Guidance

2.1 Guidance Problem

Consider an aircraft A flying in one direction towards a stationary target F. Let the state of the aircraft at time t be represented as a vector of its position and velocity relative to that of the target F: $\mathbf{x} = \begin{bmatrix} p & v \end{bmatrix}^\mathsf{T} = \begin{bmatrix} p_A - p_F & v_A \end{bmatrix}^\mathsf{T}$. The equations of motion may then be written in state-space form as a double integrator in Eq. (1), where the control u is simply the aircraft's commanded acceleration a. Given a fixed terminal time t_f, the cost function can be written as Eq. (2), a sum of terminal state cost and total control cost over the horizon. In the terminal state cost, the weights m and n, both ≥ 0, are the penalties on the final relative position and velocity respectively.

- Equations of Motion (Kinematics):

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\ \begin{bmatrix} v \\ a \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \end{aligned} \tag{1}$$

- Cost Function:

$$J = \Phi(\mathbf{x}|_{t=t_f}, t_f) + \int_{t_0}^{t_f} L(\mathbf{x}, u, t)\, dt = \frac{1}{2}\left(\mathbf{x}|_{t=t_f}^\mathsf{T} \begin{bmatrix} m & 0 \\ 0 & n \end{bmatrix} \mathbf{x}|_{t=t_f} + \int_{t_0}^{t_f} u^2\, dt \right) \tag{2}$$

The cost function may be augmented using the Lagrange multipliers (costates) $\boldsymbol{\lambda}$, where λ_p and λ_v correspond to the position and velocity respectively. The Hamiltonian H may be introduced as in Eq. (3), where $\mathbf{f} = \mathbf{x}$. The necessary conditions for the optimal control problem are hence given by the following:

$$H = L + \boldsymbol{\lambda}^\mathsf{T}\mathbf{f} = \frac{1}{2}u^2 + \begin{bmatrix} \lambda_p & \lambda_v \end{bmatrix} (\mathbf{A}\mathbf{x} + \mathbf{B}u) \tag{3}$$

- Costate Evolution:

$$\begin{aligned} \dot{\boldsymbol{\lambda}} &= -\frac{\partial H}{\partial \mathbf{x}} = -\mathbf{A}^\mathsf{T}\boldsymbol{\lambda} \\ \begin{bmatrix} \dot{\lambda}_p \\ \dot{\lambda}_v \end{bmatrix} &= -\begin{bmatrix} 0 \\ \lambda_p \end{bmatrix} \end{aligned} \tag{4}$$

- State Evolution:

$$\dot{\mathbf{x}} = \frac{\partial H}{\partial \boldsymbol{\lambda}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \tag{5}$$

- Stationarity Condition:

$$0 = -\frac{\partial H}{\partial u} = u + \lambda_v \tag{6}$$

- Origin Boundary Conditions:

$$\mathbf{x}|_{t=t_0} = \mathbf{x}_0 = \begin{bmatrix} p_0 \\ v_0 \end{bmatrix} \tag{7}$$

- Terminal Costate Conditions:

$$\begin{aligned} \mathbf{0} &= \left(\frac{\partial \Phi}{\partial \mathbf{x}} - \boldsymbol{\lambda} \right)\Big|_{t=t_f} \\ &= \left(\begin{bmatrix} m & 0 \\ 0 & n \end{bmatrix} \mathbf{x} - \boldsymbol{\lambda} \right)\Big|_{t=t_f} \\ &= \left(\begin{bmatrix} mp - \lambda_p \\ nv - \lambda_v \end{bmatrix} \right)\Big|_{t=t_f} \end{aligned} \tag{8}$$

2.2 Optimal Controller

These conditions produce a two-point boundary value problem solved by integrating the costates backward in time and the states forward in time. It is now convenient to introduce the "time-to-go" $t_{go} = t_f - t$, which represents the running time remaining for the aircraft to arrive at the target. Integrating

Eq. (4) from time t_0 to t_f gives a constant position costate and a linear velocity costate:

$$\lambda_p(t) = c_p \tag{9}$$
$$\lambda_v(t) = \lambda_v(t_f) + t_{go}\lambda_p \tag{10}$$

For simplicity, it is temporarily assumed that $t_0 = 0$. The stationarity condition, Eq. (6), gives the controller $u = -\lambda_v$, which along with the origin conditions, Eq. (7), may then be used to integrate the equations of motion, Eq. (1), from $t = 0$:

$$p(t) = p_0 + tv_0 - \frac{1}{2}t^2\left(\lambda_v(t_f) + t_f\lambda_p\right) + \frac{1}{6}t^3\lambda_p \tag{11}$$
$$v(t) = v_0 - t\left(\lambda_v(t_f) + t_f\lambda_p\right) + \frac{1}{2}t^2\lambda_p \tag{12}$$

