
Proceedings Paper:
Douthwaite, J.A., De Freitas, A. and Mihaylova, L.S. (2017) An Interval Approach to Multiple Unmanned Aerial Vehicle Collision Avoidance. In: Sensor Data Fusion: Trends, Solutions, Applications (SDF), 2017. 11th Symposium Sensor Data Fusion: Trends, Solutions, and Applications, 10-12 Oct 2017, Bonn, Germany. IEEE.

https://doi.org/10.1109/SDF.2017.8126384

An Interval Approach to Multiple Unmanned Aerial Vehicle Collision Avoidance

James A. Douthwaite
Dept. of Automatic Control
& Systems Engineering
University of Sheffield
jadouthwaite1@sheffield.ac.uk

Allan De Freitas
Dept. of Automatic Control
& Systems Engineering
University of Sheffield
a.defreitas@sheffield.ac.uk

Lyudmila S. Mihaylova
Dept. of Automatic Control
& Systems Engineering
University of Sheffield
l.s.mihaylova@sheffield.ac.uk

***Abstract*—Small/micro Unmanned Aerial Systems (UAVs) require the ability to operate with constraints of a diverse, automated airspace where obstacle telemetry is denied. This paper proposes a novel Sense, Detect and Avoid (SDA) algorithm with inherit resilience to sensor uncertainty. This is achieved through the interval geometric formulation of the avoidance problem, which by the use of interval analysis, can be extended to consider multiple obstacles. The approach is shown to demonstrate the ability to both tolerate sensor uncertainty and enact generated 3D avoidance trajectories. Monte-Carlo simulations demonstrate successful avoidance rates of 88%, 96% and 91% in two example collision scenarios and one multi-agent conflict scenario respectively.**

Keywords*—Collision avoidance, unmanned aerial systems, interval analysis, sensor uncertainty, interval geometry.

I. Introduction

Multi-agent coordination and formation control are subjects that has seen an increasing interest within the robotic systems community. Systems operating as a collective, or swarm, are capable of reaching higher levels of performance, reliability and redundancy in the same tasks of any individual agent. However, the coordination of multiple agents also presents challenges in communication, path planning and collision avoidance. In situations where the dynamics of the agents are fast, such as the case of aerial systems, preventing collision is essential in ensuring their survival.

Multi-agent collision avoidance, in a broader sense, can be seen analogous to that of modern air-traffic control. The need for more sophisticated automation tools for handling collision scenarios due to increasing conventional traffic is highlighted in [1], [2]. However, with the growing use of small domestic and commercial Unmanned Aerial Systems (UASs) or Unmanned Aerial Vehicles (UAVs), an increasingly populated and diverse automated airspace is inevitable. Ensuring the survival of a UAS in dynamic environments where telemetry from systems such as ADS-B (Automatic Dependent Surveillance - Broadcast) [3] is denied, still poses a significant challenge [4]. For this a number of prerequisites must be met: such as suitable sensing, safety protocols and design requirements. Quantifying a systems ability to cooperate in an autonomous collision scenario is therefore becoming a critical part of any systems integration into the modern airspace [1], [5], [6].

Over the last decade, numerous technological advances in the way of sensing and computational power have allowed increasingly small/cheap UASs to achieve higher levels of autonomy and cooperation. However, tasks such as collision avoidance are still predominantly achieved via higher level path-planning (i.e. non-reactive). Scalability, centralisation, communication and interaction with agents outside of the network are therefore a natural concern in an avoidance scenario. Currently, there are several documented approaches to reactive collision avoidance, namely; potential fields [7], [8], evolutionary algorithms [9], probabilistic [10] and geometric methods related to these works [11]–[17].

This paper presents an interval geometric approach to the Sense Detect and Avoid (SDA) collision avoidance problem. The key contributions presented in this paper are i) The proposed interval approach to the design of a three dimensional *reflexive* avoidance algorithm, ii) based on geometric considerations it is demonstrated how sensor uncertainty affects the escape trajectory design, iii) The approach is generalised to a formation of UAVs and it is shown how it can cope with multiple dynamic obstacles in the presence of sensor uncertainty. iv) The approach is evaluated over three example scenarios and results for its performance are reported.

The structure of the paper is as follows; Section II introduces the problem context and uncertainty perspective, Section III describes the construction of the interval geometric problem in the avoidance of one and multiple dynamic obstacles. In Section IV we introduce the agent dynamics and control methodology, Section V presents the algorithm in several simulated scenarios with discussion and concluding remarks appearing in Section VI.

II. Problem Description

We consider the scenario where two agents, referred to as *A* and *B*, are moving through a three dimensional (3D) Cartesian space with global velocities $v_a \in \mathbb{R}^{3\times1}$ and $v_b \in \mathbb{R}^{3\times1}$ and representative radii r_a and r_b, respectively (see Figure 1). From the perspective of A, B is an obstacle to be avoided. If both agent trajectories are maintained, a collision will occur at a certain time in the future.

A. Sensor Model

It is assumed that an obstacles position and width can be measured by the agents on-board camera and range-finder.

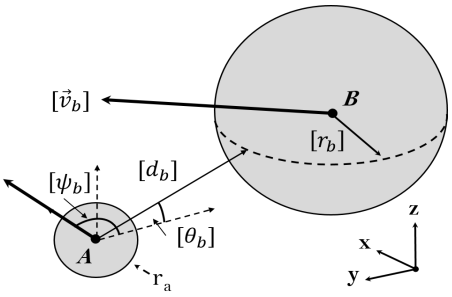

Fig. 1. A typical avoidance scenario where agent *A* is tasked with avoiding obstacle *B* with arbitrary scalar radius r_b.

