

Real-Time Collision Avoidance Trajectory Planner Using Generalized Vector Explicit Guidance

Josep Subies*, Sabyasachi Mondal[†] and Antonios Tsourdos[‡]
Cranfield University, Cranfield MK430AL, United Kingdom

Andrew Chadwick[§]
Connected Places Catapult, Milton Keynes MK91BP, United Kingdom

In this paper, a real-time new collision avoidance scheme is proposed using the Generalized Vector Explicit Guidance (GENEX) law, an optimal guidance algorithm that can simultaneously achieve design specifications on miss distance and final UAV–target relative orientation. Low computation, closed-form expression, and optimal derivation are the key features that make it a potential candidate for real-time applications. Moreover, an additional control strategy has been included (based on barrier function) to ensure the safety of the UAV while avoiding obstacles. The guidance law has been verified through simulations for the two-dimensional and three-dimensional scenarios, and the results are very satisfactory.

I. Nomenclature

α	=	angle between the line connecting the two circle centers and the horizontal
CAS	=	collision avoidance system
DDP	=	desired destination point
δ	=	angle between V and R vectors
ψ	=	current UAV heading
ψ_f	=	final UAV heading
GENEX	=	generalized vector explicit guidance
GMP	=	gap middle point
μ	=	difference between UAV current and final heading
R	=	relative distance between current and final positions
V	=	current UAV velocity
V_f	=	final UAV velocity

II. Introduction

NOWADAYS, the popularity of Unmanned Aerial Vehicles (UAVs) is exponentially increasing. However, as UAV popularity soars, so do the potential risks associated with it. For example, missions that take place in civilian areas or the common airspace are quite difficult to handle due to legal restrictions and potential collision risks [1]. UAVs need to reliably avoid other obstacles while reaching their desired destination or completing a specific task to allow for more flexibility in those missions. Therefore, UAVs need to incorporate a Collision Avoidance System (CAS) that can handle and mitigate those risks in their architecture. Another important fact to consider is that most UAV missions occur outdoors, where the environments are unknown and dynamically changing. Hence, it is important that the collision avoidance system can work in real-time and make online computations to ensure the completion of the mission and to guarantee the safety of the vehicle and its surroundings.

*MSc student in Autonomous Vehicle Dynamics and Control, School of Aerospace, Transport and Manufacturing (SATM), Cranfield University, College Road, Cranfield, Bedford, MK43 0AL, United Kingdom. Email: J.Subies-Hueso.664@cranfield.ac.uk

[†]Researcher in Aerospace Engineering and Control Systems Engineering, School of Aerospace, Transport and Manufacturing (SATM), Cranfield University, Email: sabyasachi.mondal@cranfield.ac.uk

[‡]Head of the Autonomous and Cyber-Physical Systems Centre, School of Aerospace, Transport and Manufacturing (SATM), Cranfield University, Email: a.tsourdos@cranfield.ac.uk, AIAA Senior Member.

[§]Aviation Technology Innovation Lead, Airmobility & Airports, Email: andrew.chadwick@cp.catapult.org.uk

Many research works were reported in the literature, where the collision avoidance problem was addressed. The most widely used methods and their features are shown in Table 1 [2][3]. Geometric methods work by computing the distance between the UAV and the obstacle using positions and velocities of both [3]. In [4], the collision cone approach, a widely used geometric method, is used to avoid collisions between two aircraft in a 3D environment. In [5], this approach is used for UAVs in formation flight and in [6] for pairwise non-cooperative aircraft. The potential field methods use the concept of attractive and repulsive potentials to either attract a UAV towards a target or repel it from an obstacle [3]. In [7], it is used to guide a mobile robot in a cluttered environment and in [8], real-time trajectories are adjusted to avoid collisions in 3D space using cooperative UAVs. Optimization-based methods aim at finding the optimal or near-optimal path toward the desired destination. In [9], an A* dynamic algorithm was developed for multiple ship encounters. In [10], an advanced way of formation flying is proposed where the rapidly exploring random tree (RRT) algorithm is used to define the waypoint nodes. Probabilistic modelling methods are used for decision-making in dynamic environments as information about the threat becomes available. In [11], a cost function is optimized to generate avoidance strategies.

In Table 1, we can see that there exists a trade-off between online work and optimal paths. The main reason is that it usually takes much computational time for a guidance algorithm to compute an optimal path. Therefore, optimal collision avoidance methods such as optimization-based and probabilistic modelling tend to do this work offline. On the other hand, geometric and potential field methods are faster, computationally efficient, and easy to implement. Therefore, for online computations, the geometric and the potential fields methods are more suitable than the optimization-based and probabilistic modelling. However, the local minima problem of the potential field methods can be a big issue that must be thoroughly considered when selecting this method over the others. It is important to

Table 1 Collision Avoidance Methods Comparison:
Fast (FS), Computationally Efficient (CE), Good Performance (GP), Easy Implementation (EI), Online Work (OL), Optimal (OP), Handles Local Minima Problem (HLMP)

Collision Avoidance Method	FS	CE	GP	EI	OL	OP	HLMP
Geometric	✓	✓	✓	✓	✓	✗	✓
Potential Fields	✓	✓	✓	✓	✓	✗	✗
Optimization-Based	✗	✗	✗	✗	✗	✓	✓
Probabilistic Modeling	✗	✗	✗	✗	✗	✓	✓

highlight that optimality can improve operation time or range for an energy-limited platform such as a UAV. In this paper, we present Generalized Explicit Vector Guidance (GENEX) based obstacle avoidance scheme that satisfies all the requirements overcoming the trade-off between optimality and online work. GENEX is an optimal guidance algorithm which is used for missile engagement problems [12], [13]. It can simultaneously achieve design specifications on miss distance and final UAV-target relative orientation. As mentioned earlier, optimal path generation is essential for UAVs since they have limited energy capacity. Moreover, the guidance command is easy to mechanize, computationally cheap, and computed in closed form. To get a clear understanding of the features and advantages of GENEX over the widely used collision avoidance methods, we have presented a comparison in Table 2 [2][3]. It can be seen that GENEX overcomes the trade-off problem between online work and optimality. Hence, being more suitable for online computations and real-time applications than the existing methods.

III. Problem Formulation

This study aims to develop a new collision avoidance algorithm that can perform properly in real-time, avoid static obstacles and reach the destination. In this paper, we consider the problem to be two and three-dimensional. For simplicity, we present a representative diagram of the problem scenario in 2D Figure 1. The UAV starts at point A , passes through the two obstacles Obs_1 and Obs_2 , and reaches the desired destination point (DDP) B (shown as a blue dot). The obstacles are assumed to be circular. The dotted line indicates the path taken by the UAV towards the desired destination point. The UAV shape is defined in Figure 2. The green and red blades represent the front and rear blades, respectively. An arrow is also included in the UAV definition to clearly indicate the heading of the UAV.