The terminal conditions, Eq. (8), are used to find the final costate values:

$$\lambda_p = mp|_{t=t_f} = m\left(p_0 + t_f v_0 - \frac{1}{2}t_f^2\left(\lambda_v(t_f) + t_f\lambda_p\right) + \frac{1}{6}t_f^3\lambda_p\right) \tag{13}$$
$$\lambda_v(t_f) = nv|_{t=t_f} = n\left(v_0 - t_f\left(\lambda_v(t_f) + t_f\lambda_p\right) + \frac{1}{2}t_f^2\lambda_p\right) \tag{14}$$

Eqs. (13) and (14) may be expressed as a matrix multiplication acting on the initial state x_0:

$$\begin{bmatrix}\lambda_p \\ \lambda_v(t_f)\end{bmatrix} = G(t_f)^{-1}\begin{bmatrix}t_f + \frac{1}{n} & t_f\left(\frac{1}{n} + \frac{1}{2}t_f\right) \\ -\frac{1}{2}t_f^2 & \frac{1}{m} - \frac{1}{6}t_f^3\end{bmatrix} \tag{15}$$
$$G(q) = \left(\frac{1}{3}q^3 + \frac{1}{m}\right)\left(q + \frac{1}{n}\right) - \frac{1}{4}q^4 = \frac{1}{mn} + \frac{1}{m}q + \frac{1}{3n}q^3 - \frac{1}{12}q^4$$

Now the assumption that $t_0 = 0$ is relaxed, meaning that t_{go} must replace t_f in the formulation in Eq. (15). Furthermore, since any state along the optimal path to the target may be taken to be the original state, the initial state may be replaced with the current state in Eq. (15) as well. As Lewis et al. [10] note, this denotes optimization of $J(t)$, the cost *remaining* along the path from $[t, t_f]$. Hence, the final costate is now expressed as a matrix multiplication acting on the current state x:

$$\begin{bmatrix}\lambda_p \\ \lambda_v(t_f)\end{bmatrix} = G(t_{go})^{-1}\begin{bmatrix}t_{go} + \frac{1}{n} & t_{go}\left(\frac{1}{n} + \frac{1}{2}t_{go}\right) \\ -\frac{1}{2}t_{go}^2 & \frac{1}{m} - \frac{1}{6}t_{go}^3\end{bmatrix} \tag{16}$$

This yields the optimal controller u^*, which is negative state feedback with gains K_p and K_v:

$$u^*(t) = -\begin{bmatrix}t_{go} & 1\end{bmatrix}\begin{bmatrix}\lambda_p \\ \lambda_v(t_f)\end{bmatrix} = -K_p p(t) - K_v v(t) \tag{17}$$
$$K_p = G(t_{go})^{-1}\left(\frac{t_{go}}{n} + \frac{t_{go}^2}{2}\right) \tag{18}$$
$$K_v = G(t_{go})^{-1}\left(\frac{1}{m} + \frac{t_{go}^2}{n} + \frac{t_{go}^3}{3}\right) \tag{19}$$

If it is only desired that the terminal relative position ("miss distance") $p(t_f)$ be driven to zero without any penalty on terminal relative velocity $v(t_f)$ — that is, an "intercept" — then the corresponding gains may be determined by taking the limits as m approaches ∞ and n approaches 0. This results in a "proportional

navigation"[1] (PN) guidance law:

$$u^*_{PN} = -K_{p,PN}p - K_{v,PN}v \tag{20}$$
$$K_{p,PN} = \frac{3}{t_{go}^2} \qquad K_{v,PN} = \frac{3}{t_{go}}$$

On the other hand, if a "rendezvous" at the target is desired with zero miss distance *and* zero terminal relative velocity, then the corresponding gains may be determined by taking the limits as both m and n approach ∞. This results in an "optimal rendezvous" (OR) guidance law:

$$u^*_{OR} = -K_{p,OR}p - K_{v,OR}v \tag{21}$$
$$K_{p,OR} = \frac{6}{t_{go}^2} \qquad K_{v,OR} = \frac{4}{t_{go}}$$

2.3 3D Guidance

The solution derived in Section 2.2 can be extended to three dimensions (3D) easily assuming kinematics in each direction are unrelated. In particular, suppose an aircraft moves in three dimensions with the kinematics of Eq. (1) in each distinct direction (x, y, and z). Let the speed of the aircraft in the forward (x) direction, $v_{x,A}$, be constant (i.e., uncontrolled). It is desired that the aircraft intercept or rendezvous with a stationary target at $\mathbf{p}_F = \begin{bmatrix} x_F & y_F & z_F \end{bmatrix}^\mathsf{T}$. Since $v_{x,A}$ is constant, the time-to-go t_{go} can be found by interpolation between x_A and x_F. The problem is summarised in Eq. (22):

