

# Autonomous Detect and Avoid algorithm respecting airborne Right of Way rules

Anurag Mukherjee<sup>\*</sup> and Sabyasachi Mondal<sup>†</sup> and Antonios Tsourdos<sup>‡</sup>  
*Cranfield University, Cranfield, Bedford, MK43 0AL, UK*

Timothy Bleakley<sup>§</sup>  
*General Atomics Aeronautical Systems Inc, Poway, California, 92064, USA*

**Robust conflict resolution systems are crucial for BVLOS (beyond visual line of sight) operations of UAVs (Unmanned Aerial Vehicles) in the unsegregated airspace. The present conflict resolution research focus is skewed towards optimal path planning, often ignoring the airborne Right-of-way rules prescribed by the FAA and CAA. Although this approach might result in the most optimal path to resolve the conflict, it can cause confusion among other airspace users if the rules of the air are not obeyed when operating in the vicinity of other aircraft. In the present work, a real-time model predictive control approach is proposed that heavily prioritizes adherence to the prescribed right-of-way rules of the air. The manoeuvring limitations of the involved aircraft are also taken into account. Several conflict scenarios were simulated, and the results show that the developed algorithm could resolve all conflicts.**

## I. Nomenclature

|          |   |                                                      |
|----------|---|------------------------------------------------------|
| UAV      | = | Unmanned Aerial Vehicles                             |
| DAA      | = | Detect And Avoid                                     |
| ATC      | = | Air traffic management                               |
| ACAS     | = | Airborne Collision Avoidance System                  |
| DAIDALUS | = | Detect and Avoid Alerting Logic for Unmanned Systems |
| PCA      | = | Point of Closest Approach                            |

## II. Introduction

IN recent years, the aviation industry has witnessed notable technological advancements, particularly in the realm of automation. This shift towards increased automation has given rise to UAVs, which replace traditional pilot functions with automated systems. The widespread utilization of UAVs across various domains presents the challenge of integrating them safely and efficiently into non-segregated airspace without any restrictions on operational conditions. This integration must be achieved while maintaining the existing airspace capacity, ensuring safety, minimizing controller workload, and mitigating risks to other airspace users and ground personnel. The safety of the airspace is taken care of by the DAA systems on UAVs equipped with sensors such as cameras, range finders and radio transceivers that, when combined, are responsible for emulating the See And Avoid (SAA) capability on a manned aircraft. DAA systems and conflict resolution systems of UAVs can effectively supplement the legacy of ATC services, meet regulatory requirements, adhere to airspace regulations, and maintain appropriate separation from other aircraft.

There exist a few DAA and collision avoidance systems for aircraft. One of the well-known is the Traffic Alert and Collision Avoidance System (TCAS). It is an autonomous onboard system designed to increase pilot awareness of nearby aircraft and prevent mid-air collisions. TCAS continuously scans the airspace for transponder-equipped aircraft, providing visual and audible resolution advisories (RAs) to maintain vertical separation when proximity thresholds are breached. ACAS, the successor to TCAS, offers significant improvements, with ACAS X demonstrating a remarkable

---

<sup>\*</sup>MSc Advanced Air Mobility Systems, SATM, Cranfield University, and AIAA Student Member.

<sup>†</sup>Research Fellow in Autonomous Systems, SATM, Cranfield University

<sup>‡</sup>Head of the Centre for Autonomous and Cyber-Physical Systems, SATM, Cranfield University

<sup>§</sup>Staff Engineer, Airspace Integration/Special Projects, Office of Airworthiness, 14200 Kirkham Way, Building 27, 2407

59% reduction in collision risk. It also decreases false alert rates by 59% and issues 28% fewer disruptive alerts while resolving encounters more efficiently with simpler alert sequences [1]. ACAS X uses decision theory and probabilistic models [2] to estimate aircraft states, providing optimal advisories based on surveillance data from various sensors, including ADS-B and electro-optical/infrared sensors, making it suitable for UAVs without ADS-B.

Other than TCAS and ACAS, DAIDALUS is a specialized software library designed for Unmanned Aircraft Systems (UAS) to implement a detect-and-avoid concept. It offers critical functionalities, including detection logic, manoeuvre guidance logic (including recovery from loss of separation situations), and alerting logic [3]. DAIDALUS employs formally specified mathematical algorithms that assess the well-clear status of ownship and intruder aircraft, with rigorous verification and validation involving the US Air Force Research Laboratory, MIT Lincoln Laboratory, and NASA. It provides guidance through conflict and recovery bands, allowing UAS operators to maintain or regain separation while respecting performance limits. DAIDALUS has been integrated into the Multi Aircraft Control System (MACS) and is being evaluated in human-in-the-loop experiments at NASA Langley Research Center to assess its suitability for UAS [4].

These softwares use collision avoidance logic to mechanise the deconfliction. It can be mentioned that Collision avoidance (CA) is an essential requirement for a safe flight, and many CA methods of different categories have been reported in the literature. One such category is the geometric methods, which use Euclidean distance calculations based on position and velocity data from aircraft, often obtained through ADS-B transceivers. Resolution of conflicts via geometric methods involves closed-form analytical solutions of numerical optimization problems, thus generating an optimal series of control commands for the aircraft [5], [6]. The second category includes the Force Field methods, which are inspired by the concept of attractive or repulsive forces experienced by charged particles. UAVs are treated as charged particles, and they are attracted towards waypoints and repelled by other UAVs. The resultant of the forces leads to the generation of an optimal flight path [7]. A drawback of this approach is that it is sensitive to local minima, thus often resulting in the generation of sub-optimal paths. Another category of algorithms often involves high computational complexity. Different algorithms under this approach include the Ant colony algorithm, Artificial Bee Colony (ABC) algorithm, genetic algorithms, Bayesian optimisation, etc. Authors in [8] use an ant colony optimisation-based minimum time search algorithm to search optimal collision-free paths under communication-related constraints. Different methods are analysed in [9], from classical graph search path planning algorithms to intelligent methods such as neural networks and evolutionary algorithms.

We have discussed various categories of advanced collision avoidance algorithms available in the literature in the previous paragraphs. Most of them do not focus on following the right-of-way rules in the air. This leads to the generated resolution trajectory being unpredictable and often changing dynamically, which can cause confusion to other airspace users. However, there exists another category of algorithm that uses a prediction-based approach and employs optimal control to solve it. This approach is known as Model Predictive Control (MPC). In the context of CA, MPC takes into account vehicle dynamics, predicts trajectories to identify potential conflicts, and implements control actions in advance to prevent these conflicts.