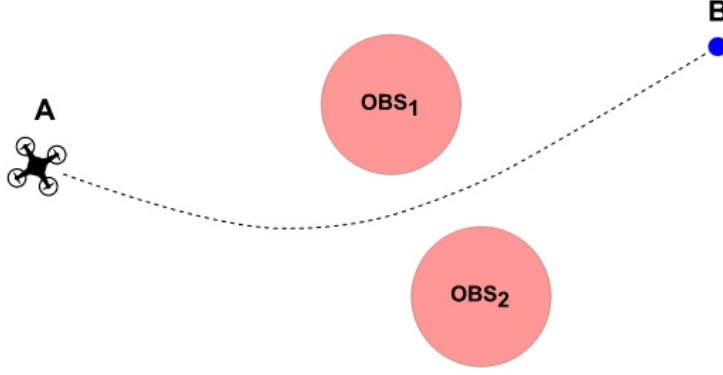


Fig. 1 Problem scenario

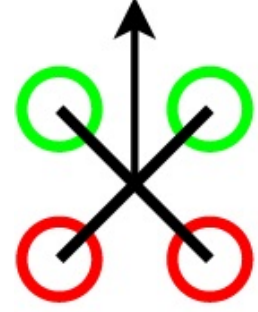


Fig. 2 UAV shape

Table 2 GENEX vs Widely Used Collision Avoidance Methods:
Fast (FS), Computationally Efficient (CE), Good Performance (GP), Easy Implementation (EI), Online Work (OL), Optimal (OP), Handles Local Minima Problem (HLMP)

Collision Avoidance Method	FS	CE	GP	EI	OL	OP	HLMP
GENEX	✓	✓	✓	✓	✓	✓	✓
Geometric	✓	✓	✓	✓	✓	✗	✓
Potential Fields	✓	✓	✓	✓	✓	✗	✗
Optimization-Based	✗	✗	✗	✗	✗	✓	✓
Probabilistic Modeling	✗	✗	✗	✗	✗	✓	✓

IV. Static obstacle avoidance of UAV

In this section, we present brief overview of GENEX guidance law and obstacle avoidance strategy.

A. Brief overview of GENEX

The geometry for the GENEX guidance law is explained in Figure 3. This figure is drawn similarly as given in [12](renaming the predicted intercept point as the desired destination point (DDP)), where the GENEX guidance law was designed for target interception problems. We explain this figure in the context of UAV. In this figure, the UAV's current and final velocity (at DDP) is defined by V and V_f , respectively. ψ and ψ_f are the current and final heading of the UAV, respectively. μ is the difference between the current and final heading. R is the relative distance between the current and final positions. Finally, δ is the angle between V and R . The guidance command of GENEX [12] is given by

$$u = \frac{1}{T^2} [K_1 (R_f - R - VT) + K_2 (V_f - V) T] \quad (1)$$

where, R_f is the position of the desired destination point (DDP), R is the current position of the UAV, T is time-to-go. K_1 and K_2 are optimal gain.

B. Obstacle avoidance strategy

To avoid the obstacles, a safe waypoint for the UAV will be around the middle of the separation between the obstacles OBS_1 and OBS_2 (we assume that there is adequate space to go through). The middle point is defined as the GAP middle point (GMP, shown as a blue dot in Figure 4 for a two-obstacle scenario). The radius of the obstacles OBS_1 and OBS_2 are R_1 and R_2 respectively ($R_1 \neq R_2$). Due to the different values of the radius of the obstacles, the position of the GMP is not in the middle of the centres (C_1 and C_2) of the two circles. The calculations of GAP and GMP for two and three-dimensional scenarios are shown in the following subsections.

1. 2D scenario

In 2D scenario, location of the centers C_1 and C_2 are (x_{c1}, y_{c1}) and (x_{c2}, y_{c2}) respectively.

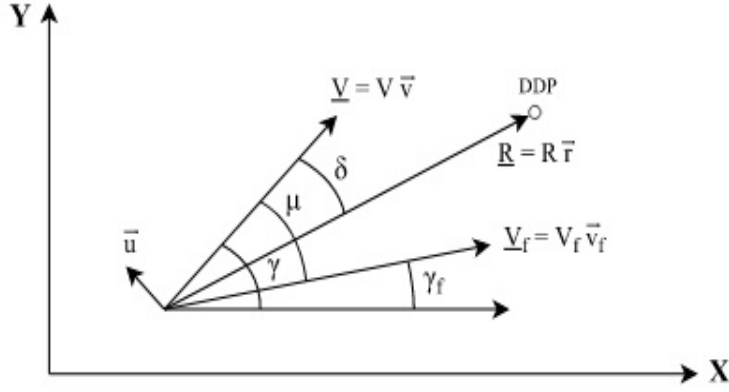


Fig. 3 Geometry for GENEX guidance law [12]

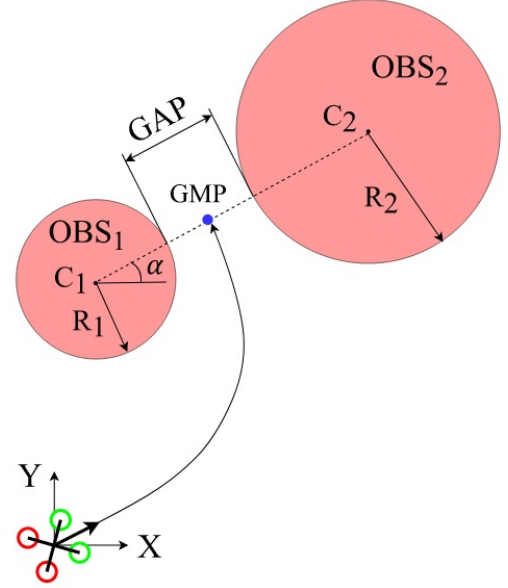


Fig. 4 GAP middle point (GMP) diagram

The distance between the centers of both obstacles is defined using the Pythagorean theorem as follows.

$$dist_{2D} = \sqrt{(x_{c1} - x_{c2})^2 + (y_{c1} - y_{c2})^2} \quad (2)$$

This distance can be also computed as the sum of the radius of both objects plus the gap between them.

$$dist_{2D} = R_1 + R_2 + GAP_{2D} \quad (3)$$

Therefore,

$$R_1 + R_2 + GAP_{2D} = \sqrt{(x_{c1} - x_{c2})^2 + (y_{c1} - y_{c2})^2} \quad (4)$$

$$GAP_{2D} = \sqrt{(x_{c1} - x_{c2})^2 + (y_{c1} - y_{c2})^2} - R_1 - R_2 \quad (5)$$

Therefore, GMP_{2D} can be computed as follows.

$$GMP_{2D} = (x_{c1}, y_{c1}) + \left(R_1 + \frac{GAP}{2} \right) \cdot (\cos(\alpha), \sin(\alpha)) \quad (6)$$

where,

$$\cos(\alpha) = \frac{x_{c2} - x_{c1}}{dist_{2D}} \quad \sin(\alpha) = \frac{y_{c2} - y_{c1}}{dist_{2D}} \quad (7)$$

It can be mentioned that, a general way of computing the GMP between i^{th} and j^{th} circle can be given as follows.

$$GMP_{2D} = (x_{ci}, y_{ci}) + \left(R_i + \frac{GAP_{2D}}{2} \right) \cdot (\cos(\alpha), \sin(\alpha)) \quad (8)$$

where,

$$\cos(\alpha) = \frac{x_{cj} - x_{ci}}{dist_{2D}} \quad \sin(\alpha) = \frac{y_{cj} - y_{ci}}{dist_{2D}} \quad (9)$$

2. 3D scenario

The same procedure is followed to obtain the GMP in 3D space. However, the z coordinate is now considered. The location of the centers C_1 and C_2 are (x_{c1}, y_{c1}, z_{c1}) and (x_{c2}, y_{c2}, z_{c2}) respectively. First, the 3D distance between the centres of both obstacles is defined using the Pythagorean theorem as follows:

$$dist_{3D} = \sqrt{(x_{obs,1} - x_{obs,2})^2 + (y_{obs,1} - y_{obs,2})^2 + (z_{obs,1} - z_{obs,2})^2} \quad (10)$$

This distance can be also obtained as the sum of the radius of both obstacles plus the gap between them.

$$dist_{3D} = R_1 + R_2 + GAP_{3D} \quad (11)$$

Therefore, solving Equation 10 and 11 the minimum separation between the obstacles GAP_{3D} can be defined as:

$$R_1 + R_2 + GAP_{3D} = \sqrt{(x_{obs,1} - x_{obs,2})^2 + (y_{obs,1} - y_{obs,2})^2 + (z_{obs,1} - z_{obs,2})^2} \quad (12)$$

$$GAP_{3D} = \sqrt{(x_{obs,1} - x_{obs,2})^2 + (y_{obs,1} - y_{obs,2})^2 + (z_{obs,1} - z_{obs,2})^2} - R_1 - R_2 \quad (13)$$