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$
$$\begin{bmatrix} \dot{x}_A \\ \dot{y}_A \\ \dot{z}_A \\ \dot{v}_{x,A} \\ \dot{v}_{y,A} \\ \dot{v}_{z,A} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_A \\ y_A \\ z_A \\ v_{x,A} \\ v_{y,A} \\ v_{z,A} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ a_{y,A} \\ a_{z,A} \end{bmatrix} \tag{22}$$

$$t_{go} = \frac{x_F - x_A}{v_{x,A}} \tag{23}$$

A controller that produces acceleration commands $\mathbf{u}$ to minimize the corresponding 3D version of the cost function in Eq. (2) can be easily found to have a similar form to Eq. (17) through the same solution procedure that was followed above, for both the cases of intercept (PN) and rendezvous (OR). The corresponding PN and OR gains respectively remain the same.

$$\mathbf{u}^* = K_p \begin{bmatrix} 0 \\ y_F - y_A \\ z_F - z_A \end{bmatrix} + K_v \begin{bmatrix} 0 \\ -v_{y,A} \\ -v_{z,A} \end{bmatrix} \tag{24}$$

[1]The term "proportional" is used because it can be shown that the optimal controller is proportional to the (small) line of sight angle between the aircraft and the target in a two-dimensional problem.

3 Collision Avoidance

Now suppose the aircraft A must travel to a target but avoid several static and/or dynamic obstacles in its path along the way. In particular, the aircraft must keep a certain minimum distance away from each obstacle in its path. To solve this problem, previous work by Han and Bang [3] and Watanabe et al. [4, 5] suggested a "collision cone" approach for assessing collision criticality, coupled with a sequential proportional navigation (SPN) guidance method for avoidance maneuvers.

3.1 Collision Cone

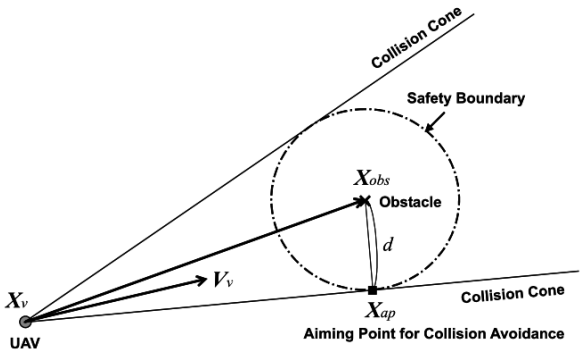


Fig. 1 The geometry of the collision cone approach [4].

In the original formulation of the collision assessment step [3–5], a "safety sphere" with radius of the given minimum miss distance d_{safe} is constructed with its center at the estimated position of each (static) obstacle O in view. Consider the intersection of this safety sphere with the plane formed by (1) the relative position vector to the obstacle, $\mathbf{p}_{\text{rel}} = \mathbf{p}_{\text{obs}} - \mathbf{p}_A$ and (2) the relative velocity vector $\mathbf{v}_{\text{rel}} = \mathbf{v}_A - \mathbf{v}_{\text{obs}}$. As shown in Fig. 1, this intersection forms a circle with radius equal to the miss distance. The surface of the "collision cone" is tangential to this circle at two points[2], $\mathbf{p}_{\text{ap}}$. Suppose (1) $\mathbf{v}_{\text{rel}}$ lies inside the collision cone around O, (2) t_{go} to O is less than a certain threshold t_{safe}, *and* (3) t_{go} to O is less than the time left to reach the target itself[3]. If these three criteria are all met, then the obstacle O is "critical" and collision is imminent without a maneuver, so the sequential guidance method is then used to avoid the obstacle.

Once an object O is judged to be critical, one of the two corresponding aiming points $\mathbf{p}_{\text{ap}}$ is temporarily chosen to be the waypoint to which the vehicle should guide itself. As the ownship navigates towards this waypoint, it moves out of the path of the obstacle through a given guidance law, avoiding collision, and the process repeats. If there are multiple critical obstacles, the obstacle with the least t_{go} is chosen to avoid first. In the absence of any critical obstacle, the ownship sets the target as the waypoint. The necessary control input for each maneuver is calculated at each timestep using the PN or OR guidance law. This method, which utilizes the PN or OR guidance "in sequence" as critical obstacles are encountered, is termed "sequential proportional navigation" (SPN) or "sequential optimal rendezvous" (SOR) guidance.

3.2 Safety Ellipsoid

However, since the collision cone approach constructs a sphere around the estimated position of each obstacle, it only allows for a single miss distance in all directions around each obstacle. This may be suboptimal, as using a constant value in all directions around the obstacle may overestimate the required miss distance in some directions while underestimating the miss distance in others. For example, the ASTM International "remain well-clear" (RWC) requirement for collision hazard avoidance [1, 11] demands a miss distance of 2000 ft in the horizontal (x, y) directions but only 250 ft in the vertical (z) direction. To accommodate this criterion, the collision cone approach would require a uniform miss

[2]The exact coordinates of these two points are known in closed form through projective geometry [4].