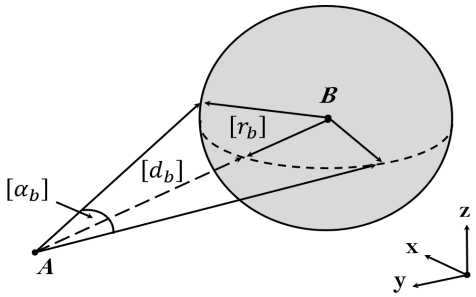

Fig. 2. Definition of the characteristic radius interval $[r_b]$ from its angular width and proximity intervals; $[\alpha_b]$ and $[d_b]$, respectively.

The pixel location of the obstacle provides estimates of its relative elevation $\theta_b \in \mathbb{R}^1$, azimuth angle $\phi_b \in \mathbb{R}^1$ and angular width $\alpha_b \in \mathbb{R}^1$. The range-finder attains scalar values of proximity $d_b \in \mathbb{R}^1$. Each sensor is known to bring with it its own measurement uncertainty and inaccuracies, assumed to be Gaussian distributed with a zero mean. These measurements are used to construct an estimation of the obstacles spherical position in the agent-local *East-North-Up (ENU)* coordinate frame, as it would be seen by a pilot.

The notion of *interval analysis* [18], [19] is a powerful tool initially introduced to cope with the computer rounding errors and later extended to guaranteed state estimation. In this paper we apply ideas from the interval analysis to collision avoidance. We develop an approach for handling parameter uncertainty without linearisation or approximation. The true value of a state of interest is not known. Instead, it is assumed that it lies within a certain interval $[x] = [\underline{x}, \overline{x}]$. Measurements provided by the UAV's sensors allow an interval to be defined containing all possible values including the true state. By applying the interval arithmetic we are then able to define an interval enclosing the problem solution.

For the purpose of of this paper, maximum measurement uncertainty is defined as $[x] = [x - 3\sigma_x, x + 3\sigma_x]$; where σ_x is the standard deviation of the measurements [19]. The Cartesian position of the obstacle in the UAV's local coordinate frame can then be inferred as the interval vector $[\vec{S}_b]$ via Equation (1).

$$[\vec{S}_b] = \begin{bmatrix} [x_b] \\ [y_b] \\ [z_b] \end{bmatrix} = \begin{bmatrix} \cos([\phi_b]) \cdot \cos([\theta_b]) \\ \sin([\phi_b]) \cdot \cos([\theta_b]) \\ \sin([\theta_b]) \end{bmatrix} \cdot [d_b] \quad (1)$$

The region defining the obstacles range $[d_b]$ and width in the azimuth $\alpha_b \in [\underline{\alpha_b}, \overline{\alpha_b}]$ are then used to discern the limits of the obstacles tangential radius $[r_b]$. This relationship can be seen in Figure 2 and described in Equation (2).

$$\begin{aligned} \sin\left(\frac{1}{2}[\alpha_b]\right) &= \frac{[r_b]}{[d_b] + [r_b]} \\ \sin\left(\frac{1}{2}[\alpha_b]\right).[d_b] &= \left(1 - \sin\left(\frac{1}{2}[\alpha_b]\right)\right) \cdot [r_b] \\ [r_b] &= \frac{sin\left(\frac{1}{2}[\alpha_b]\right)}{1 - \sin\left(\frac{1}{2}[\alpha_b]\right)} \cdot [d_b] \end{aligned} \quad (2)$$

It is otherwise assumed that the agent A has no other prior knowledge of obstacle B. Based on the agents measurements of B, the interval regions containing the obstacles relative Cartesian positions $[\vec{S}_b]$ and radius $[r_b]$ can be derived. Given the uncertainty in its position, estimation of the obstacle velocity $\vec{v}_b = [v_x, v_y, v_z]^T$ naturally takes interval form. This can be seen expressed in Equation (3) as the change in position over the discrete sample time Δt.

$$[\vec{v}_{b,k}] = \frac{1}{\Delta t}\left([\vec{S}_{b,k}] - [\vec{S}_{b,k-1}]\right) \quad (3)$$

The interval bounding the velocity vector of B at the current sample time k can then be estimated through successive samples of the Cartesian position interval $[\vec{S}_b]$. It is also assumed that agent A measures its own velocity with a given uncertainty $[\vec{v}_a]$.

III. Interval Avoidance

It has been demonstrated how estimates of the obstacles relative position $[\vec{S}_b]$, relative velocity $[\vec{v}_b]$ and defining radius $[r_b]$ are achieved. By associating measurement uncertainty with each of these parameters in the form of intervals, we propagate this uncertainty through the interval-geometric avoidance problem as follows.

A. Discerning Likelihood of Collision

The vector interval $[\vec{r}_m]$, termed the *miss interval*, can be seen in Figure 3 as the interval containing the shortest distances between A and obstacle B's current trajectory. The interval enveloping these points is known as the minimal separation or *closest approach* $[r_{ca}]$ interval. It can then be said that given $[\vec{v}_b]$ at some time in the future τ the obstacle will pass through $[r_{ca}]$. The miss vector is related to the obstacles relative velocity $[\vec{v}_b]$ and position $[\vec{S}_b]$ by Equation (4).

$$[\vec{r}_m] = [\vec{v}_b] \times ([\vec{S}_b] \times [\vec{v}_b]) \quad (4)$$

From Equation (4) it is known that $\vec{v}_b$ and $\vec{r}_m$ are orthogonal ($\vec{v}_b \cdot \vec{r}_m = 0$). Through some rearrangement, the time to collision interval $[\tau]$ can be inferred through Equation (5).

$$\begin{aligned} [\vec{r}_m] &= [\vec{S}_b] + [\vec{v}_b] \cdot [\tau] \\ [\vec{v}_b] \cdot [\vec{r}_m] &= [\vec{S}_b] \cdot [\vec{v}_b] + ([\vec{v}_b] \cdot [\vec{v}_b])[\tau] \\ ([\vec{v}_b] \cdot [\vec{v}_b])[\tau] &= [\vec{v}_b] \cdot [\vec{r}_m] - [\vec{S}_b] \cdot [\vec{v}_b] \\ [\tau] &= -\frac{[\vec{S}_b] \cdot [\vec{v}_b]}{[\vec{v}_b] \cdot [\vec{v}_b]} \end{aligned} \quad (5)$$