Once the GAP_{3D} is obtained, the GMP_{3D} can be computed as follows:

$$GMP_{3D} = \begin{bmatrix} x_{obs,1} \\ y_{obs,1} \\ z_{obs,1} \end{bmatrix} + \left(R_1 + \frac{GAP_{3D}}{2} \right) \cdot \begin{bmatrix} \cos(\alpha_v) \cdot \cos(\alpha_h) \\ \cos(\alpha_v) \cdot \sin(\alpha_h) \\ \sin(\alpha_v) \end{bmatrix} \quad (14)$$

Where α_h and α_v are the horizontal and vertical angle of the straight line connecting the centres of both obstacles:

$$\alpha_h = \frac{y_{obs,1} - y_{obs,2}}{x_{obs,1} - x_{obs,2}} \quad \alpha_v = \frac{z_{obs,1} - z_{obs,2}}{\sqrt{(x_{obs,1} - x_{obs,2})^2 + (y_{obs,1} - y_{obs,2})^2}} \quad (15)$$

Finally, a general way of computing the GMP_{3D} between i^{th} and j^{th} obstacle can be given by:

$$GMP_{3D} = \begin{bmatrix} x_{obs,i} \\ y_{obs,i} \\ z_{obs,i} \end{bmatrix} + \left(R_i + \frac{GAP_{3D}}{2} \right) \cdot \begin{bmatrix} \cos(\alpha_v) \cdot \cos(\alpha_h) \\ \cos(\alpha_v) \cdot \sin(\alpha_h) \\ \sin(\alpha_v) \end{bmatrix} \quad (16)$$

Where,

$$GAP_{3D} = \sqrt{(x_{obs,i} - x_{obs,j})^2 + (y_{obs,i} - y_{obs,j})^2 + (z_{obs,i} - z_{obs,j})^2} - R_i - R_j \quad (17)$$

$$\alpha_h = \frac{y_{obs,i} - y_{obs,j}}{x_{obs,i} - x_{obs,j}} \quad \alpha_v = \frac{z_{obs,i} - z_{obs,j}}{\sqrt{(x_{obs,i} - x_{obs,j})^2 + (y_{obs,i} - y_{obs,j})^2}} \quad (18)$$

3. Orientation of velocity vector at GMP

Along with the location of GMP, we need to calculate the orientation of the final velocity vector V_f , i.e., the UAV velocity heading (ψ_f) at the GMP, also referred to as the UAV final heading angle. In Figure 5, the UAV trajectories for eight different final heading angles ($\psi_{f,i}$) are presented. Although there exists multiple ψ_f values one can choose from, we have selected the one which is perpendicular to the line connecting the centres of the obstacles because it allows the UAV to naturally avoid the obstacles without the need of computing extra accelerations to ensure the safety of the UAV.

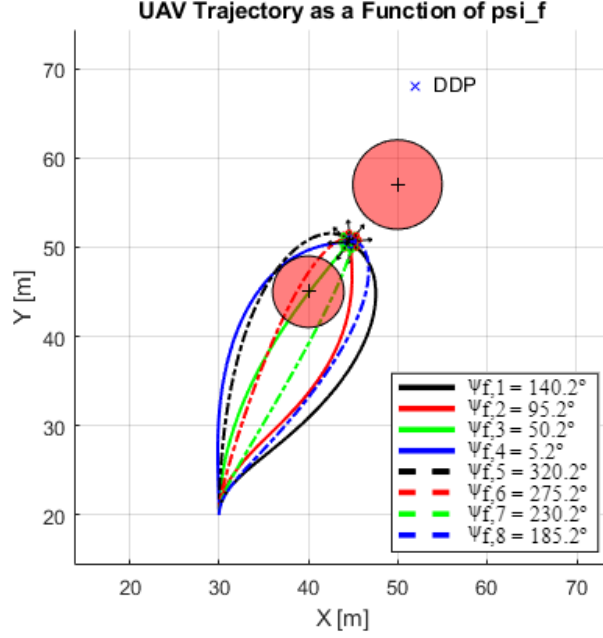


Fig. 5 UAV trajectory as a function of ψ_f

The calculation of ψ_f is shown in Figure 6. First, we join the two centres (C_1 and C_2) with a straight line which intersects both circles at points P_1 and P_2 (shown as blue dots in Figure 6). After that, we obtained the circle tangent at one of that points (P_1 in this case). Finally, ψ_f is defined as the tangent angle with respect to the horizontal. It can be seen that there exist two possible values of ψ_f . They are defined as follows.

$$\psi_{f,1} = \alpha + 90^\circ \quad \psi_{f,2} = \alpha - 90^\circ \quad (19)$$

Where α can be defined using Equation 9 as

$$\alpha = \arctan \left(\frac{y_{c2} - y_{c1}}{x_{c2} - x_{c1}} \right) \quad (20)$$

As a result of this operations, the following UAV trajectory in Figure 7 is obtained ($\psi_{f,1}$ has been selected since it is the

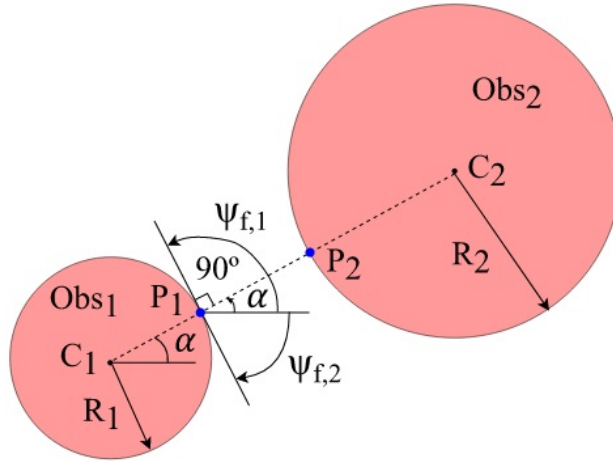


Fig. 6 UAV final velocity heading (ψ_f) computation diagram

more natural trajectory).

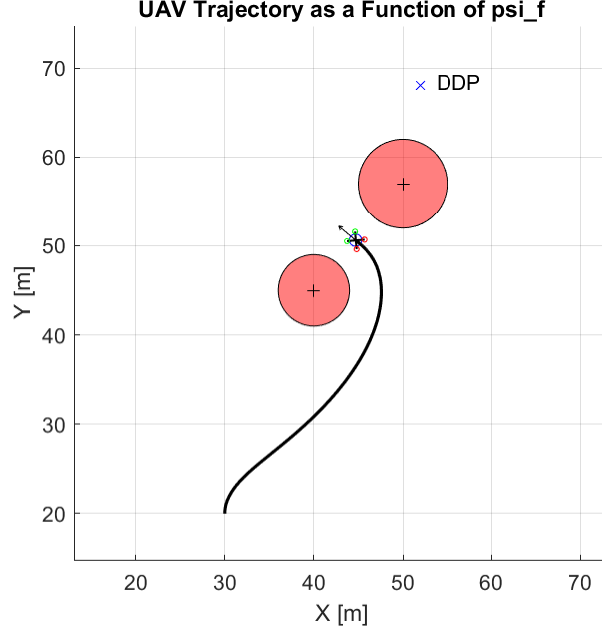


Fig. 7 UAV Selected Trajectory with $\psi_{f,1} = \alpha + 90^\circ = 140.2^\circ$

When considering a 3D space scenario, the vertical plane needs to be considered. Therefore, ψ_f along with γ_f need to be defined. In 3D space, ψ_f is defined exactly as the one for 2D space. Therefore, it also takes the two possible values defined in Equation 19 since the definition of α is also equal to the definition of α_h in Equation 15.