[3]If the mission will be completed before the obstacle is encountered, there is no need to complete a maneuver.

distance of 2000 ft in *all* directions, which is likely to lead to overcompensation in the vertical maneuver necessary to avoid the obstacle.

To account for these guidelines, we propose a modification of the collision cone approach referred to as the "safety ellipsoid" approach. Rather than constructing a sphere around the obstacle, an *ellipsoid* with different radii in each direction ($d_{\text{safe,x}}$, $d_{\text{safe,y}}$, and $d_{\text{safe,z}}$) is constructed. Such an approach will permit criteria like the ASTM RWC. The principle of the collision cone remains the same in such a case — the definition of a critical obstacle does not change, and if the safety ellipsoid method assesses the obstacle to be critical, an aiming point is used as a temporary waypoint.

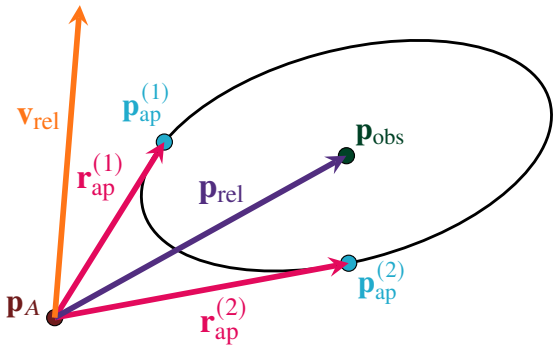


Fig. 2 The geometry of the safety ellipsoid approach. The plane on the page is the plane spanned by (1) the relative position vector $\mathbf{p}_{\text{rel}}$ and (2) the relative velocity vector $\mathbf{v}_{\text{rel}}$.

The geometry of the safety ellipsoid approach is outlined in Fig. 2. Compared to the collision cone[4], the only major difference is that the cross-section of the safety ellipsoid formed by its intersection of the relative position-relative velocity plane is now an ellipse[5] when viewed from above rather than a circle. The coordinates of the aiming points $\mathbf{p}_{\text{ap}}$ are found in closed-form through projective geometry using homogeneous coordinates [13].

$$\mathbf{D} = \text{diag}(d_{\text{safe,x}}^{-2}, d_{\text{safe,y}}^{-2}, d_{\text{safe,z}}^{-2}) \tag{25}$$

$$\mathbf{C} = \begin{bmatrix} C_{11} & C_{12} \\ C_{12} & C_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{\text{rel}}^{\mathsf{T}}\mathbf{D}\mathbf{p}_{\text{rel}} & \mathbf{v}_{\text{rel}}^{\mathsf{T}}\mathbf{D}\mathbf{p}_{\text{rel}} \\ \mathbf{p}_{\text{rel}}^{\mathsf{T}}\mathbf{D}\mathbf{v}_{\text{rel}} & \mathbf{v}_{\text{rel}}^{\mathsf{T}}\mathbf{D}\mathbf{v}_{\text{rel}} \end{bmatrix} \tag{26}$$

$$K = \sqrt{\det C} \tag{27}$$

$$\mathbf{r}_{\text{ap}} = \left(\frac{KC_{11} - K \pm C_{12}\sqrt{C_{11} - 1}}{KC_{11}} \right) \mathbf{p}_{\text{rel}} \mp \left(\frac{\sqrt{C_{11} - 1}}{K} \right) \mathbf{v}_{\text{rel}} \tag{28}$$

$$\mathbf{p}_{\text{ap}} = \mathbf{p}_{own} + \mathbf{r}_{\text{ap}} \tag{29}$$

Since two aiming points are returned by the safety ellipsoid approach[6], if it is assessed that a maneuver is required, the aiming point i for which $\mathbf{r}_{\text{ap}}^{(i)}$ forms a smaller angle with $\mathbf{v}_{\text{rel}}$ is chosen as the waypoint. The vehicle is already moving towards that point before the maneuver begins, making the maneuver smoother. Also, it should be noted that when the safety ellipsoid approach is implemented, it is useful to overestimate $\mathbf{d}_{\text{safe}}$ by a small factor of roughly 1-2% to safeguard against noise in measurements of $\mathbf{p}_{\text{obs}}$ or $\mathbf{v}_{\text{obs}}$ causing singularities during calculation of the aiming point locations.

In this work, we consider the guidance generation to be frequent enough that a moving but non-accelerating obstacle (e.g., an intruder aircraft flying towards the ownship) can be assumed fixed for the purpose of collision avoidance using the safety ellipsoid approach with SPN/SOR guidance. That is, the time between consecutive construction of safety ellipsoids to determine whether an object is critical is small enough that the dynamic obstacle has not translated by a significant amount in the gap.

[4]Note that Fig. 1 is in fact a specific case of the safety ellipsoid geometry in which $d_{\text{safe,x}} = d_{\text{safe,y}} = d_{\text{safe,z}}$.