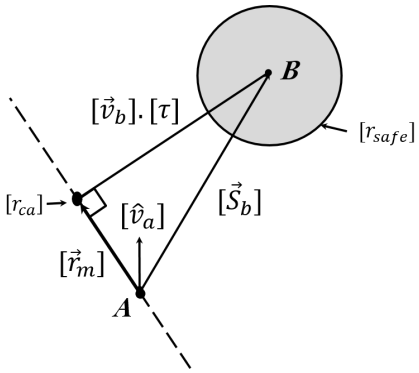

Fig. 3. The miss interval $[\vec{r}_m]$ defined geometrically, in the configuration space of A, from the obstacles relative position $\vec{S}_b$, velocity $\vec{v}_b$ and the time to nearest approach τ.

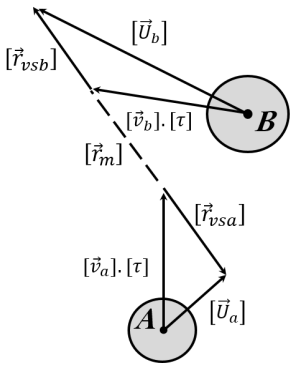

Fig. 4. Definition of the shared resolution intervals $[\vec{r}_{vsa,vsb}]$ and the required correction velocity intervals $[\vec{U}_a]$. Here, the velocities are defined in the plane of avoidance.

From Equation (5) it can be inferred that if the bounds of $[\tau]$ are both positive (i.e $[\tau] > 0$), a collision is likely to occur and the avoidance routine is necessary. Ambiguity occurs when $0 \in [\tau]$. This is because the uncertainty in the obstacle trajectory means that there is a possibility for the collision to occur, but also for it not to occur.

It is clear that if there is even a small possibility of collision the UAV should act to avoid the threat. We can define this condition simply in interval terms; as sign of the supremum $\overline{\tau}$. The avoidance routine should therefore be executed where $\overline{\tau} \geq 0$ is met.

B. Safe Separation

In this section we define the collision condition $||S_b|| \leq (r_a + r_b)$ in terms of the obstacle and agent radii r_a and r_b, respectively. r_a is assumed to be constant known to agent A, while $[r_b]$ is subject to measurement uncertainty as seen in Equation (2). By inclusion of an obstacle safety factor s_f, the minimum safe separation r_{safe} is defined in Equation (6).

$$[r_{safe}] = r_a + s_f * ([r_b]) \tag{6}$$

$$[r_{res}] = [r_{safe}] - |[\vec{r}_m]| > 0 \tag{7}$$

Relating Equation (6) to the miss interval $[\vec{r}_m]$, we define the difference to be the *resolution zone* r_{res} in Equation (7) [15]. A conflict is therefore said to be occurring when the minimum value (infinum) $\underline{r}_{res} \leq 0$; indicating less the sufficient separation at $t = \tau$.

C. Optimal Correction

In the event that $\overline{\tau} > 0$, an optimal manoeuvre must be designed to avert collision with the obstacle. As seen in Figure 3, any avoidance manoeuvre should act to maximise the miss distance interval $[\vec{r}_m]$ at τ (see Equation (9)). The optimal direction of avoidance can be seen represented as the minimal solution to the following Hamiltonian (8).

$$H = -\vec{r}_m \cdot \vec{v}_b - (|A| \cdot \tau_c) \cdot \vec{r}_m \cdot \hat{a} \tag{8}$$

$$\min_a J = -\frac{1}{2}||\vec{r}_m||^2 \tag{9}$$

It can be deduced from the relationship between $\vec{r}_m$ and the acceleration unit vector $\hat{a}$, that H is minimal when $\hat{a} \cdot \vec{r}_m = 0 \therefore \hat{a} \parallel \hat{r}_m$. Under the principle of *Vector Sharing*, we can determine that the optimal course correction is one parallel to the miss interval $[\vec{r}_m]$ [11], [15].

D. Vector Sharing

The magnitude of the interval correction vector required to avoid the obstacle is determined through the process of *Vector Sharing*. We assume that if the obstacle is able, it would act to prevent collision with the agent. Based on the uncertainty in both agent trajectories it is possible to define an expression for the *Shared Seperation Intervals* (10).

$$[\vec{r}_{vsa,vsb}] = \frac{|[\vec{v}_{b,a}]|}{|[\vec{v}_{a,b}]| + |[\vec{v}_{b,a}]|} \frac{[r_{res}]}{|[\vec{r}_m]|} (\mp[\vec{r}_m]) \tag{10}$$

Equation (10) describes the distribution of the correction magnitude $[r_{res}]$ between the two agents from their observed velocities $[\vec{v}_a]$ and $[\vec{v}_b]$. As shown in Figure 4, the agent with the larger velocity is therefore required to exert a larger correction interval vector $[\vec{r}_{vsa}]$ in the optimal direction $\vec{r}_m$.

We aim to define the interval containing the optimal heading vector $\vec{U}_a^* \in [\vec{U}_a]$ given the obstacles trajectory uncertainty. The interval itself can be assembled geometrically by extrapolating the agents current velocity to the time of closest approach, τ, as seen in Figure 4. The resulting avoidance heading interval $[\vec{U}_a]$ for the agent can then be seen expressed in Equation (11).

$$[\vec{U}_a] = [\vec{v}_a] \cdot [\tau] + [\vec{r}_{vsa}] \tag{11}$$

The avoidance interval $[\vec{U}_a]$ defines a region enveloping the resolution vector necessary to optimally avoid obstacle B. The relative elevation and heading angle intervals $[\theta_a]$ and $[\gamma_a]$, respectively, can be assembled geometrically from the relative heading vector $[\hat{U}_a]$. *Note: From this point onwards* $[\vec{U}_a]$ *is redefined as* $[\vec{U}]$ *as all trajectory corrections are instigated by agent* A.