An integrated path planning algorithm for unmanned ground vehicles (UGVs) is introduced in [10] to navigate around static and moving obstacles. The process involves preprocessing the environment to merge close obstacles and expand their boundaries for safety, computing a global path using the visibility graph method, and optimizing it with nonlinear model predictive control (NMPC), considering various constraints. An algorithm for resolving conflicts in scenarios involving multiple UAVs is proposed in [11], utilizing Distributed Nonlinear Model Predictive Control (DNMPC). This algorithm anticipates potential conflicts and modifies the paths of conflicting UAVs by synchronizing their heading angles and velocities. A decentralized rule-based MPC is used to address collision avoidance among multiple moving vehicles in [12]. Each vehicle employs model predictive control (MPC) for trajectory planning towards its assigned target. Predicted trajectories are exchanged among neighbouring vehicles to anticipate conflicts, and decentralized coordination is applied to resolve these conflicts. [13] addresses the global collision avoidance problem in aviation, which goes beyond pairwise analysis of aircraft trajectories and considers the entire traffic in a given area. The proposed algorithm computes speed and heading adjustments for each aircraft in real-time to ensure collision-free flight trajectories along predetermined waypoints. [14] predicts the dynamics of multiple obstacles and addresses only those relevant to the UAV's immediate prediction horizon using active set algorithms. The approach also enhances avoidance manoeuvres by representing obstacles as orientable ellipsoids, reducing the risk of getting stuck in local minima and allowing for preferred avoidance directions. A decentralized model predictive control (MPC) algorithm is introduced in [15] for planning three-dimensional collision-free flight paths for multiple aircraft. This algorithm utilizes mixed-integer quadratic programming to optimize the trajectory of each aircraft while ensuring safe separation from others. It incorporates compatibility constraints to ensure ongoing feasibility and synchronized calculations within

the decentralized framework. Although these algorithms provide predicted trajectories, the formulation didn't include the constraints to respect the right-of-way rules [16], which are in place to establish protocols ensuring that aircraft operating in close proximity under visual meteorological conditions (VMC) are aware of which aircraft holds the right of way and the appropriate measures to avert collisions. We get the mention of right-of-way rules in the MPC framework in [17], where a collision avoidance system was designed using distributed model predictive control (MPC) in conjunction with ICAO right-of-way rules.

In this paper, we proposed a conflict resolution scheme that implements a model predictive strategy to generate a collision-free path and respects the right-of-way rules. Focus is concentrated on when the intruder is between 75 to 30 seconds away from the ownship. This is because automatic tactical collision avoidance measures such as TCAS are triggered when the intruder and ownship are on a collision course and are only 30 seconds away from each other. Thus, the developed conflict resolution approach can be considered quasi-strategic, implying a degree of strategy that falls between purely tactical and fully strategic approaches. Although our approach is based on MPC, we have implemented the collision condition and constraints in a much simpler way, which can be regarded as our contribution given as follows.

- we have implemented collision constraints such that the predicted trajectory avoids the collision conditions. It can be mentioned that the collision conditions we consider are similar to Proportional Navigation (PN) guidance, i.e., range rate ( $\dot{R} < 0$ ) and LOS rate ( $\dot{\sigma}=0$ ). Implementing these collision conditions in the prediction is new.
- We have considered the case when the intruder does not have the right of way rule or is not cooperative. This is realistic.
- We have implemented the right-of-way rules as path constraints in the optimisation problem. Also, we have used a nonlinear prediction model, which is more realistic in the MPC framework and respects the right-of-way rule.
- We have considered the manoeuvring capability of the intruder in the problem formulation. This is novel as it is not reported in the literature. In previous works, the researchers considered the intruder trajectory a straight line. However, this assumption was too restrictive. In this work, we relaxed the constraint by considering the manoeuvring capability of the intruder and obtaining the ownship trajectory.

### III. Mathematical modelling

#### A. PCA, Line of sight rate and relative distance rate

The PCA is a key term in aircraft conflict resolution, indicating when two aircraft are nearest during a potential conflict. The Time to Closest Approach (TCA) is calculated first to determine the PCA. The relative position vector, which is the difference between the positions of the two aircraft, is used in this calculation. The miss distance vector  $r_m$  is the relative distance between the two aircraft at the PCA.

$$\vec{r}_m = \hat{V}_{rel} \times (\vec{r} \times \hat{V}_{rel}) \quad (1)$$

where  $\vec{r}$  is the relative position and  $\hat{V}_{rel}$  is the unit vector in the direction of relative velocity  $\vec{V}_{rel}$ , which are defined as follows:

$$\vec{r} = \vec{r}_I - \vec{r}_O \quad (2)$$

$$\vec{V}_{rel} = (V_I * \cos(\psi_I) - V_O * \cos(\psi_O), V_I * \sin(\psi_I) - V_O * \sin(\psi_O)) \quad (3)$$

It can be shown that the miss distance vector  $\vec{r}_m$  and relative velocity vector are orthogonal.

$$\vec{r}_m \cdot \vec{V}_{rel} = 0 \quad (4)$$

The relation between relative position and miss distance vector can be used to determine  $\tau$ , the time to closest approach (TCA):

$$\vec{r}_m = \vec{r} + \vec{V}_{rel} \cdot \tau \quad (5)$$

From equation 5 and 4, the TCA  $\tau$  can be found as follows:

$$\tau = -\frac{\vec{r} \cdot \vec{V}_{rel}}{\vec{V}_{rel} \cdot \vec{V}_{rel}} \quad (6)$$

The point of closest approach PCA can be calculated separately for the ownship and the intruder once the TCA is

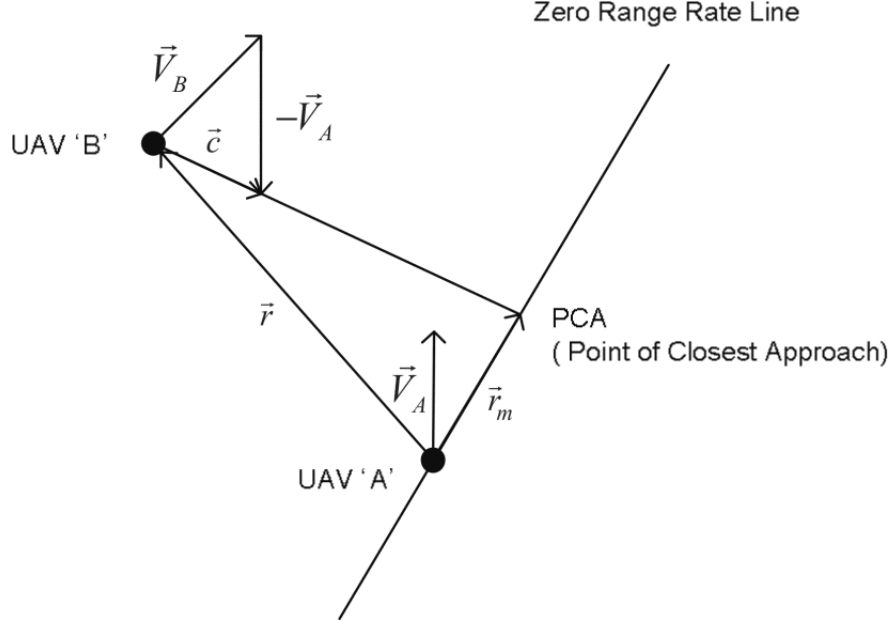


**Fig. 1 Relative motion between two UAVs A and B**

known as follows:

$$PCA_O = \vec{r}_O + \vec{V}_O \cdot \tau \quad (7)$$

$$PCA_I = \vec{r}_I + \vec{V}_I \cdot \tau \quad (8)$$

Figure 1 shows all the vectors and distances as calculated above in reference to two UAVs [18]. The relative Distance Rate is necessary to calculate the time to go (TTG), which can be defined as the time for the 2 UAVs to collide (assuming they are on a collision course). In figure 2, the schematic diagram is shown for a two-dimensional case. Ownship UAV heading and velocity are denoted by  $\psi$  and  $\vec{V}(V_x, V_y)$ . Corresponding Intruder UAV parameters are denoted by  $\psi_I$  and  $\vec{V}_I(V_{x,I}, V_{y,I})$ . The relative distance rate  $\dot{R}$  is defined as follows:

$$\dot{R} = V_I * \cos(\psi_I - \sigma) - V * \cos(\psi - \sigma) \quad (9)$$

here  $\sigma$  is the line of sight angle between the ownship and intruder defined as follows:

$$\sigma = \arctan \frac{y_I - y_O}{x_I - x_O} \quad (10)$$

where the intruder location is  $(x_I, y_I)$  and the ownship location is  $(x_O, y_O)$ . The line of sight rate, which is the derivative of the line of sight angle, can be calculated using the following equation:

$$\dot{\sigma} = \frac{V_I * \sin(\psi_I - \sigma) - V * \sin(\psi - \sigma)}{R} \quad (11)$$

where  $R$  is the instantaneous distance between the ownship and the intruder. The time to go (TTG) can be defined as:

$$TTG = -\frac{R}{\dot{R}} \quad (12)$$

## B. UAV mathematical modelling

the fixed-wing UAV is kinematically modelled as a point mass in a 2D horizontal plane. The state space form of the UAV is as shown below:

$$X = \begin{bmatrix} x \\ y \\ \psi \\ V \end{bmatrix} \quad (13)$$

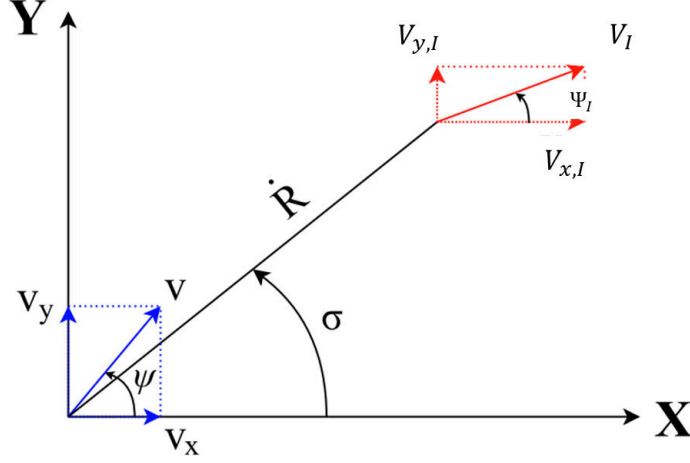


**Fig. 2 ownship and intruder UAV schematics [19]**

where  $x$  and  $y$  represent coordinate positions,  $\psi$  is the heading angle, and  $V$  is the scalar speed of the UAV measured positively in the direction of the heading angle. The kinematic model of the UAV is thus:

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{V} \end{bmatrix} = \begin{bmatrix} V \cos(\psi) \\ V \sin(\psi) \\ \alpha \\ a \end{bmatrix} \quad (14)$$

The control inputs to the UAV do not interfere with low-level signal inputs such as throttle or control surface deflections. Instead, the control vector is defined as:

$$U = \begin{bmatrix} \alpha \\ a \end{bmatrix} \quad (15)$$

where  $\alpha$  and  $a$  are the lateral and linear accelerations. It can be shown that such simple dynamics are often used for experimental UAVs. Constraints have been imposed on the minimum and maximum turn rates  $\alpha$  to restrict UAV movements within its performance capabilities. The discrete version of the UAV dynamics as obtained from simple Euler integration is as follows:

$$\begin{aligned} x(k+1) &= x(k) + \Delta T V \cos(\psi) \\ y(k+1) &= y(k) + \Delta T V \sin(\psi) \\ \psi(k+1) &= \psi(k) + \Delta T \alpha \\ V(k+1) &= v(k) + \Delta T a \end{aligned} \quad (16)$$

The Euler integration method exhibits a greater level of truncation error compared to methods such as Runge-kutta order 3 or 4. The RK4 method was thus chosen for the dynamics modelling in the model predictive control of the UAV due to its accuracy and because the MATLAB non-linear model predictive control toolbox uses an ode45 differential equation solver that is based on the RK4 approach.

#### IV. Formulation of a Model Predictive Control Problem

The primary aim is to make sure the developed guidance algorithm incorporates the right-of-way rules of the air while generating resolution trajectories. This makes sure the resolution of conflict is carried out in an orderly and predictable manner. The developed algorithm must work in real-time to respond dynamically to changes in obstacle and intruder aircraft flight characteristics. An optimal control-based approach must be taken to minimize control inputs and other parameters while generating the optimum conflict resolution trajectory.

A non-linear model predictive control problem, is formulated to solve the conflict resolution problem that respects the right-of-way rules. It is assumed the optimisation problem is convex in nature, i.e., the cost function and all constraint functions are convex functions.

- **Sample time:** Sample time refers to the discrete interval at which control actions are computed and applied to the system. A sample time of 0.01 seconds has been used in the present MPC formulation.
- **Prediction Horizon:** The prediction horizon is the time duration over which future system states are predicted or estimated using a model. It is used to forecast the system's behaviour, enabling the controller to anticipate future states. A prediction horizon of 60 has been used.
- **Control Horizon:** The control horizon is the finite time window into the future over which control inputs are determined and optimised. A control horizon of 15 has been chosen to be used in this case.

#### A. Cost Function

The cost function chosen for this problem aims to minimise the control input  $u$ . It makes sure the optimal trajectory thus obtained is smooth and does not have abrupt changes in speed or heading. The control input vector, as described earlier, contains the linear acceleration and yaw angle change rate. Thus, minimization of this results in minimal changes in aircraft speed and heading angle. The discrete cost function  $J$  is as shown below:

$$J = \min_u \sum_{k=1}^N u_k R_p u_k^T \quad (17)$$

where  $R_p$  is a positive definite matrix.

#### B. Constraints Formulation

The optimization problem involves three primary types of constraints: input constraints, state constraints, and kinematics constraints. Input constraints and kinematics constraints are imposed on the input parameters of the UAV, taking into account the UAV's physical characteristics. The kinematics constraints specify how the UAV moves and establish the relationship between the state vector  $x$  and control input vector  $u$  based on the non-linear UAV kinematic model. The control input constraints used in this problem represent the performance limitations of the UAV. They are as follows:

$$\begin{aligned} a_{min} = -10 \text{units/s}^2 \leq u(1) \leq a_{max} = 10 \text{units/s}^2 \\ \dot{\psi}_{min} = -5^\circ/\text{s} \leq u(2) \leq \dot{\psi}_{max} = 5^\circ/\text{s} \end{aligned} \quad (18)$$

A limit of  $10 \text{ units/s}^2$  was set on the magnitude of the linear acceleration of the UAV to make sure the UAV has enough thrust and braking power to achieve commanded accelerations. The parameters are left to be dimensionless to make it easier to adapt to different measurement systems. Usually, the accelerating and decelerating power of a UAV is not the same magnitude-wise; however, this was considered in this research to simplify the problems. It can be easily changed if required. A limit on turning rates was set to be  $5^\circ/\text{s}$ .