There exist multiple ways to define the final UAV pitch angle γ_f when reaching the GMPs. If one of the values of ψ_f is selected and fixed, then a plane of possible γ_f angles exists to choose from. The easiest way to define it is to select it by hand, considering that the obstacles are static and that the user can shape the UAV trajectory through them as desired. On the other hand, another interesting and more autonomous method is to select γ_f as the vertical angle between the current target and the next one, i.e., the vertical LOS angle between the current GMP and the next one. In Equation 21, the γ_f of the (i^{th}) GMP is defined using this last method.

$$\gamma_{f,(i)} = \frac{z_{GMP,(i+1)} - z_{GMP,(i)}}{\sqrt{(x_{GMP,(i+1)} - x_{GMP,(i)})^2 + (y_{GMP,(i+1)} - y_{GMP,(i)})^2}} \quad (21)$$

Where i takes the natural numbers from 1 to the number of GMPs (N_{GMPs}). To compute the value of γ_f for the last GMP ($\gamma_f(N_{GMPs})$) the position of the $GMP(N_{GMPs} + 1)$ is needed. This position is the DDP position.

4. Barrier function for safety

Usually, the UAV trajectory can be shaped using the final heading angle so that it avoids obstacles naturally. However, in some cases, the UAV trajectory gets too close to an obstacle. In these cases, the barrier function concept has been used to ensure the safety of the UAV. A barrier function is a continuous function whose value grows until infinity when it gets closer to a user-defined boundary [14]. Figure 8, extracted from [15], shows a barrier function graph as a function of the user-defined parameter μ . As μ gets bigger, the slope of the function gets steeper when it gets below the user-defined boundary Q . In this paper, extra acceleration commands are computed using a logarithmic barrier function in their definition to ensure the safety of the UAV. These extra commands allow the UAV to shape its trajectory around a specific obstacle to avoid it. It is important to highlight that they are just computed if the relative distance between the UAV and a particular obstacle is lower than the safety radius R_s .

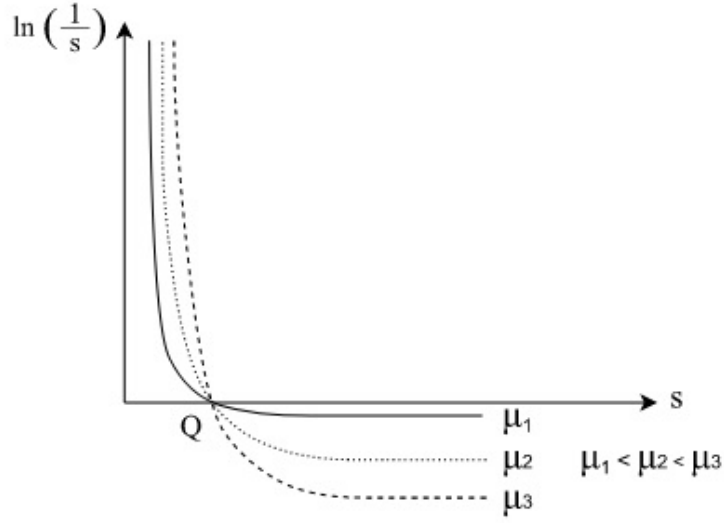


Fig. 8 Barrier function graph as a function of μ

The 2D barrier functions definition is given by Equation 22.

$$a_{x,extra} = \mu_i \cdot \rho \cdot \ln \left(\frac{Rs_i}{|(Rx_{rel,i} - R_i \cdot \cos(\sigma_i))|} \right) \quad (22)$$

$$a_{y,extra} = \mu_i \cdot \rho \cdot \ln \left(\frac{Rs_i}{|(Ry_{rel,i} - R_i \cdot \sin(\sigma_i))|} \right)$$

Where μ_i is the user-defined parameter for the i^{th} obstacle, ρ is a global user-defined parameter that affects every obstacle. Rs_i is the safety radius that the UAV considers when avoiding the i^{th} obstacle. Rs_i is also known as the user-defined boundary (Q in Figure 8). $[Rx_{rel,i}, Ry_{rel,i}]$ are the components of the relative distance between the UAV and the i^{th} obstacle. R_i is the radius of the i^{th} obstacle. σ_i is the LOS angle between the UAV and the i^{th} obstacle.

These extra acceleration commands are defined in the inertial reference frame in Equation 22. Therefore, they need to be transferred to the velocity reference frame to be added to the main commands computed by the GENEX guidance law.

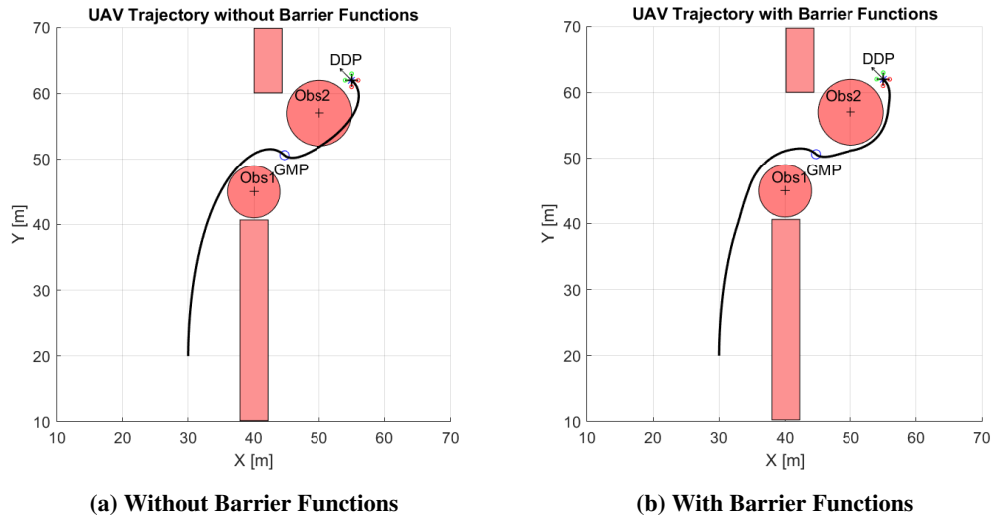


Fig. 9 UAV Trajectory without/with Barrier Functions.

Next, an example is given where the barrier functions are needed. Consider the scenario described in Figure 9 where two walls have been added so that the UAV must follow a particular trajectory. It can be seen how the trajectory followed by the UAV gets remarkably close to Obs_1 and inside Obs_2 if the barrier function concept is not used (Figure 9 *a*). On the other hand, the UAV reaches the DDP without colliding against any obstacle if a logarithmic barrier function like the one in Equation 22 is used with the following parameters (Figure 9 *b*)

$$\begin{cases} Rs_1 = 5m \\ Rs_2 = 9m \end{cases} \quad \begin{cases} \mu_1 = 1.75 \\ \mu_2 = 6.5 \end{cases} \quad \rho = 2 \quad (23)$$

This tuning is just required for some special GENEX cases like the one in Figure 9. Therefore, it can be done offline before the UAV flight starts since we are dealing with static obstacles.

V. Implementation

The pseudocode of the algorithm implementation is shown in Algorithm 1. There exit two loops. The outer loop iterates through each GMP. Once the UAV reaches the current GMP, the next one is computed and set as the UAV's new desired destination. Inside the inner loop is where the GENEX acceleration command is computed and used to update the system model of the UAV.