E. Multiple Obstacle Consideration

In the event that there is a collision likelihood for multiple obstacles ($\tau_i > 0$), an interval must be defined containing the globally valid avoidance trajectories $[\vec{U}^*]$. Under the principle of *Interval Analysis* we are able to consider multiple obstacles by defining the *intersection* of their resolution headings (see

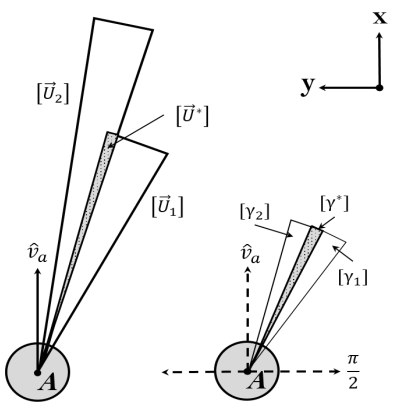

Fig. 5. Determination of the relative heading interval vector $[\vec{U}^*]$ and its associated relative heading angle interval $[\gamma]$ as a result of the intersection process.

Figure 5). The process of multiple obstacle consideration is detailed in Algorithm (1).

To compute the global avoidance set, the intersection of the individual obstacle subsets is used such that $[\vec{U}_a^*] = [\vec{U}_a^*] \cap [\vec{U}_a]_{i=1:n}$ where n is the number of obstacles (see Algorithm 1) [19].

Scenarios may exist where there can be no intersection between avoidance headings (i.e $[\vec{U}_1] \notin [\vec{U}_2]$). In such cases some strategy must be applied in order ensure a valid trajectory interval is available. An immediate solution is to prioritise the avoidance trajectory set based on obstacle proximity $P_{i=1:n} = mid(1/(||\vec{S}_b||_{i=1:n})))$. Here mid is used to define the center of the interval. This ensures that in the worst case, optimal avoidance of the closest obstacle is to be achieved.

As seen in Algorithm 1, first the agent makes a measurement of its own state intervals $[\vec{S}_a]$ and $[\vec{v}_a]$. A sample of the observable objects is then taken from the sensor model to update the obstacles known parameters $[\vec{S}_b], [\vec{v}_b]$ and $[r_b]$. The obstacle set is then reordered based on their priority $P_{i=1:n}$. By iterating over the obstacle set, the algorithm defines each avoidance interval. The intersection procedure then defines final resolution interval heading $[\gamma]$ and elevation $[\theta]$ before handing the requests to the flight controller.

IV. UAV Control & Dynamics

A. Way-point Representation

In addition to the obstacle field $[\vec{U}]_{i=1:n}$, the agents also receive a target way-point defining their desired flight path. Way-points are defined by their relative position $[\vec{S}_w] \in \mathbb{R}^{3\times1}$, velocity $[\vec{v}_w] \in \mathbb{R}^{3\times1}$ and separation tolerance $[r_w] \in \mathbb{R}^1$; assumed obtained by the same sensory mechanism.

Where no obstacle collisions are likely to occur (i.e. $\tau < 0$ is true for all obstacles) the agent is guided towards a designated target heading ($[\hat{S}_w]$) via expressions (12) to (16). Way-points are said to be achieved using the collision condition defined in Equations (6) and (7). *Note: No safety factor is required* $\therefore s_f = 1$ *and* $[r_w]$ *denotes the way-point tolerance.*

Algorithm 1: Calculation of the global optimal resolution region $[\vec{U}^*]$ from the observed obstacle set and the resulting relative $[\theta]$ pitch and $[\gamma]$ heading error intervals.

Data: observationSet, numObstacles
Result: globalControlInputs; θ, γ
// Read the agents local state.
$[\vec{S}_a], [\vec{v}_a], r_a$ = getAgentState()
// Measure the new obstacle states.
obstacleSet = sensorModel(observationSet)
// Prioritise the obstacles.
obstacleSet = sort(obstacleSet.P, descending)
for *i=1:numObstacles* **do**
  // Get the obstacle states.
  $[\vec{S}_b], [\vec{v}_b], [r_b]$ = obstacleSet(i).parameters
  // Compute optimal avoidance trajectory.
  $[\vec{U}_i]$ = computeTrajectory($[\vec{S}_a], [\vec{v}_a], [\vec{S}_b], [\vec{v}_b], [r_b]$)
  // Store the trajectory.
  avoidanceSet = cat(avoidanceSet,$[\vec{U}_i]$)
end
while *validIntersect*($[\vec{U}^*]$) *&* $n \leq$ *numObstacles* **do**
  // Recursively compute the intersection.
  $[\vec{U}^*]$ = intersect($[\vec{U}^*]$,avoidanceSet(n)) n++
end
// Evaluate the control intervals for the resulting intersection.
$[\theta^*], [\gamma^*]$ = computeControlInputs($[\vec{U}^*]$)

B. Agent Trajectory Generation

The agent is tasked with avoiding the obstacles in its visual range, but also in maintaining its defined flight path; specified by way-points. The agent's control inputs are defined from the target heading vector as shown in Figure 5. The agent's equivalent elevation angle $[\theta^*]$ and heading angle $[\gamma^*]$ correction intervals are defined in Equations (12) to (16) from the target interval vector projections:

$$[\vec{S}_V] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot [\hat{S}_w] = \boldsymbol{R}_V \cdot [\hat{S}_w] \tag{12}$$

$$[\theta^*] = \tan^{-1}\left(\frac{[\vec{S}_V]}{|||[\vec{S}_H]|||}\right) \tag{13}$$

$$[\vec{v}_H] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot [\hat{v}_a] = \boldsymbol{R}_H \cdot [\hat{v}_a] \tag{14}$$

$$[\vec{S}_H] = \boldsymbol{R}_H \cdot [\hat{S}_w] \tag{15}$$

$$[\gamma^*] = sign\left(mid\left([\hat{S}_H] \times [\hat{v}_H]\right)\right) \cdot \cos^{-1}\left(\frac{[\hat{v}_H] \cdot [\hat{S}_H]}{|||[\hat{v}_H]|||}\right) \tag{16}$$