State constraints are crucial to make sure the UAV flies realistically and conforms to all the constraints imposed by airspace restrictions and flight rules. A constraint was set on the speed of the aircraft so that it does not fly slower than the stall speed of 100 knots (typical of some General Atomics fixed-wing UAVs [20]), which, after scaling, came to be around two units in the simulation environment. Thus,

$$2 \leq X(4) \quad (19)$$

*Collision constraints* are included to make sure the conflicts are resolved as follows:

$$\begin{aligned} \dot{R} &> 0 \\ \dot{\sigma} &\neq 0 \\ d_{PCA} &> r \end{aligned} \quad (20)$$

Where  $d_{PCA}$  is the distance between the two UAVs when each of them is at their respective point of closest approach (PCA), and  $r$  is the minimum allowable separation. This condition ensures that the two UAVs are separated by a distance greater than  $r$  when they are closest to each other. Along with the other two conditions on  $\dot{R}$  and  $\dot{\sigma}$ , make sure that the two UAVs resolve the conflict in the least time elapsed.

The other state constraints are based on the right-of-way rules that must be followed when a conflict is resolved. The International Civil Aviation Organization (ICAO) has established fundamental rules for determining the right of way [16]. These rules are used to evaluate which aircraft has the right of way in different flight scenarios. The relevant rules are reproduced here as follows:

**Rule e FAA CFR-91.113 - Approaching headon.** *When aircraft are approaching each other headon, or nearly so, each pilot of each aircraft shall alter course to the right.*

**Rule d FAA CFR-91.113 - Converging.** *When aircraft of the same category are converging at approximately the same altitude (except head-on, or nearly so), the aircraft to the other's right has the right-of-way.*

**Rule f FAA CFR-91.113 - Overtaking.** *Each aircraft that is being overtaken has the right-of-way and each pilot of an overtaking aircraft shall alter course to the right to pass well clear.*

These rules are mathematically formulated as constraints in the model predictive control problem, as explained in the following sections.

### 1. Head-on collision

In this scenario, the ownship is required to move over to the right and pass the intruder from its left. The intruder is also expected to move to its right and pass the ownship on its left. However, the ownship does not depend on the intruder to cooperate and move over; thus, the constraint is imposed for the ownship always to make sure to move over to its right. This is ensured by making the  $x$ -coordinate of the ownship greater than the  $x$ -coordinate of the intruder throughout the duration of the conflict resolution manoeuvre. It is better explained using equation 21 and the figure 3.

$$x_I - x_O \leq 0 \quad (21)$$

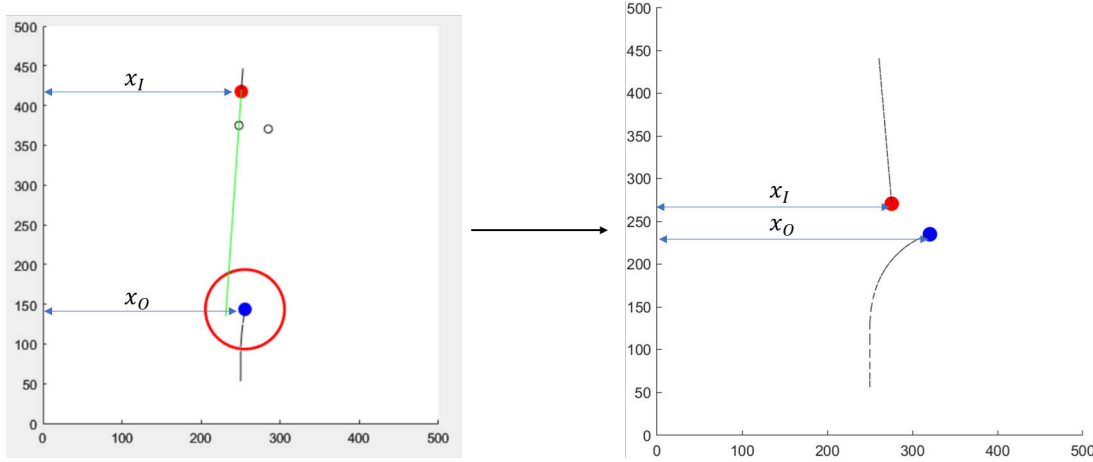


**Fig. 3 Satisfaction of right of way constraint for head-on collision scenario**

### 2. When intruder has the right of way

In this scenario, the intruder is on the right side of the ownship and thus has the right of way. The ownship cannot cut in front of the intruder and should pass the intruder from behind. This also required the ownship to deviate to the right of its current path and try to cross the intruder keeping it to its left. Although the same constraint as in the head-on collision condition should theoretically be able to satisfy this condition, a different constraint was chosen to satisfy this condition. It was decided to restrict the heading angle of the ownship to be less than the current line of sight angle between the ownship and the intruder. This would ensure that the ownship always pointed in a direction that passed behind the intruder location at any given time instant. Figure 4 and equation 22 elaborate this constraint.

$$\psi_O - \sigma \leq 0 \quad (22)$$

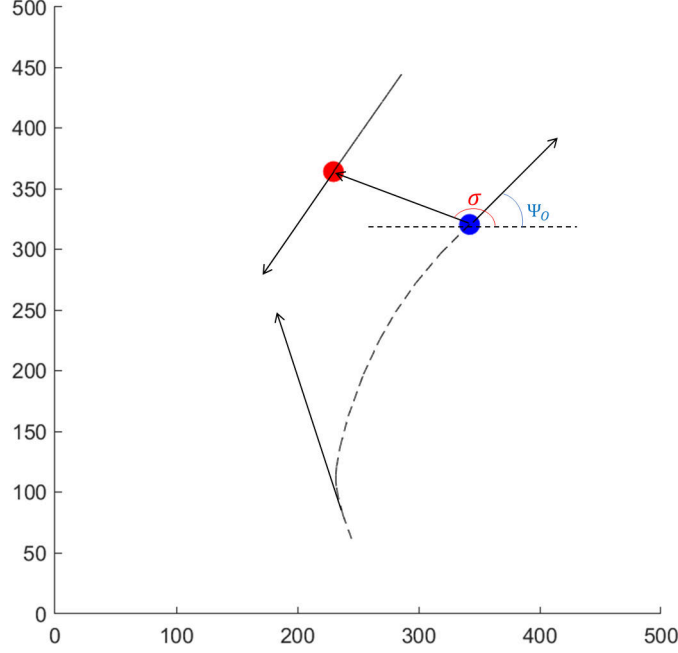


**Fig. 4 Satisfaction of right of way constraint when an intruder has the right of way**

### 3. When ownship has the right of way

In a similar case as above, the ownship is situated to the right of the intruder and thus has the right of way. It is allowed to continue on its path and only conforms to the other constraints related to collision avoidance and physical and kinematic limitations to resolve the conflict. To make sure the constraint in this condition is always satisfied, the constraint function, in this case, is just a high negative value.