Algorithm 1 Main Pseudocode

```

1: Initialisation
2: for  $i \leftarrow 1, n_{GMP}$  do
3:   Calculate_GMP_Initial_Parameters
4:   while  $t_{go} > 0.01$  do
5:     Update_System_Model
6:     Calculate_Relative_Dynamics
7:     Calculate_Acceleration_Command
8:   end while
9: end for

```

Using this algorithm, we have performed different simulations considering two obstacles (one GMP) and multiple obstacles in the two-dimensional space. The 3D GENEX main algorithm (Algorithm 1) has the same structure as the 2D GENEX one. The only difference is that the 3D version has more parameters to compute because the vertical plane is considered.

VI. Simulation Results and Discussions

A. UAV mathematical model

In this section, the mathematical model of the UAV is presented in 2D and 3D. It can be mentioned that the kinematic models are considered for simulation. Also, we have considered the velocity of the UAV to be fixed.

1. 2D scenario

Mathematical model of UAV in 2D is given as follows.

$$x = \begin{bmatrix} x_x \\ x_y \end{bmatrix}, \quad v = \frac{dx}{dt} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} v \cos \psi \\ v \sin \psi \end{bmatrix} \quad (24)$$

$$\psi = \arctan\left(\frac{v_y}{v_x}\right) \quad (25)$$

$$\dot{\psi} = \frac{d\psi}{dt} = \frac{u}{v} \quad (26)$$

Along the inertial x and y axes, position and velocity components are given as $x = \begin{bmatrix} x_x \\ x_y \end{bmatrix}$ and $v = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$ respectively. ψ denotes the heading angle.

2. 3D scenario

Mathematical model of UAV in 3D is given as follows.

$$x = \begin{bmatrix} x_x \\ x_y \\ x_z \end{bmatrix}, \quad v = \frac{dx}{dt} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v \cdot \cos \gamma \cdot \cos \psi \\ v \cdot \cos \gamma \cdot \sin \psi \\ v \cdot \sin \gamma \end{bmatrix} \quad (27)$$

$$\psi = \arctan\left(\frac{v_y}{v_x}\right), \quad \gamma = \arctan\left(\frac{v_z}{v_h}\right) = \arctan\left(\frac{v_z}{\sqrt{v_x^2 + v_y^2}}\right) \quad (28)$$

$$\dot{\psi} = \frac{d\psi}{dt} = \frac{u_h}{v \cdot \cos \gamma}, \quad \dot{\gamma} = \frac{d\gamma}{dt} = \frac{u_v}{v} \quad (29)$$

Along the inertial x , y , and z axes, position and velocity components are given as $x = \begin{bmatrix} x_x \\ x_y \\ x_z \end{bmatrix}$ and $v = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$ respectively. ψ, γ denote the heading and flightpath angle respectively.

B. Static obstacle avoidance: 2D scenario

1. Two obstacles

Here, two obstacles are considered for the simulation study. The initial condition of the UAV is $x = [0, 0]m$. The initial velocity and heading are $10m/s$ and 90 deg, respectively. The final position is $x_{DDP} = [25, 45]$, and the final heading angle is 0 deg. The centers of the obstacles are located at $[20, 27]m$, and $[10, 25]m$ with radius $5m$ and $4m$ respectively. In Figure 10, six frames have been extracted from the UAV trajectory through the two obstacles (red circles). As planned, the UAV reaches the DDP (blue cross) while successfully avoiding the obstacles. To avoid both obstacles, it goes between them through the GMP (small blue circumference).

Figure 11a shows the generated acceleration command u . The spike at $t = 6.31$ s is due to the UAV reaching the GMP1 and suddenly changing its target from the GMP1 to the DDP. The heading angle of the UAV during the flight is shown in Figure 11b. It can be observed that the heading angle achieved at DDP by the UAV is 0 deg. Therefore, the UAV satisfies the terminal constraints.