Here the projection matrices $\boldsymbol{R}_V$ and $\boldsymbol{R}_H$ are used to map interval components of the unit direction $[\hat{S}_w]$ and unit velocity $[\hat{v}_a]$ onto the vertical and horizontal projection vectors (subtext V and H, respectively). The direction of rotation is found by determining the $sign$ of the rotation axis $[\hat{S}_H] \times [\hat{v}_H]$. Finally, considering the centroid of the elevation and heading angle

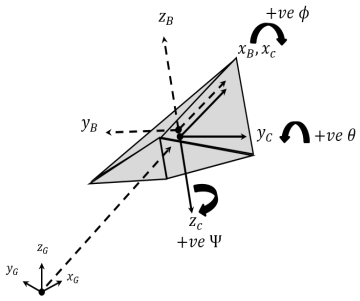

Fig. 6. Definition of the agents local avoidance frame of reference (ENU) and control axes (NED).

intervals such that $\theta = mid([\theta^*])$ and $\gamma = mid([\gamma^*])$ we obtain the best estimates for the inputs UAVs flight controller.

C. Flight Controller

A discrete PD controller was introduced to enact changes to the agents attitude. The flight controller formulation can be seen in Equation (17); generating an angular acceleration vector $\vec{\ddot{\omega}}_c$ in response to a set point attitude error ($e_{\phi,\theta,\psi}$).

$$\vec{\ddot{\omega}}_c = \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix}_c = \begin{bmatrix} 0 & 0 & 0 \\ 0 & k_{p\theta} & 0 \\ 0 & 0 & k_{p\psi} \end{bmatrix} \cdot \begin{bmatrix} e_\phi \\ e_\theta \\ e_\psi \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & k_{d\theta} & 0 \\ 0 & 0 & k_{d\psi} \end{bmatrix} \cdot \begin{bmatrix} \Delta e_\phi \\ \Delta e_\theta \\ \Delta e_\psi \end{bmatrix} \quad (17)$$

The subscript p and d denote the proportional and differential control gains, respectively, while the subscript θ and ψ denote the body axis rotation.

D. Vehicle Dynamics & Constraints

Dependant on whether the unmanned system is fixed or rotary-wing, different dynamic constraints are introduced based on the method of lift generation. For the purposes of this paper, the UAV is modelled as a particle moving with constrained angular acceleration and constant linear velocity.

In Figure 6 the global, body and control axis systems are presented, subtexted as G, B and C, respectively. The obstacle's relative position and velocity are rotated from the global ENU axis system into the body frame of the agent (as seen by the pilot) while the attitude rotations and plant states are represented under the standard North-East-Down (NED) convention.

$$\vec{X} = [x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T \quad (18)$$

$$\vec{X}_{k+1} = f(\vec{X}_k, \vec{\ddot{\omega}}_c) + \vec{w}_k \quad (19)$$

The agents motion is defined by its Cartesian x, y, z and Euler ϕ, θ, ψ positions and velocities $\dot{x}, \dot{y}, \dot{z}$ and $\dot{\phi}, \dot{\theta}, \dot{\psi}$ respectively. The corresponding local NED state vector and state progression can be seen expressed in Equations (18) and (19), respectively.

It is assumed that the UAV is able to generate corrective angular accelerations ($\vec{\ddot{\omega}}_c$) instantaneously. The vehicles motion is however subject to the angular acceleration constraints shown in Table I. These constraints are introduced to emulate saturation of the vehicles actuators.

TABLE I. UAV KINEMATIC CONSTRAINTS.

Control Input	Resulting Acceleration
$\ddot{\phi}_{min} < \ddot{\phi} < \ddot{\phi}_{max}$	$\ddot{\phi}$
$\ddot{\phi} < \ddot{\phi}_{min}, \ddot{\phi} > \ddot{\phi}_{max}$	$\ddot{\phi}_{min}, \ddot{\phi}_{max}$
$\ddot{\theta}_{min} < \ddot{\theta} < \ddot{\theta}_{max}$	$\ddot{\theta}$
$\ddot{\theta} < \ddot{\theta}_{min}, \ddot{\theta} > \ddot{\theta}_{max}$	$\ddot{\theta}_{min}, \ddot{\theta}_{max}$
$\ddot{\psi}_{min} < \ddot{\psi} < \ddot{\psi}_{max}$	$\ddot{\psi}$
$\ddot{\psi} < \ddot{\psi}_{min}, \ddot{\psi} > \ddot{\psi}_{max}$	$\ddot{\psi}_{min}, \ddot{\psi}_{max}$

V. EXPERIMENTAL RESULTS

In the following section, the proposed collision avoidance algorithm is presented in three example scenarios designed to emulate typical aerial encounters. In all cases, the agents are assumed to be small/mini unmanned systems [20], [21]; expected to operate in close proximity ($\leq 500m$) to one another. The experimental conditions used in the following studies are presented in Table II.

TABLE II. GENERAL SIMULATION PARAMETERS USED IN SCENARIOS ONE TO THREE.

Parameter	Value
Sensor Range	$250m$
Camera Standard Deviation	$5.208 \times 10^{-5} rad$
Range-finder Standard Deviation	$0.5m$
Airspeed Standard Deviation	$0.5m/s$
Position Standard Deviation	$0.5m$
Measurement Confidence	3σ
Cruise Speed	$18m/s \approx 40mph$
Agent wing-span	$2.5m$
Simulation frequency	$4Hz$

All UAVs are assumed to have a maximum characteristic size of $r = 2.5m$; the average size of a *mini* class UAV [21] and move at a typical cruise speed of $18m/s \approx 40mph$. The simulation time step is defined as $4Hz$; the repetition frequency of existing Laser Obstacle Avoidance Marconi (LOAM) sensor devices [20].