[5]The lengths of the major and minor axes of this cross-sectional ellipse are known functions (see Eqs. 45 and 19 in [12], where $d = 0$) of the miss distances in each direction, but are not useful in the collision avoidance.

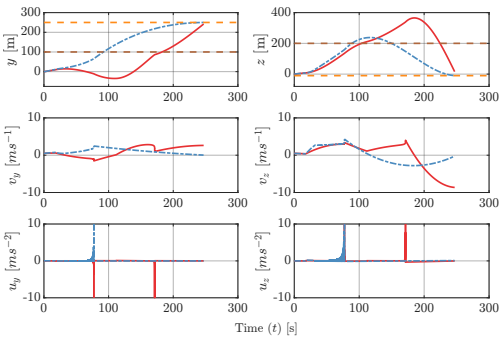
[6]Note that the $\pm$ and $\mp$ in Eq. (28) should be applied at once, i.e., + and − for $\mathbf{r}_{\text{ap}}^{(1)}$, − and + for $\mathbf{r}_{\text{ap}}^{(2)}$.

4 Results

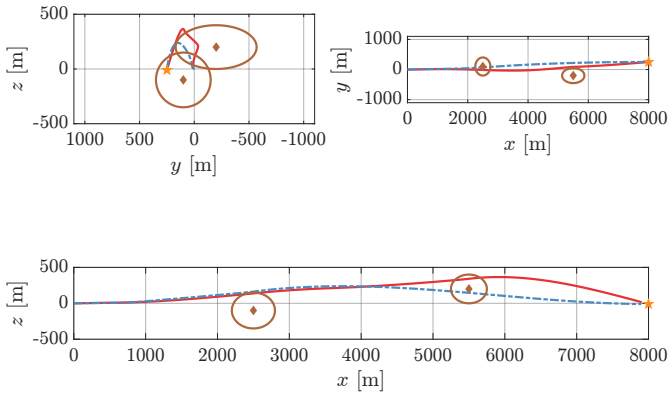
4.1 Demonstration via MATLAB Simulation

The safety ellipsoid collision avoidance approach is first demonstrated in two example scenarios via simulation in MATLAB [14]. Both utilize a simple double integrator model of vehicle state, as shown in Eq. (22), and are simulated discretely at 75 Hz. Measurements of all obstacle positions, with added Gaussian white noise of zero mean and standard deviation $\sigma_p = 5\,\mathrm{m}$, are supplied to the ownship at a frequency of 15 Hz, regardless of whether the obstacles are static or dynamic. In the case of dynamic obstacles, velocities are supplied at the same frequency without noise. The t_{safe} threshold is set to be 60 s for all demonstrations. In both demonstrations, the ownship begins at the origin with a forward (x) speed of $32\,\mathrm{m\,s^{-1}}$ and $0.5\,\mathrm{m\,s^{-1}}$ in the lateral (y) and vertical (z) directions. The destination to which the vehicle must navigate is at $\begin{bmatrix}8000\,\mathrm{m} & 250\,\mathrm{m} & -10\,\mathrm{m}\end{bmatrix}^\mathsf{T}$ Because the destination is 8000 m away in the forward direction, the ownship reaches the destination in 250 s.

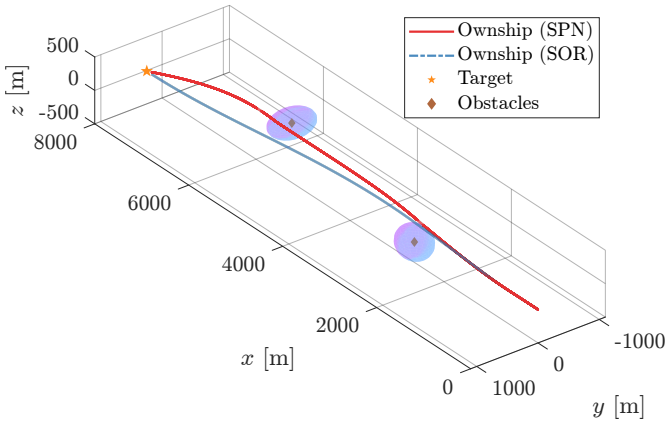
4.1.1 Avoidance of Two Static Obstacles (S)



(a) Time evolution of state and control during the mission.



(b) Trajectories viewed from three planar perspectives.



(c) 3D trajectories.

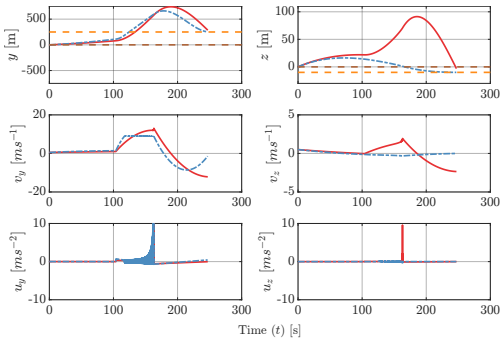
Fig. 3 Results of simulation S.