## V. Simulation, Results and Discussion

### A. Simulation environment

A simulation scenario is created in MATLAB to test the performance of the conflict resolution algorithm, as shown in figure 5. The simulation environment is designed to represent an airspace. An intruder aircraft spawns at a specific location. The intruder is programmed to fly at a random speed and direction. A degree of randomness is required to make sure the conflict resolution algorithm that is implemented is robust and can tackle a range of conflict scenarios. The ownship aircraft is spawned at a different location with user-defined initial heading, speed, and acceleration, introducing a degree of variation and uncertainty. The simulation environment represents an airspace with two aircraft: the intruder (red dot) and the ownship (blue dot). The green line predicts the intruder's path based on past observations. The black circles represent the point of closest approach between the two aircraft. The thick red circle around the ownship indicates the minimum safe separation distance. The research aims to prevent the intruder from entering this circular zone around the ownship aircraft by adjusting its heading angle and speed. The red lines indicate the spatial limitations of each aircraft, considering their manoeuvring capability. The region bounded by these lines in front of each aircraft represents the possible locations of the aircraft based on a certain lookahead time. With the conflict resolution algorithm active, as time passes, these areas reduce and eventually stop overlapping, eliminating all conflict possibilities.

### B. Results and Discussion

Figure 6 shows the trajectories taken by the ownship in some common conflict scenarios.

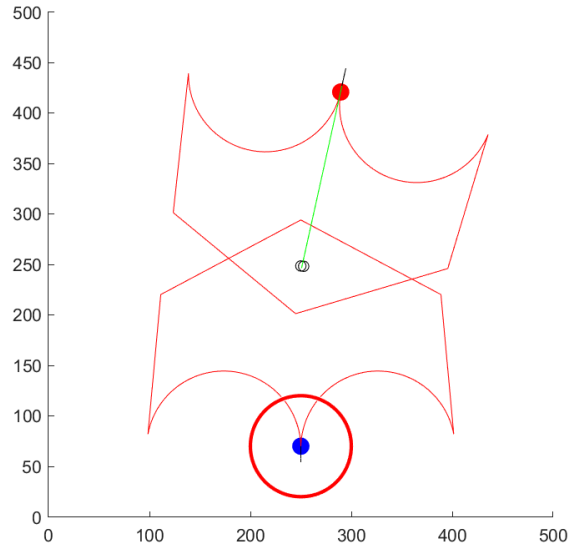
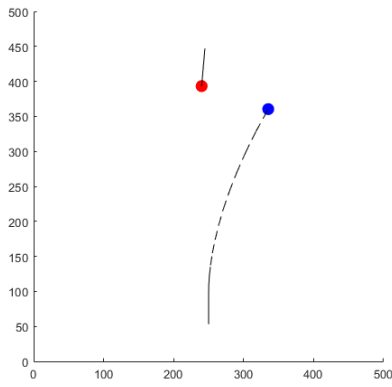
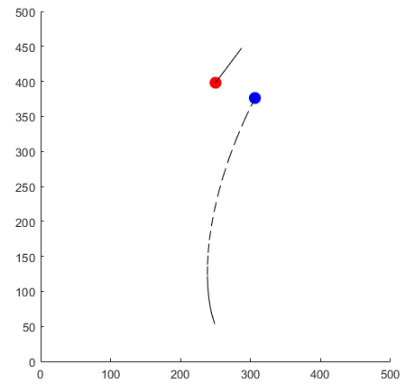


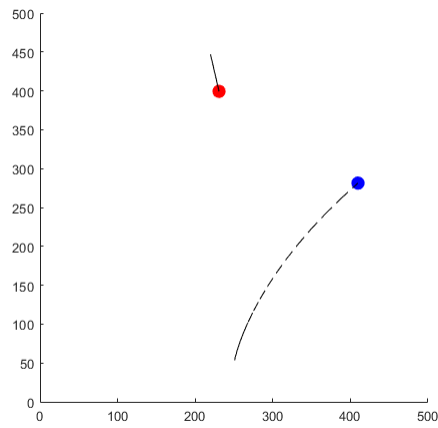
**Fig. 5 Simulated environment**



**(a) Headon collision**



**(b) Intruder has the right of way**



**(c) Ownship has the right of way**

**Fig. 6 Resolution trajectories of ownship in different conflict scenarios**

The trajectories, as depicted in figure 6, clearly show that the algorithm is capable of resolving conflicts in a variety of scenarios. When the ownship and the intruder are flying towards each other on a head-on collision course, the right-of-way rules require the ownship to deviate to the right and pass the intruder on its left. A head-on collision is said to be imminent when the intruder position is somewhere between  $70^\circ$  and  $110^\circ$  heading of the ownship. The results demonstrate that the MPC algorithm can successfully resolve the conflict while respecting the right-of-way rules and maintaining safe separation. Every simulation scenario ended when the relative distance increased ( $\dot{R} > 0$ ). This implies that the intruder is moving away from the ownship, and the conflict has been resolved.

### 1. Head-on

Detailed analysis of the conflict resolution scenario was done when the intruder and ownship were on a head-on collision course, as depicted in figure 6a. The plots for range rate, line of sight rate and control inputs are shown in Figure 7 as a function of simulation iteration.