The agents are initialised with no communicated or prior knowledge of the other aircraft. Trajectory data is collected once the opposing agents come within their designated sensor range (Table II). The collision and way-point achieved conditions are defined in Equation (6). Way-points are given a representative tolerance of a modern civilian Global Position System (GPS) receiver $t_{wp} = 2m$.

A. Overlapping Flightpath Scenario

In the first scenario two agents proceed towards a concentric location, with their target way-point at the opposing side of an intersection. The scenario is defined by the initial global configuration specified in Table III. The scenario is designed to emulate a typical flight path complication with two UAVs operating at the same altitude.

As both Figures 7 and 8 depict, both agents successfully enact conflict resolution manoeuvres in the event of a likely collision. A separation of $8m$ can be see maintained through

TABLE III. INITIALISATION CONDITIONS OF TWO UAVS ON INTERSECTING FLIGHT PATHS.

Initial Condition	UAV 1:alpha001	UAV 2:Beta001
Global Position (m)	[0.00; -250.00; 0.00]	[-176.78; -176.78; 0.00]
Global Velocity (m/s)	[0.00; 18.00; 0.00]	[12.73; 12.73; 0.00]
Global Euler Pose (rad)	[0.00; 0.00; 1.57]	[0.00; 0.00; 0.79]

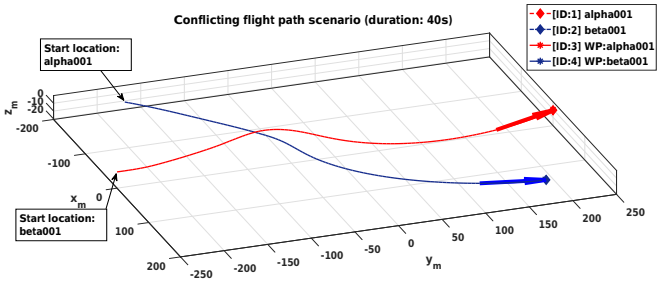

Fig. 7. Example trajectories of two UAVs negotiating conflicting flight paths at the same altitude.

the point of closest approach $[r_{ca}]$. Observing the control inputs around this point we see control oscillations (chatter) prior to the point of closest approach until the second agent responds to the motion of the first agent by enacting an opposing pitch input $\ddot{\theta}_c$ between $t = 6 - 10$. The first agent reacts similarly by adjusting both its pitch and heading angle at $7s$.

Upon completion of the avoidance routine (i.e. $\tau < 0$) which occurs at $t = 10s$, a new heading is quickly attained in order to orient the agent towards their designated way-point (see Figure 9). At $t = 25$ both agents can be seen making final adjustments to their trajectories as they reach the way-point.

B. Head-on Scenario

A direct *Head-On* collision is presented to example the worst case scenario for a *reflexive* collision avoidance algorithm. In such cases there is initially a zero miss distance, with complete ambiguity in the optimal direction of resolution. Both agents are initially on contradicting flight paths, defined by the configuration described in Table IV. The target way-points for each agent is initially behind the opposing agent.

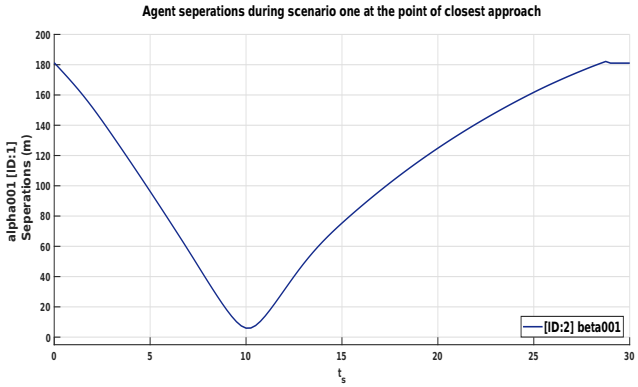

Fig. 8. Agent maintained separation during the conflicting flight path scenario described in Figure 7.

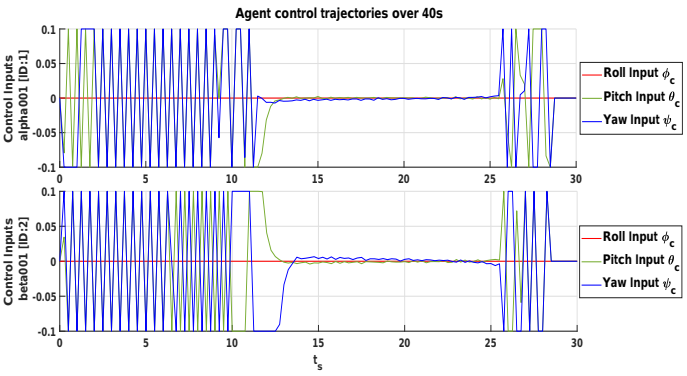

Fig. 9. The control inputs of agent one and two during scenario one.

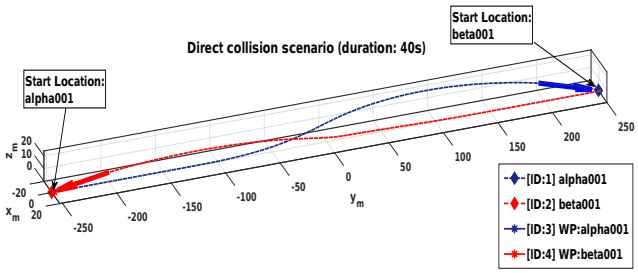

Fig. 10. Resultant trajectories of the two UAV's with directly contradicting flight paths.