In the first demonstration (S), the ownship must avoid two static obstacles in its path as it attempts to navigate to the destination. The miss distances differ for each obstacle. For the first obstacle at $\begin{bmatrix}2500\,\mathrm{m} & 100\,\mathrm{m} & -100\,\mathrm{m}\end{bmatrix}^\mathsf{T}$, $\mathbf{d}_{\text{safe}} = \begin{bmatrix}300\,\mathrm{m} & 250\,\mathrm{m} & 250\,\mathrm{m}\end{bmatrix}^\mathsf{T}$. For the second obstacle at $\begin{bmatrix}5500\,\mathrm{m} & -200\,\mathrm{m} & 200\,\mathrm{m}\end{bmatrix}^\mathsf{T}$, $\mathbf{d}_{\text{safe}} = \begin{bmatrix}250\,\mathrm{m} & 370\,\mathrm{m} & 200\,\mathrm{m}\end{bmatrix}^\mathsf{T}$.

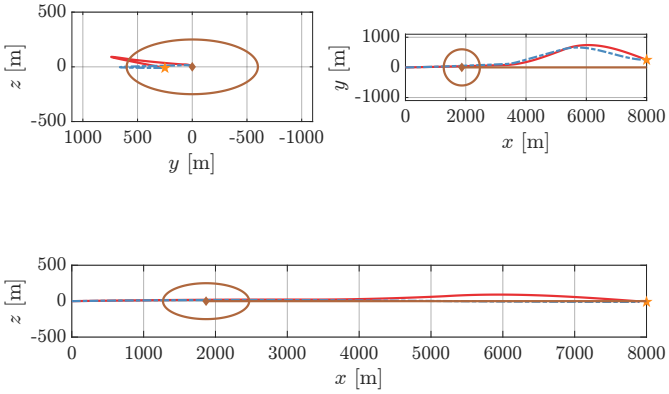
The trajectories the ownship takes in demonstration S are depicted in Fig. 3. Videos of simulation S are available at the following URL, with the suffix "_static": https://zenodo.org/record/6373737 [15].

The ownship succeeds in navigating around each of the two obstacles, both via SPN and SOR guidance. The total cost of the maneuver as calculated via a trapezoidal approximation of the integral in Eq. (2) is 26.211 for the SPN guidance, and a more expensive 63.384 for the SOR guidance, which may be due to the SOR guidance requiring the ownship to decelerate to zero lateral/vertical velocity by the time it reaches the target.

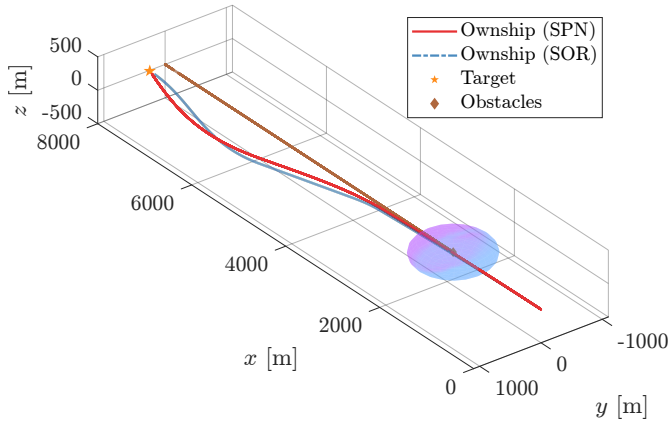
4.1.2 Head-On Encounter with Dynamic Intruder Aircraft (D)



(a) Time evolution of state and control during the mission.



(b) Trajectories viewed from three planar perspectives.



(c) 3D trajectories.

Fig. 4 Results of simulation D.

The second demonstration (D) is a simulation of a realistic head-on encounter with an intruder aircraft. The ownship must avoid a single dynamic intruder flying head-on towards the ownship as it attempts to navigate to the destination. The ownship begins its flight at the origin with the same velocity as in demonstration S. Because the ownship only receives noisy measurements of the obstacle position at a frequency of 15 Hz despite generating guidance commands at 75 Hz, an additional challenge is frequently operating on out-of-date information. The intruder begins at $\begin{bmatrix}12\,000\,\text{m} & 0\,\text{m} & 0\,\text{m}\end{bmatrix}^\mathsf{T}$ and flies at constant velocity $\begin{bmatrix}41\,\text{m}\,\text{s}^{-1} & 0\,\text{m}\,\text{s}^{-1} & 0\,\text{m}\,\text{s}^{-1}\end{bmatrix}^\mathsf{T}$ towards the origin. The miss distance for the intruder is $\mathbf{d}_{\text{safe}} = \begin{bmatrix}600\,\text{m} & 600\,\text{m} & 250\,\text{m}\end{bmatrix}^\mathsf{T}$. These values were chosen for convenient visualization, as well as being broadly similar to the ASTM RWC guidelines mentioned earlier.