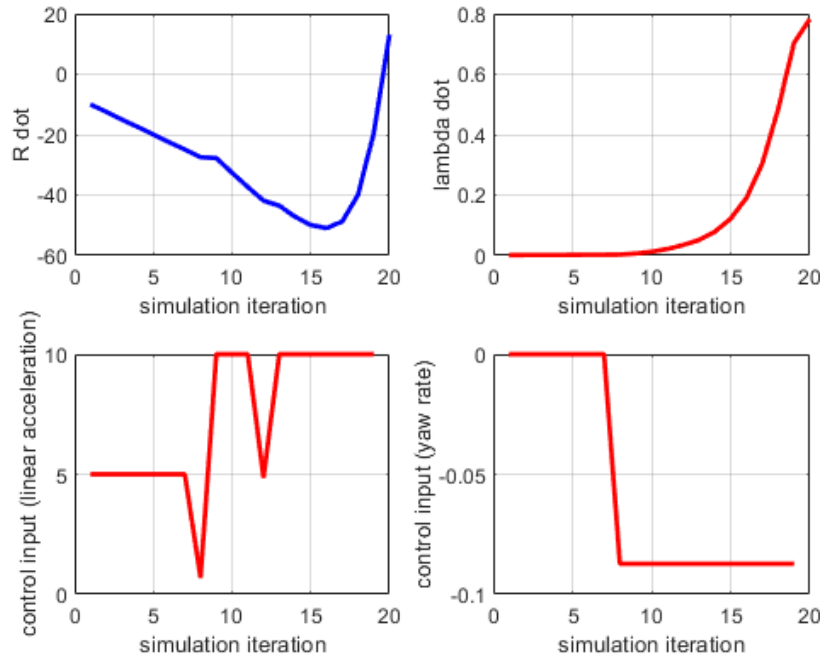


**Fig. 7 Aircraft state and control inputs vs iteration in a head-on collision scenario**

The initial analysis of the simulation showed a collision condition indicated by a negative relative distance rate ( $\dot{R}$ ) and a stationary line of sight angle rate ( $\dot{\sigma}$ ). However, the MPC controller effectively intervened, making these values positive and non-zero respectively as the simulation progressed. Initially, the linear acceleration remained constant due to the prediction horizon not accounting for the intruder, leading to a potential collision. As time passed, the MPC controller detected the intruder and responded by decelerating the ownship sharply and changing its heading angle (negative yaw rate,  $\dot{\psi}$ ), indicating a right turn as per airborne right-of-way rules. In Figure 8, the ownship velocity and relative distance are shown. It can be observed that the velocity changes as per the linear acceleration. The limitations are primarily performance-based. Figure 8 also illustrates the separation between the intruder and ownship across time. As expected, the separation diminishes, reaching a minimum of approximately 100 by the simulation's end. This value exceeds the minimum safe separation of 50, confirming the MPC controller's success in ensuring the ownship maintains a safe distance from the intruder. It effectively resolves head-on conflicts.

### 2. Intruder has the right of way

A similar detailed analysis was done for the scenario where the intruder had the right of way (figure 6b). As seen in the simulation's initial phase, the intruder UAV starts with a speed of 5 units and a heading angle of  $231^\circ$ . At the same

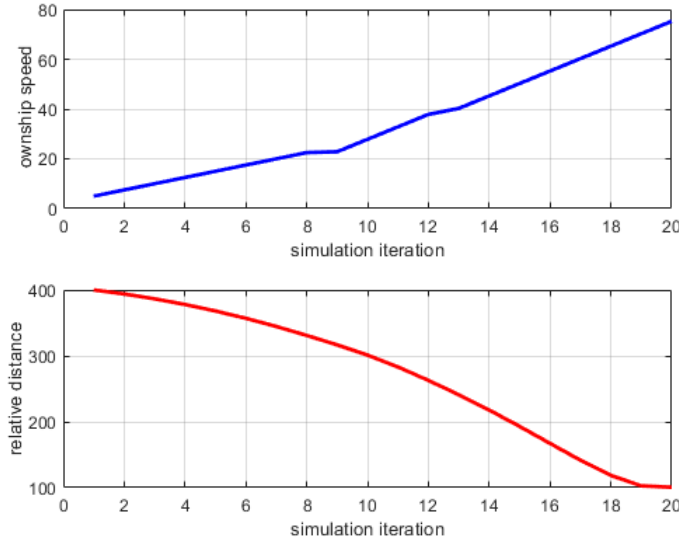


**Fig. 8 Ownship speed and inter UAV separation vs simulation iteration for head-on collision scenario**

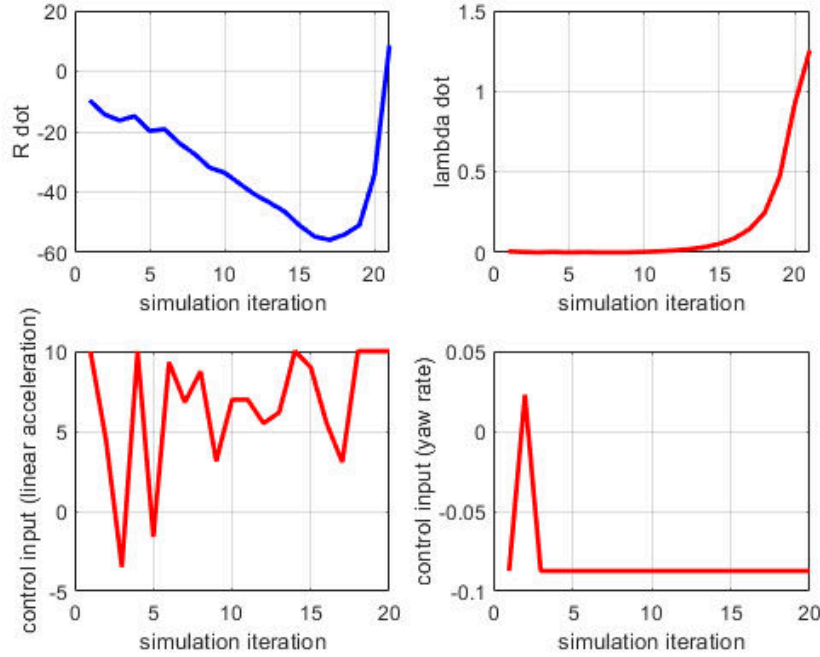


**Fig. 9 Simulation parameters and control inputs vs iteration when an intruder has the right of way**

time, the ownship moves at an initial speed of 4 units along a heading angle of  $120^\circ$  with an acceleration of 3 units. Compared to the previous head-on collision scenario, the controller takes action earlier due to the easier calculation of  $\dot{\sigma}$  when it is not close to multiples of  $90^\circ$ . Near  $90^\circ$  angles, such as in head-on collisions, can cause numerical and mathematical errors, especially when dealing with trigonometric functions like sine and cosine. The plots for range rate, line of sight rate and control inputs are shown in Figure 9. The simulation concludes when  $\dot{R}$  becomes non-negative around the  $16^{th}$  iteration. Initially the line of sight rate was 0 but gradually became non-zero as the MPC controller constraints were met. This combination of conditions ensured conflict resolution. Linear acceleration and  $\dot{\psi}$  vs iteration

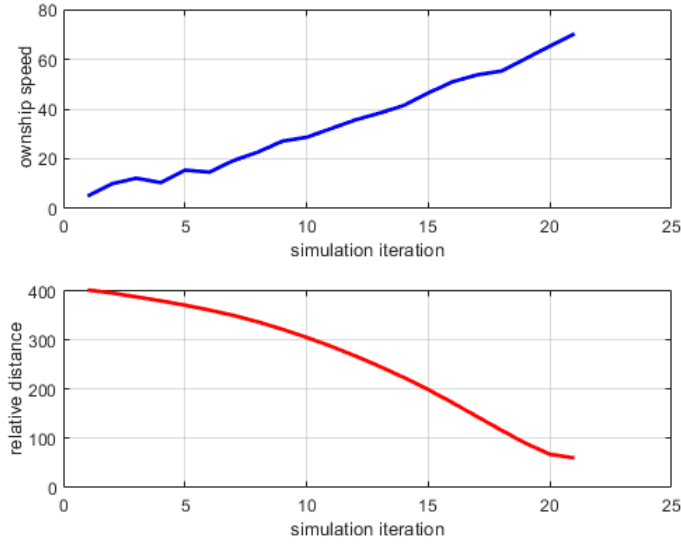


**Fig. 10** Ownship speed and inter-UAV separation vs simulation iteration when an intruder has the right of way

plots demonstrate how the MPC controller guided the UAV away from the conflict. The  $\psi$  curve shows gradual changes without abrupt shifts, indicating a smooth resolution process. The velocity and the relative distance between the ownship and the intruder are shown in figure 10. It is clear that the relative distance satisfies the minimum separation constraint.