Fig. 11. Agent separations during scenario two around the point of closest approach ($t_s = 5 - 25s$)

TABLE IV. INITIALISATION CONDITIONS OF TWO UAVS ON A DIRECT COLLISION COURSE.

	UAV 1:alpha001	UAV 2:Beta001
Global Position (m)	[0.00; -250.00; 0.00]	[0.00; 250.00; 0.00]
Global Velocity (m/s)	[0.00; 18.00; 0.00]	[0.00; -18.00; 0.00]
Global Euler Pose (rad)	[0.00; 0.00; 1.57]	[0.00; 0.00; -1.57]

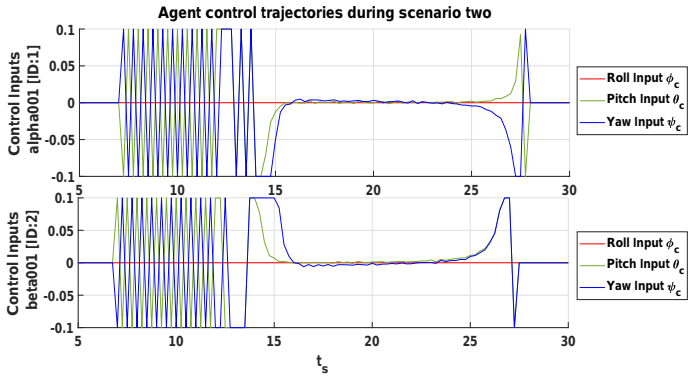

Fig. 12. The control inputs of agent one and two during the direct collision scenario.

The complete agent trajectories in scenario two can again be seen in Figure 10. Here it is shown how both agents were able to generate a non-zero miss distance and negotiate a mutual avoidance manoeuvre. A final separation of $8m$ is then later achieved at the point of closest approach (see Figure 11).

Examining Figure 12, we see both agents begin to generate pitch and yaw inputs upon entering sensor range ($t = 6.5-7s$). The resulting control inputs are shown to be a reflection of the other agent's in both pitch and yaw. This acts to maximise the miss distance until the separation condition is met and the agents pass each other at $t = 14s$. At this point, the way-point guidance then acts to restore the original trajectory; evidenced by the near helical course corrections towards the way-point locations.

C. Multi-Agent Intersection Scenario

The following example scenario presents the algorithm in the context of a three agent, concentric collision. This serves to demonstrate the algorithms ability to generate multiple valid escape trajectories in order to optimally avoid several agents. As Figure 13 shows, the agents are initialised in a ring formation (radius of $250m$) with concentric trajectories and a central point of conflict. The scenario presents a symmetrical problem from each agents perspective.

It can be seen from Figure 14 that in the third scenario the proposed algorithm was able to maintain inter-agent separations of over $2m$ as each agent passes through central conflict zone. The resulting trajectories can be seen in Figure 13; formed from the common heading and pitch intervals satisfying avoidance of the other conflicting agents. This result demonstrates how interval analysis can be exploited to tolerate obstacle trajectory uncertainty and sensor noise in multi-agent conflict resolution scenarios.

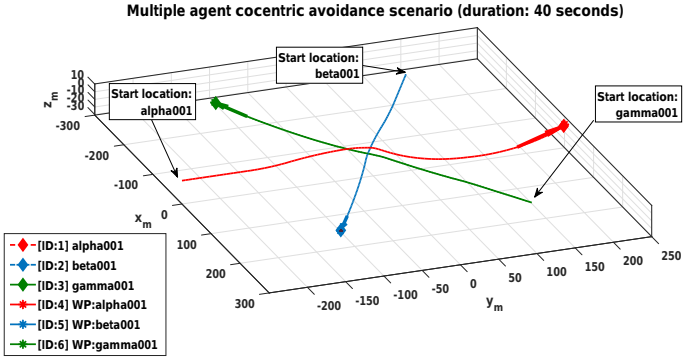

Fig. 13. Resultant conflict aversion trajectories of three homogeneous UAV's with concentric flight paths.

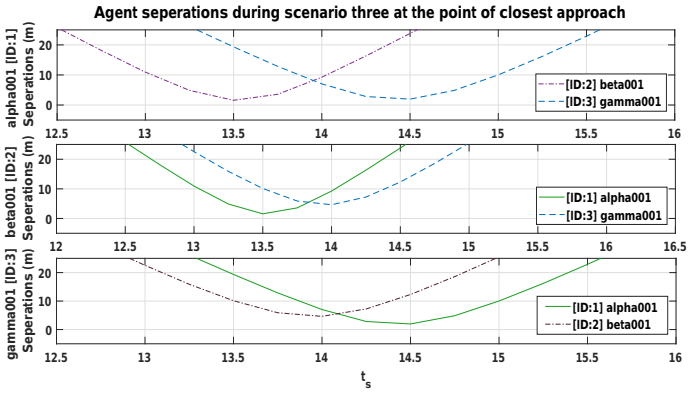

Fig. 14. Agent separations during scenario three around the point of closest approach ($t_s = 12.0s - 16.0s$)

D. Monte-Carlo Evaluation

A Monte-Carlo analysis was undertaken to assess the algorithms rigidity over successive iterations. The agents were initialised in the scenarios presented in Sections V-A to V-C with perturbations applied to their global velocity, heading and position. One way-point is allocated per each agent in each defined scenario. The total number of collisions and way-points achieved across 100 Monte Carlo independent runs are presented in Table V.

TABLE V. MONTE-CARLO SIMULATION RESULTS OF THE THREE PRESENTED CONFLICT SCENARIOS.

Scenario	Flight-path Intersection	Direct Collision	Multi-agent Co-centric
Monte-Carlo Cycles	100	100	100
Collisions	12	4	9
Way-points Attained	181 (200)	199 (200)	298 (300)
Mean Computation Time (ms)	13.2	13.6	24.6

As Table V suggests, in the majority of cases, collision avoidance was successfully achieved by the proposed interval SDA algorithm. This is demonstrated by a collision avoidance success rate in scenario one of 88%, 96% in the *direct collision* scenario and 91% over 100 iterations of the multi-agent case. A way-point success rate of 90.5% was also attained in the first scenario, 99.5% in the second scenario, and 99.3% in the multi-agent scenario.