The trajectories the ownship takes in demonstration D are depicted in Fig. 4. Videos of simulation D are available at the following link, with the suffix "_dynamic": https://zenodo.org/record/6373737 [15]. The ownship succeeds in avoiding collision with the intruder, both via SPN and SOR guidance, overcoming the lag time in obstacle information along the way. It uses an essentially entirely lateral maneuver to navigate around the intruder as it approaches. The total cost of the maneuver is approximately 30.615 for the SPN guidance and 28.443 for the SOR guidance.

4.2 Demonstration via High-Fidelity Simulation in Microsoft AirSim

(a) The intruder in the environment.

(b) The ownship in the environment.

Fig. 5 Images from simulation in AirSim.

Our approach was also validated through simulation in Microsoft AirSim [9], which is an open-source, photorealistic, high-fidelity flight simulator based on the Unreal Engine gaming engine. The simulations were carried out by UtopiaCompression Corporation through an interface that allows AirSim's Python library with Robot Operating System (ROS), and take place in AirSim's "Western World" environment. These simulations (H and A) are similar to simulation D as described in the previous section: a quadcopter is required to avoid a dynamic intruder aircraft, a small business jet, flying towards it. However, rather than being directly supplied measurements of the obstacle state at fixed intervals in time, the ownship must estimate the state of the obstacle on its own. It accomplishes this through UtopiaCompression's Passive Collision Alert System (PCAS), which is a low-cost detection, tracking, and ranging system for detect-and-avoid (DAA) systems that relies only on a visual (electro-optical or infrared) sensor/lens, an INS/GPS device, and a microprocessor. INS/GPS measurements and images from the virtual on-board camera provided by AirSim are transmitted to PCAS through a Python wrapper and ROS node. PCAS takes this information into its detector, tracker, and range estimator to determine the state of the intruder. Once the intruder is detected, the avoidance algorithm uses the safety ellipsoid + SPN/SOR approach to generate the guidance commands, which are transmitted back to AirSim using the ROS wrapper.

Two simulations were carried out in this manner. In both simulations, the ownship navigates via SPN guidance and begins its flight approximately at the origin with a forward (x) speed of $10\,\mathrm{m\,s^{-1}}$ and zero lateral and vertical speed.

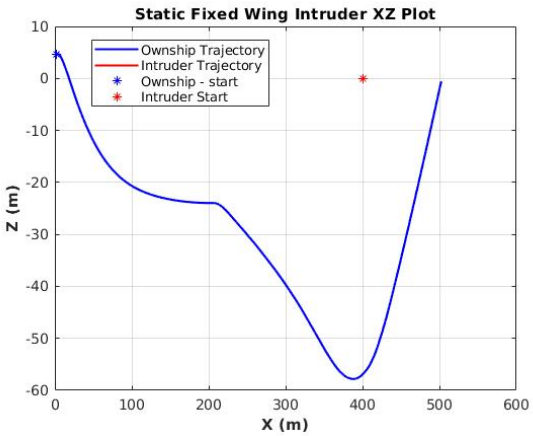
4.2.1 Encounter with Hovering Intruder Aircraft (H)

In simulation H, a static intruder hovers at a point initially 400 m in front of the ownship and does not move as the ownship approaches it. The resulting mission trajectory is displayed in Fig. 6a. The intruder is judged to be a critical obstacle roughly 20 s into the journey, and a maneuver is initiated for avoidance. The ownship moves above the intruder to successfully avoid it and completes the mission by reaching the destination 500 m ahead of the origin.

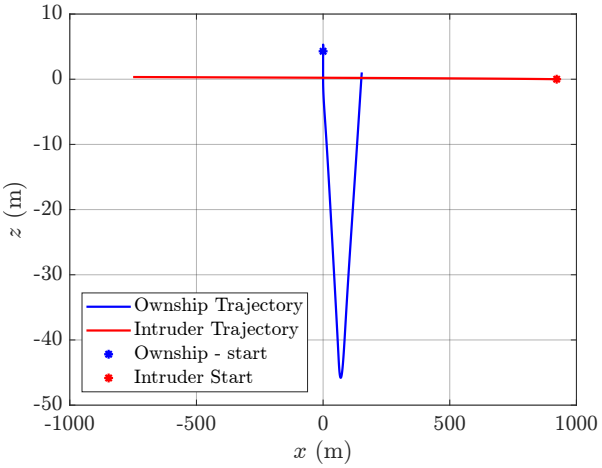
4.2.2 Head-On Encounter with Dynamic Intruder Aircraft (A)

In simulation A, a dynamic intruder aircraft approaches the ownship head-on. The intruder, initially 1.5 km away, approaches at a forward speed of $85\,\mathrm{m\,s^{-1}}$. The resulting mission trajectory is displayed in Fig. 6b. When the ownship detects the intruder, the collision maneuver begins and the ownship moves

above the intruder to avoid it. Because PCAS happens to only recognize the intruder and estimate its position accurately when it is relatively close, the most efficient maneuver turns out to be in the direction with the least miss distance, which is vertical. The ownship accordingly maneuvers above the intruder to avoid it. After avoiding the intruder, the ownship returns to its original trajectory to reach the destination 200 m ahead of the origin.