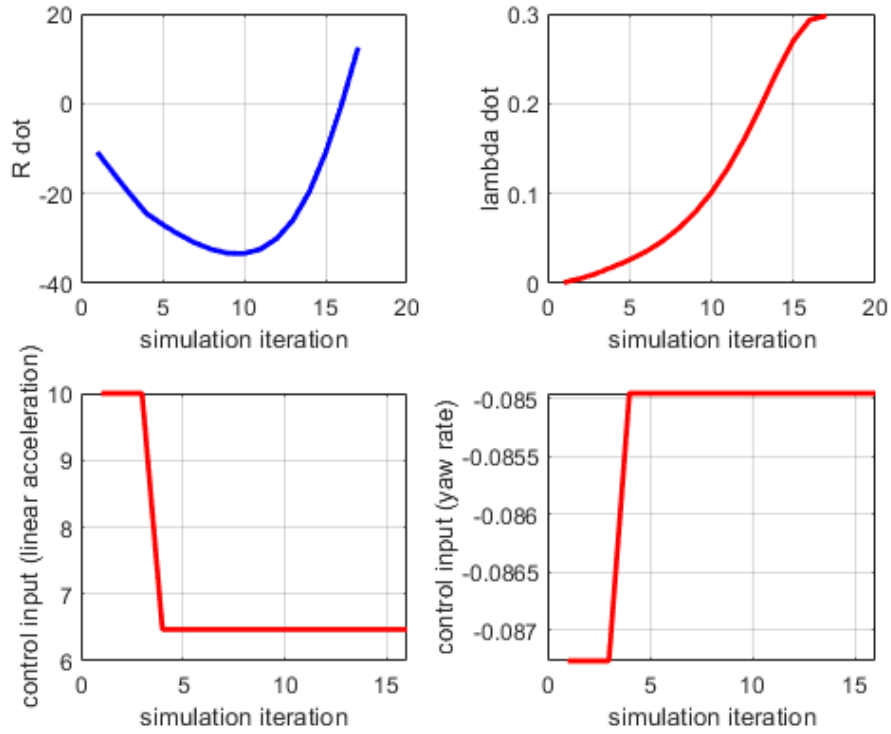


**Fig. 11** Simulation parameters and control inputs vs iteration when ownship has the right of way

### 3. Ownship has the right of way

When ownship had the right of way (figure 6c), the results, upon further analysis, produced plots as shown in figure 11. In this situation, the ideal conflict resolution would involve the intruder maneuvering to its right, passing the ownship from behind. However, considering the assumption that the intruder might be non-cooperative, the ownship has to take action and move out of the way to maintain safe separation.

The plot of  $\dot{R}$  vs iteration shows the initial approach of the intruder toward the ownship (figure 11). However, the controller effectively ensures  $\dot{R}$  becomes positive, indicating the intruder is moving away from the ownship ultimately. The line of sight rate curve further confirms conflict resolution, transitioning from 0 (implying the intruder directly moves toward the ownship) to a positive value. Analysis of control inputs reveals relatively smooth changes in linear acceleration and heading angle rate ( $\dot{\psi}$ ). The controller commands a negative  $\dot{\psi}$  to guide the ownship in turning away from the intruder, a manoeuvre evident in its trajectory. Also, the relative distance (in figure 12) constraint is also satisfied.

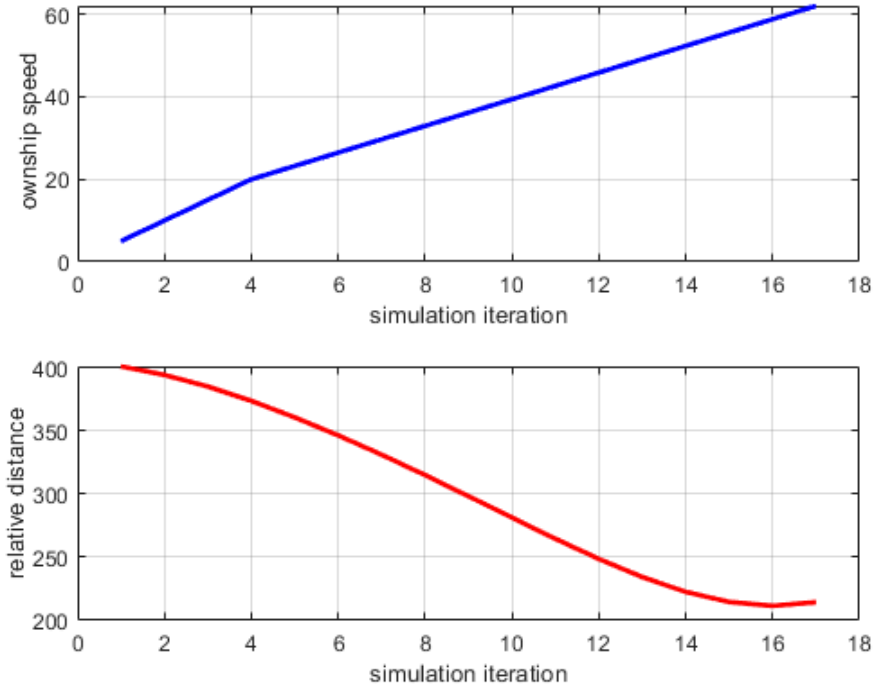


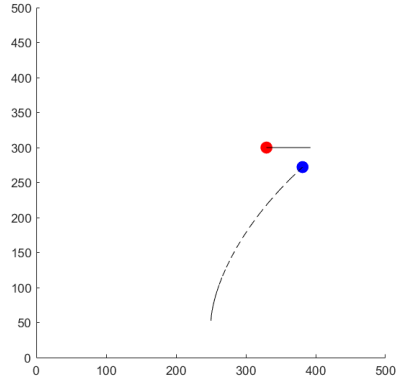
**Fig. 12** ownship speed and inter UAV separation vs simulation iteration when ownship has the right of way

## VI. Conclusion and Future Work

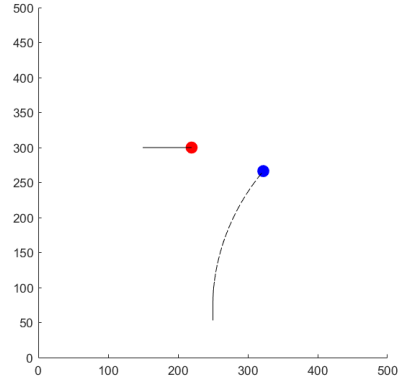
In this research, an optimization-based approach using model predictive control was employed to develop a real-time conflict resolution algorithm for UAVs, emphasizing strict adherence to right-of-way rules for airspace order and predictability. The algorithm successfully resolved conflicts in various scenarios with a variety of initial parameters. However, limitations were noted, including the algorithm's tendency to ignore the minimum speed constraint, occasionally leading to impractical manoeuvres. Future improvements could involve refining cost functions and constraints to enforce gentler controls. Additionally, expanding the algorithm to 3D airspace and handling multiple intruder aircraft could enhance its effectiveness. In conclusion, the simulation study demonstrates the algorithm's effectiveness while highlighting areas for further refinement and expansion.