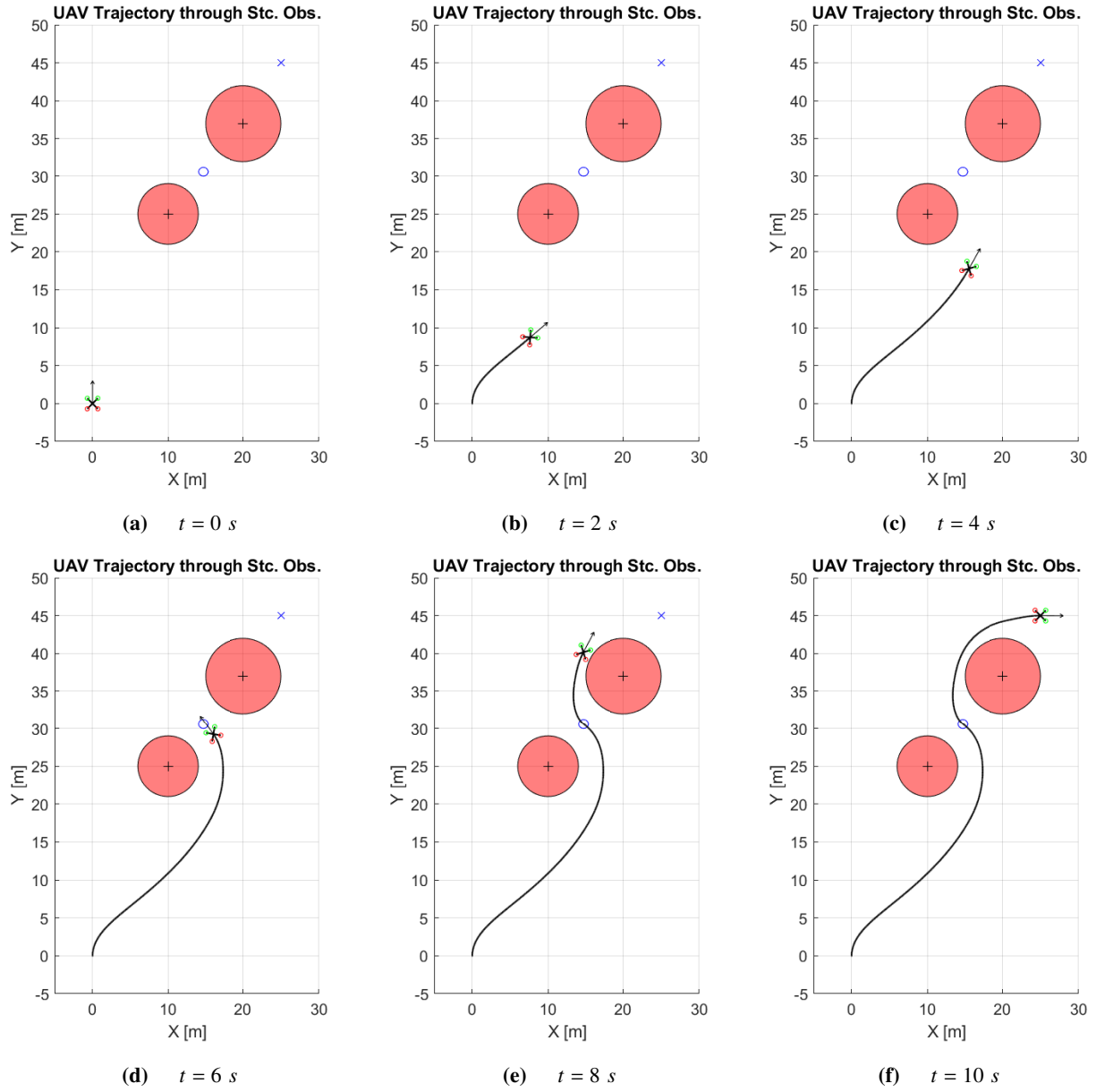


Fig. 10 UAV trajectory through two obstacles

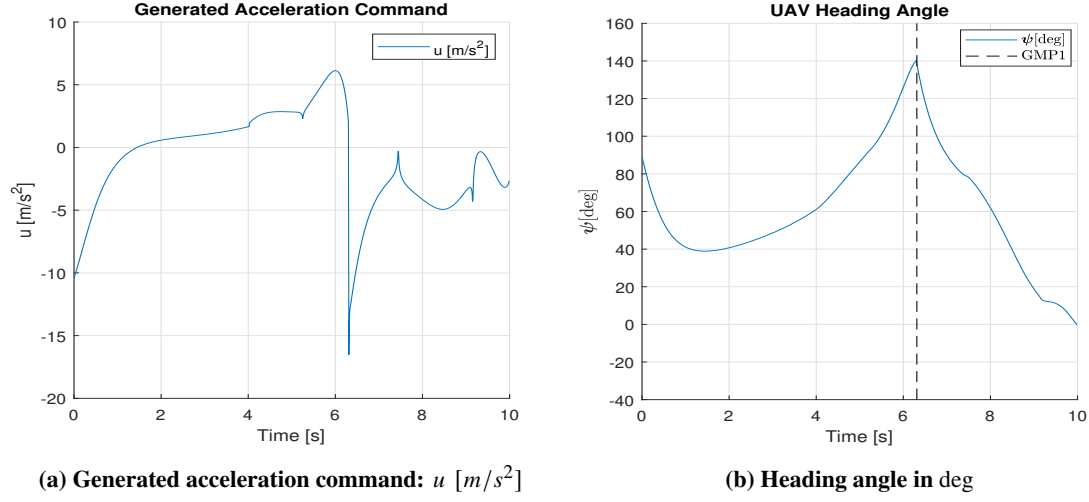


Fig. 11 Acceleration and heading angle of the UAV.

2. Multiple obstacles

Regarding the multiple obstacle simulation, we consider five obstacles. The initial position of the UAV is $x = [0, 0]m$. The initial velocity and heading are $10m/s$ and 90 deg, respectively. The final position is $x_{DDP} = [35, 35]$, and the final heading angle is 0 deg. The details of the obstacles are given in Table 3.

Table 3 Laptop Specifications

Obstacles	1	2	3	4	5
x_{obs}	20	10	15	-7	30
y_{obs}	27	25	10	20	25
r	5	4	11	7	5

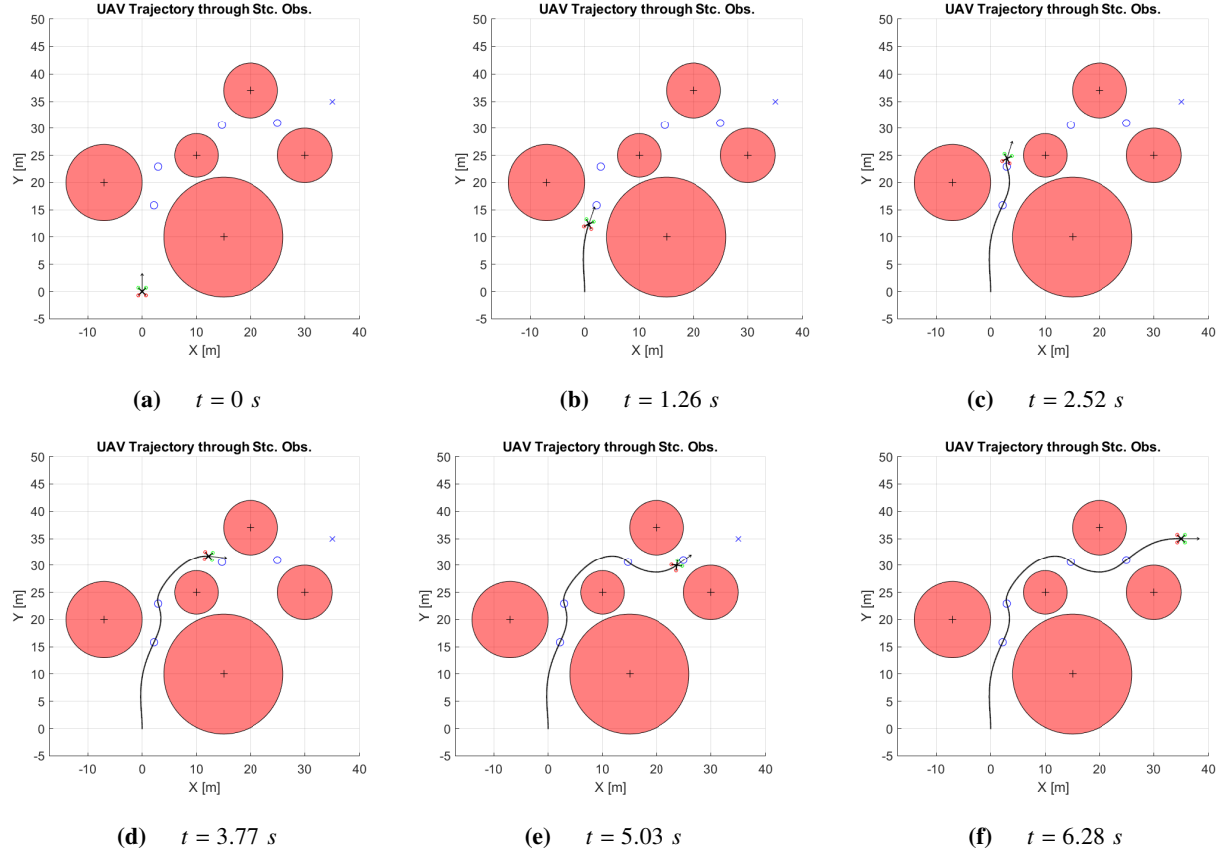


Fig. 12 UAV trajectory through multiple obstacles

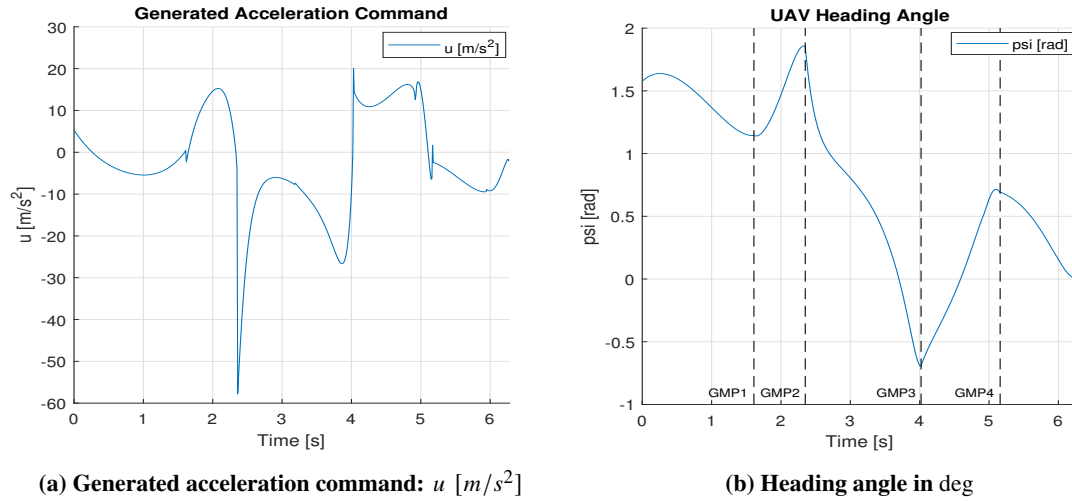


Fig. 13 Acceleration and heading angle of the UAV.

In Figure 12, six frames have been extracted from the UAV trajectory through the five obstacles. Once again, the UAV reaches the DDP while successfully avoiding the obstacles. To avoid them, it goes between them through the different GMPs. The acceleration command u generated by the algorithm is shown in Figure 13a. It can be observed that the acceleration reaches a maximum absolute value of -57.8 m/s^2 when $t = 2.36\text{ s}$. The heading angle during flight

is shown in Figure 13b. The UAV reaches the DDP with a heading angle of 0 deg, which satisfies the constraint.