(a) Result of simulation H.



(b) Result of simulation A.

Fig. 6 Images from simulation in AirSim. A negative value of z indicates an upward location.

5 Conclusion

This work presents a method for navigation and collision avoidance in obstacle-rich 3D environments by transforming the problem into one of waypoint tracking. The approach, which consists of navigating to waypoints that satisfy miss distance requirements, is capable of navigating UAVs away from both static and dynamic obstacles with minimal control effort. Of particular interest is that this approach accommodates different miss distances in each direction around the obstacles, which permits more optimal maneuvers and also satisfies previously unmet international UAV safety guidelines. The approach was first demonstrated via basic simulation in MATLAB. High-fidelity simulation in AirSim involving visual estimation of obstacle position via a collision alert system, namely UtopiaCompression's PCAS, further validated the approach. Future work may consider incorporating an estimate of the probability of collision with each obstacle into the cost function, or use it for determination of whether an obstacle is "critical." Alternatively, the covariance and mean of the current obstacle state estimate may inform the cost function as well.

Acknowledgments

The research and development effort presented in this paper has been sponsored by AFWERX under a Small Business Technology Transfer (STTR) Program Phase I (contract number FA8649-21-P-0184) awarded to UtopiaCompression Corporation. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

References

[1] Standard specification for detect and avoid system performance requirements. DOI: 10.1520/f3442_f3442m-20, https://doi.org/10.1520/f3442_f3442m-20.

[2] A. Chakravarthy and D. 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. DOI: 10.1109/3468.709600.

[3] Su-Cheol Han, Hyochoong Bang, and Chang-Sun Yoo. Proportional navigation-based collision avoidance for UAVs. *International Journal of Control, Automation and Systems*, 7(4):553–565, Aug. 2009. DOI: 10.1007/s12555-009-0407-1.

[4] Yoko Watanabe, Anthony Calise, Eric Johnson, and Johnny Evers. Minimum-effort guidance for vision-based collision avoidance. In *AIAA Atmospheric Flight Mechanics Conference and Exhibit*. American Institute of Aeronautics and Astronautics, June 2006. DOI: 10.2514/6.2006-6641.

[5] Yoko Watanabe, Anthony Calise, and Eric Johnson. Vision-based obstacle avoidance for UAVs. In *AIAA Guidance, Navigation and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, June 2007. DOI: 10.2514/6.2007-6829.

[6] Austin L. Smith and Frederick G. Harmon. UAS collision avoidance algorithm based on an aggregate collision cone approach. *Journal of Aerospace Engineering*, 24(4):463–477, Oct. 2011. DOI: 10.1061/(asce)as.1943-5525.0000081.

[7] Jongho Park and Youdan Kim. Stereo vision based collision avoidance of quadrotor uav. In *2012 12th International Conference on Control, Automation and Systems*, pages 173–178, 2012.

[8] Zijie Lin, Lina Castano, Edward Mortimer, and Huan Xu. Fast 3d collision avoidance algorithm for fixed wing UAS. *Journal of Intelligent & Robotic Systems*, 97(3-4):577–604, June 2019. DOI: 10.1007/s10846-019-01037-7.

[9] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[10] Frank L. Lewis, Draguna Vrabie, and Vassilis L. Syrmos. *Optimal Control, 3rd Edition*. John Wiley and Sons, 2012.

[11] Andrew Weinert, Scot Campbell, Adan Vela, Dieter Schuldt, and Joel Kurucar. Well-clear recommendation for small unmanned aircraft systems based on unmitigated collision risk. *Journal of Air Transportation*, 26(3):113–122, July 2018. DOI: 10.2514/1.d0091.

[12] Peter Paul Klein. On the ellipsoid and plane intersection equation. *Applied Mathematics*, 03(11):1634–1640, 2012. DOI: 10.4236/am.2012.311226.

[13] John Alexiou (https://math.stackexchange.com/users/3301/john-alexiou). Finding a point on an ellipsoid where the normal is perpendicular to a line segment. Mathematics Stack Exchange. https://math.stackexchange.com/q/1784014.

[14] The Mathworks, Inc., Natick, Massachusetts. *MATLAB R2021b*, 2021.

[15] European Organization For Nuclear Research and OpenAIRE. Zenodo, 2013. DOI: 10.25495/7GXK-RD71, https://www.zenodo.org/.