The mean computational time is shown to be higher in the multi-agent scenario, roughly twice that of the two cases. This likely due to the second iteration of the interval trajectory and intersection region. The computation times for scenarios one and two indicate a difference of 0.4ms between the *Flight-path Intersection* and *Direct Collision* cases.

VI. Conclusions

This paper has introduced the concept of an interval geometric avoidance to a reflexive SDA algorithm. It has been shown how the approach considers measurement uncertainty and noise in the generation of optimal avoidance trajectories in both the singular and multiple agent case. Preliminary simulations indicate promising results in three example scenarios designed to present the agents with ambiguous (symmetric) conflict resolution problems.

In all presented cases conflict resolution was achieved with obstacle trajectory data taken from a simulated camera and range-finder in the presence of their respective measurement uncertainties. Monte Carlo results show that the developed interval collision avoidance approach achieves a high rate of conflict resolution. This presents scope for further investigation with more realistic agent dynamics as to better determine the capabilities of the algorithm on real world systems. The proposed algorithm was also shown able to avoid multiple agent collisions. Consideration into higher numbers of obstacles (agents) and the resulting effect on computation time would also allow the approach to be better assessed for more complex coordination applications.

Acknowledgement

The authors gratefully acknowledge the support from the UK EPSRC under grant number EP/M506618/1.

References

[1] R. Paielli, H. Erzberger, D. Chiu, and K. Heere, "Tactical Conflict Alerting Aid for Air Traffic Controllers," *AIAA Journal of Guidance, Control and Dynamics*, vol. 32, no. 1, pp. 1–19, 2009.

[2] S. John and K. John, "Case Study : Efficient Algorithms for Air Traffic Management Decision Support Tools," in *Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. Bonn: IEEE, 2015, pp. 1–4.

[3] JPDO Aircraft Working Group, "NextGen Avionics Roadmap Version 2.0," Joint Planning and Development Office, Washington DC, Tech. Rep., 2011.

[4] H. Chen, V. Jilkov, and X. Li, "On threshold optimization for aircraft conflict detection," in *Proceedings of the 18th International Conference on Information Fusion (Fusion 2018)*. IEEE, 2015, pp. 1198–1204.

[5] A. Lacher, D. Maroney, and A. Zeitlin, "Unmanned aircraft collision avoidancetechnology assessment and evaluation methods," in *Proceedings of the 7th Air Traffic Managemtent Research & Development Seminar*. Virginia: The MITRE Corporation, 2007, pp. 1–10.

[6] A. Zhahir, A. Razali, and M. Ajir, "Current development of UAV sense and avoid system," in *IOP Conference Series: Materials Science and Engineering*, vol. 152, Universiti Putra Malaysia, 2016.

[7] R. Bamberger, D. Watson, D. Scheidt, and K. Moore, "Flight Demonstrations of Unmanned Aerial Vehicle Swarming Concepts," *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, vol. 27, no. 1, pp. 41–55, 2006.

[8] M. Park, J. Jeon, and M. Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," in *Proceedings of the IEEE International Symposium on Industrial Electronics*, vol. 3, 2001, pp. 1530–1535.

[9] J. Marin, R. Radtke, D. Innis, D. Barr, and A. Schultz, "Using a genetic algorithm to develop rules to guide unmanned aerial vehicles," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, vol. 1, 1999, pp. 1055–1060.

[10] K. Kim, J. Park, and M. Tahk, "UAV Collision Avoidance Using Probabilistic Method in 3-D," in *Proceedings of the International Conference on Control, Automation and Systems*. ICROS, 2007, pp. 826–829.

[11] A. Merz, "Maximum-miss aircraft collision avoidance," *Dynamics and Control*, vol. 1, no. 1, pp. 25–34, 1991.

[12] Z. Fiorini, P.and Shiller, "Motion Planning in Dynamic Environments Using Velocity Obstacles," *The International Journal Of Robotics Research*, no. July 1998, p. 28, 1998.

[13] Z. Shiller, F. Large, S. Sekhavat, and C. Laugier, "Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, 2001, pp. 3716–3721.

[14] J. Berg, M. Lin, and D. Manochag, "Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation," in *IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1928–1935.

[15] J. Park, H. Oh, and M. Tahk, "UAV collision avoidance based on geometric approach," in *Proceedings of the SICE Annual Conference*, 2008, pp. 2122–2126.

[16] D. Wilkie, J. Berg, and D. Manocha, "Generalized Velocity Obstacles," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, North Carolina, 2009, pp. 5573–5578.

[17] J. Berg, J. Snape, S. Guy, and D. Manocha, "Reciprocal Collision Avoidance with Acceleration-Velocity Obstacles," University of North Carolina at Chapel Hill, North Carolina, Tech. Rep., 2011.

[18] G. Alefeld and G. Mayer, "Interval analysis: Theory and applications," *Journal of Computational and Applied Mathematics*, vol. 121, no. 1, pp. 421–464, 2000.

[19] L. Jaulin, "Robust set-membership state estimation ; application to underwater robotics," *Automatica*, vol. 45, no. 1, pp. 202–206, 2009.

[20] R. Sabatini, A. Gardi, and M. Richardson, "LIDAR Obstacle Warning and Avoidance System for Unmanned Aircraft," *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, vol. 8, no. 4, pp. 704–715, 2014.

[21] G. Cai, J. Dias, and L. Seneviratne, "A Survey of Small-Scale Unmanned Aerial Vehicles: Recent Advances and Future Development Trends," *Unmanned Systems*, vol. 2, no. 2, pp. 175–199, 2014.