C. Static obstacle avoidance: 3D scenario

1. Two obstacles

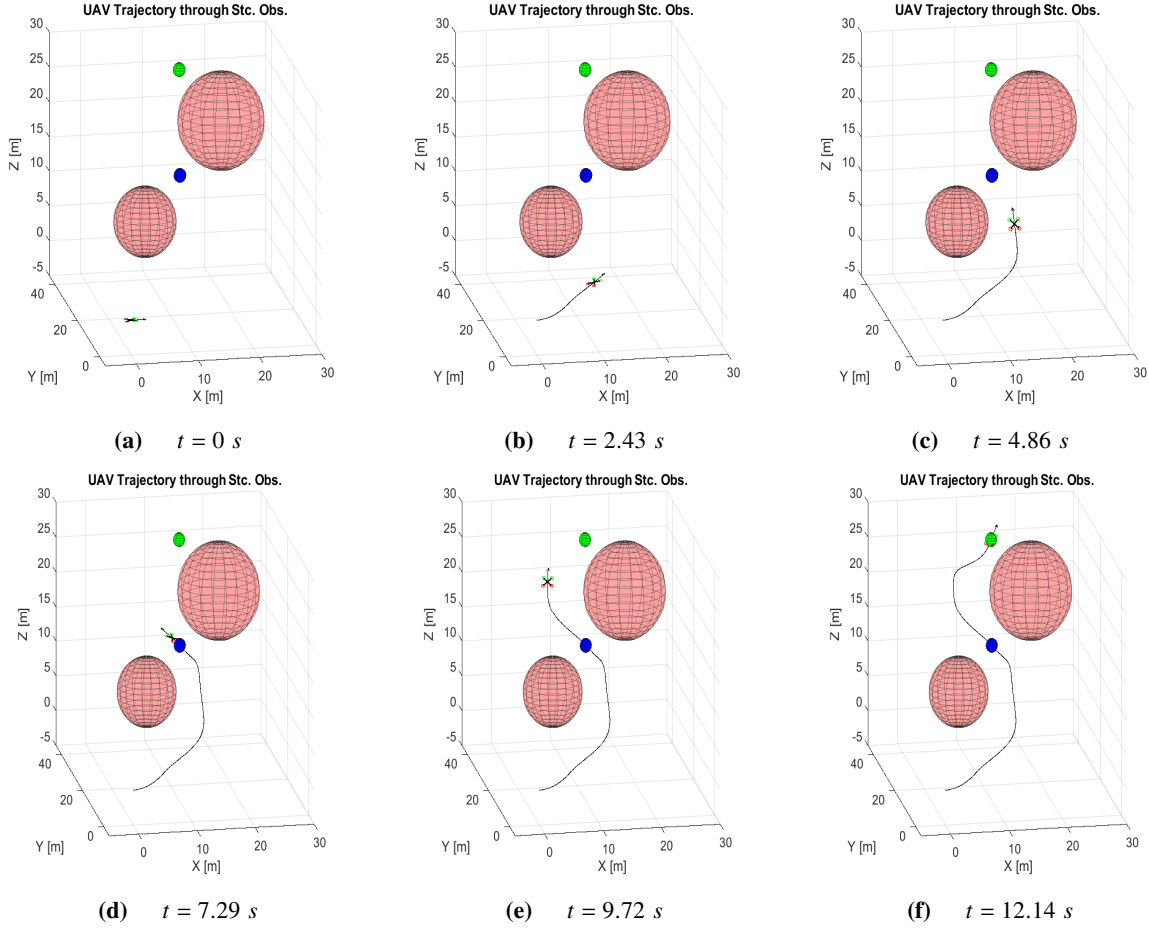


Fig. 14 UAV 3D trajectory through two obstacles.

Here, we will present the simulation results of collision avoidance in 3D. We first considered only two obstacles to demonstrate the algorithm. The initial condition of the UAV is $x = [0, 0, 0]m$. The initial velocity is $10m/s$. The flightpath and heading angles are 20 deg each. The final position is $x_{DDP} = [15, 45, 25]m$. The final flight path and the heading angle at DDP are 45 deg each. The centers of the obstacles are located at $[20, 30, 20]m$, and $[5, 15, 10]m$ with radius $7m$ and $5m$ respectively.

In Figure 14, six frames have been extracted from the UAV 3D trajectory through the two static obstacles (red spheres). As planned, the UAV reaches the DDP (green sphere) while successfully avoiding the static obstacles. To avoid both obstacles, it goes between them through the GMP (blue sphere).

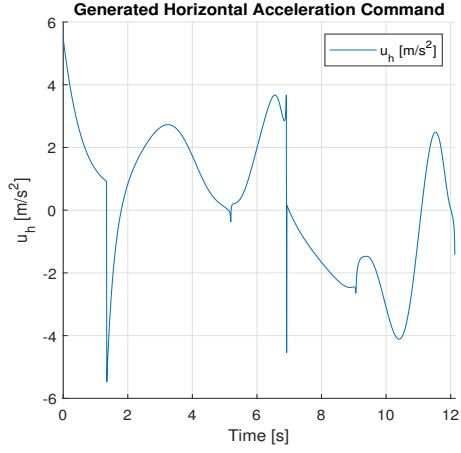


Fig. 15 Generated Horizontal Acceleration Command: u_h [m/s^2]

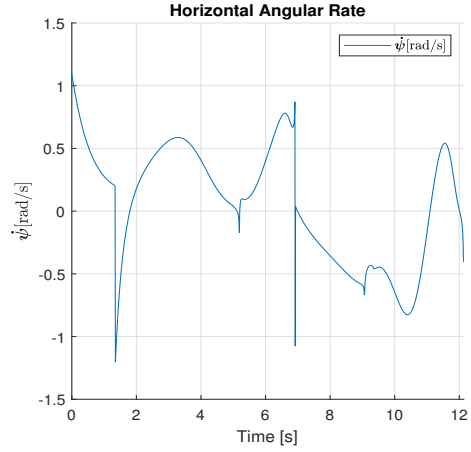


Fig. 16 Horizontal Angular Rate: $\dot{\psi}$ [rad/s]

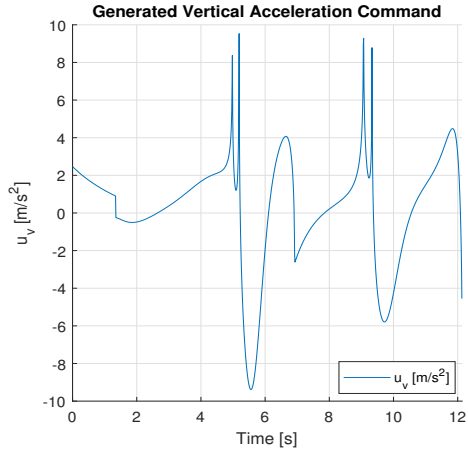


Fig. 17 Generated Vertical Acceleration Command: u_v [m/s^2]

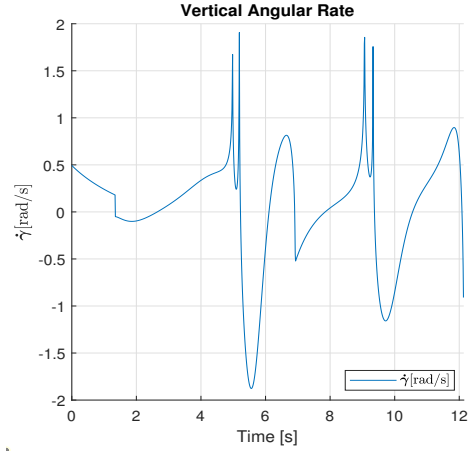


Fig. 18 Vertical Angular Rate: $\dot{\gamma}$ [rad/s]

Figures 15 and 17 show the horizontal and vertical generated acceleration commands u_h and u_v , respectively. On the other hand, Figs. 16 and 18 show the horizontal and vertical angular rates $\dot{\psi}$ and $\dot{\gamma}$, respectively. These angular rates are computed using Equation 29. However, for the 3D case, the horizontal angular rate depends also on the value of $\cos(\gamma)$.