## Appendix

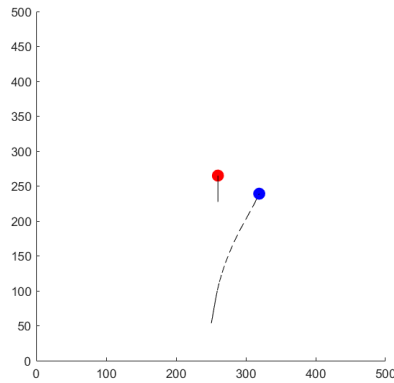
The developed MPC resolution controller was implemented in some more conflict scenarios, and the plots showing the trajectories of the intruder and ownship in such scenarios are shown below in figure 13.



(a) Intruder approaches from the right



(b) Intruder approaches from the left



(c) Ownship tries to overtake the intruder

**Fig. 13 Additional conflict scenario**

## References

- [1] Holland, J. E., Kochenderfer, M. J., and Olson, W. A., "Optimizing the next generation collision avoidance system for safe, suitable, and acceptable operational performance," *Air Traffic Control Quarterly*, Vol. 21, No. 3, 2013, pp. 275–297.
- [2] Kochenderfer, M. J., Chrysanthopoulos, J. P., and Weibel, R. E., "A new approach for designing safer collision avoidance systems," *Air Traffic Control Quarterly*, Vol. 20, No. 1, 2012, pp. 27–45.
- [3] Muñoz, C., Narkawicz, A., Hagen, G., Upchurch, J., Dutle, A., Consiglio, M., and Chamberlain, J., "DAIDALUS: detect and avoid alerting logic for unmanned systems," *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, IEEE, 2015, pp. 5A1–1.
- [4] Chamberlain, J. P., Consiglio, M. C., Comstock Jr, J. R., Ghatas, R. W., and Muñoz, C., "NASA controller acceptability study 1 (CAS-1) experiment description and initial observations," Tech. rep., 2015.
- [5] Bilimoria, K., "A geometric optimization approach to aircraft conflict resolution," *18th Applied aerodynamics conference*, 2000, p. 4265.

- [6] Goss, J., Rajvanshi, R., and Subbarao, K., "Aircraft conflict detection and resolution using mixed geometric and collision cone approaches," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004, p. 4879.
- [7] Oroko, J. A., and Nyakoe, G., "Obstacle avoidance and path planning schemes for autonomous navigation of a mobile robot: a review," *Proceedings of the Sustainable Research and Innovation Conference*, 2022, pp. 314–318.
- [8] Pérez-Carabaza, S., Scherer, J., Rinner, B., López-Orozco, J. A., and Besada-Portas, E., "UAV trajectory optimization for Minimum Time Search with communication constraints and collision avoidance," *Engineering Applications of Artificial Intelligence*, Vol. 85, 2019, pp. 357–371.
- [9] Polvara, R., Sharma, S., Wan, J., Manning, A., and Sutton, R., "Obstacle avoidance approaches for autonomous navigation of unmanned surface vehicles," *The Journal of Navigation*, Vol. 71, No. 1, 2018, pp. 241–256.
- [10] Hang, P., Huang, S., Chen, X., and Tan, K. K., "Path planning of collision avoidance for unmanned ground vehicles: A nonlinear model predictive control approach," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, Vol. 235, No. 2, 2021, pp. 222–236.
- [11] Chen, H.-X., Nan, Y., and Yang, Y., "Real-time conflict resolution algorithm for multi-UAV based on model predict control," *Algorithms*, Vol. 12, No. 2, 2019, p. 47.
- [12] Izadi, H. A., Gordon, B. W., and Zhang, Y., "Rule-based cooperative collision avoidance using decentralized model predictive control," *Infotech@ Aerospace 2011*, 2011, p. 1610.
- [13] Bousson, K., "Model predictive control approach to global air collision avoidance," *Aircraft Engineering and Aerospace Technology*, Vol. 80, No. 6, 2008, pp. 605–612.
- [14] Castillo-Lopez, M., Sajadi-Alamdari, S. A., Sanchez-Lopez, J. L., Olivares-Mendez, M. A., and Voos, H., "Model predictive control for aerial collision avoidance in dynamic environments," *2018 26th Mediterranean Conference on Control and Automation (MED)*, IEEE, 2018, pp. 1–6.
- [15] Yokoyama, N., "Decentralized model predictive control for planning three-dimensional conflict-free trajectories," *AIAA Guidance, Navigation, and Control Conference*, 2014, p. 0970.
- [16] "Code of Federal Regulations (CFR) Title 14 Part 91.113," U.S. Government Publishing Office, 2023. URL <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-91/subpart-B/subject-group-ECFR4c59b5f5506932/section-91.113>, accessed on 19/06/2023.
- [17] D'Amato, E., Mattei, M., and Notaro, I., "Distributed reactive model predictive control for collision avoidance of unmanned aerial vehicles in civil airspace," *Journal of Intelligent & Robotic Systems*, Vol. 97, No. 1, 2020, pp. 185–203.
- [18] Park, J.-W., Oh, H.-D., and Tahk, M.-J., "UAV collision avoidance based on geometric approach," *2008 SICE Annual Conference*, 2008, pp. 2122–2126. <https://doi.org/10.1109/SICE.2008.4655013>.
- [19] Subies Hueso, J., Mondal, S., Tsourdos, A., and Chadwick, A., "Real-Time Collision Avoidance Trajectory Planner Using Generalized Vector Explicit Guidance," *AIAA SCITECH 2023 Forum*, 2023, p. 1734.
- [20] Forces, A., "General Atomics MQ-1 Predator," 2017. URL [https://armedforces.eu/air\\_forces/drone/General\\_Atomics\\_MQ-1\\_Predator#:~:text=The%20General%20Atomic%20MQ%2D1%20Predator%20can%20achieve%20maximum%20speed,is%20about%20100%20km%2Fh.](https://armedforces.eu/air_forces/drone/General_Atomics_MQ-1_Predator#:~:text=The%20General%20Atomic%20MQ%2D1%20Predator%20can%20achieve%20maximum%20speed,is%20about%20100%20km%2Fh.), accessed on April 20, 2023.



**DSPACE**

<https://dspace.org/>

## **Autonomous detect and avoid algorithm respecting airborne right of way rules**

**Mukherjee, Anurag; Mondal, Sabyasachi; Tsourdos, Antonios**

**2024-01-04**

American Institute of Aeronautics and Astronautics (AIAA)

<https://doi.org/10.2514/6.2024-1080>

Mukherjee A, Mondal S, Tsourdos A. (2024) Autonomous detect and avoid algorithm respecting airborne right of way rules. In: AIAA SCITECH 2024, 8-12 January 2024, Orlando, USA. Paper number AIAA 2024-1080