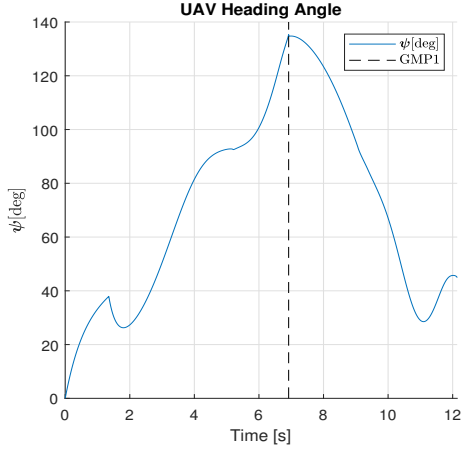


Fig. 19 UAV Heading Angle:
 ψ [deg]

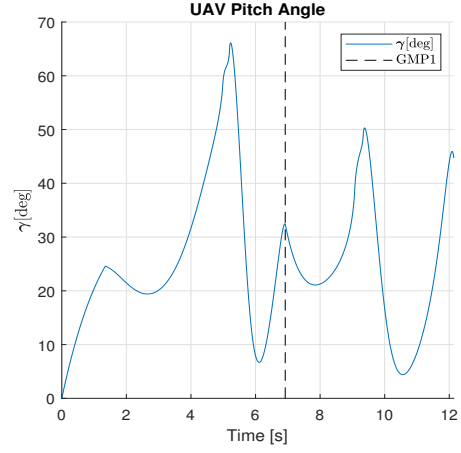


Fig. 20 UAV Pitch Angle:
 γ [deg]

Figures 19 and 20 show the UAV heading and pitch angle extracted from the simulation. It can be observed that the UAV reaches the DDP with the desired final angles values of $\psi_{f,DDP}$ and $\gamma_{f,DDP}$. Therefore, the UAV satisfies the terminal constraints.

2. Multiple obstacles

Here, we will present the simulation results of collision avoidance in 3D with multiple obstacles. We considered eight obstacles. The initial condition of the UAV is $x = [0, 0, 0]m$. The initial velocity is $5m/s$. The flightpath and heading angles are 20 deg and 20 deg, respectively. The final position is $x_{DDP} = [47, 47, 33]m$. The final flight path and the heading angle at DDP are 45 deg each. The centres and radius of the obstacles are given in Table 4.

Table 4 Laptop Specifications

Obstacles	1	2	3	4	5	6	7	8
$x_{obs}(m)$	38	0	20	12	23	22	33	44
$y_{obs}(m)$	38	20	10	28	24	23	33	44
$z_{obs}(m)$	33	15	0	20	10	28	17	20
$r(m)$	7	7	10	8	6	7	7	5

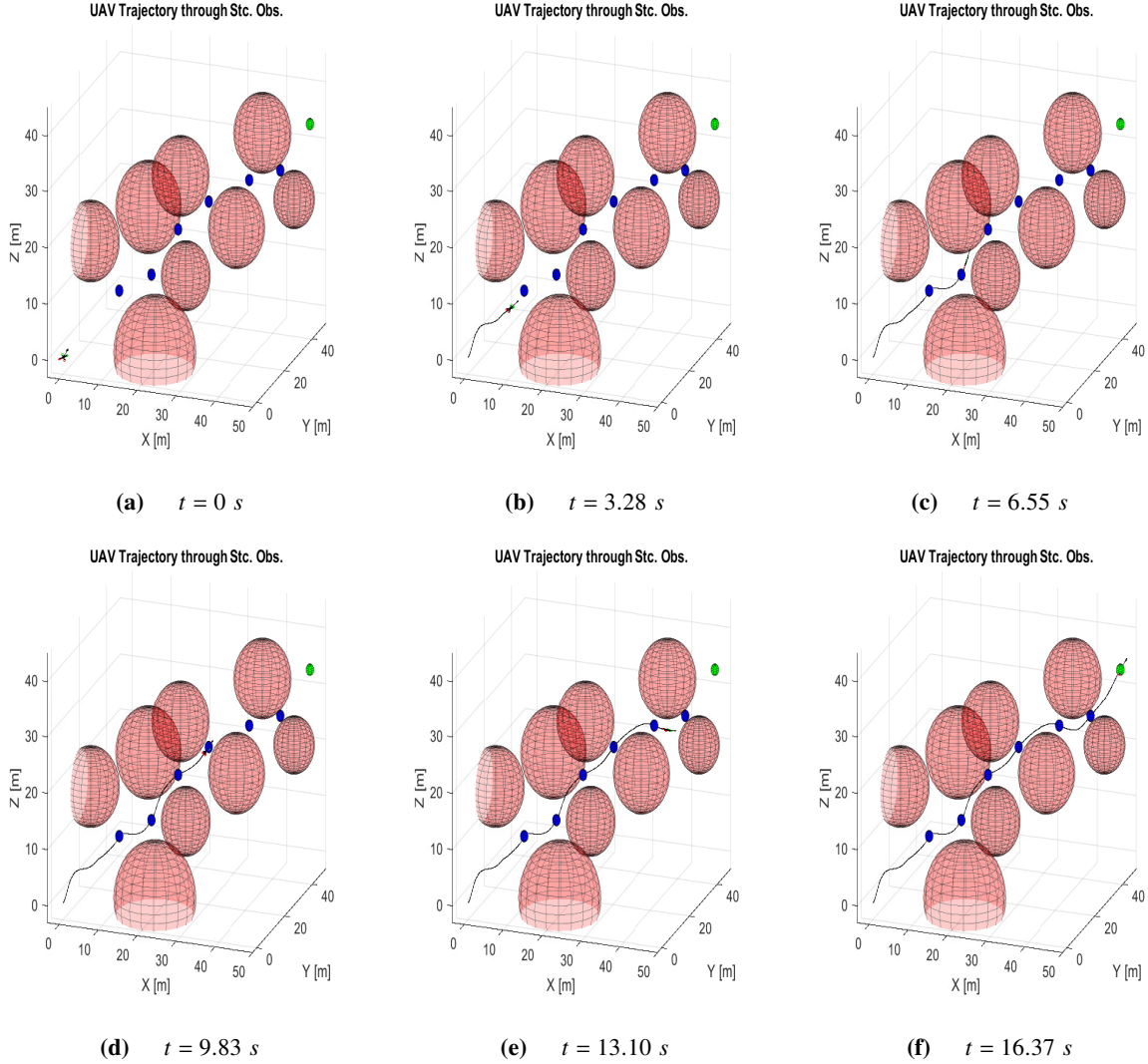


Fig. 21 UAV 3D trajectory through multiple obstacles.

In ??, six frames have been extracted from the UAV 3D trajectory through the eight static obstacles (red spheres). As planned, the UAV reaches the DDP (green sphere) while successfully avoiding the static obstacles. To avoid them, it goes between them through the different GMPs (blue spheres).

Figures 22 and 24 show the horizontal and vertical generated acceleration commands u_h and u_v , respectively. Both stay within a range of $\pm 25 \text{ m/s}^2$, and their values change very fast due to the increased number of obstacles in 3D space. On the other hand, Figures 23 and 25 show the horizontal and vertical angular rates $\dot{\psi}$ and $\dot{\gamma}$, respectively.

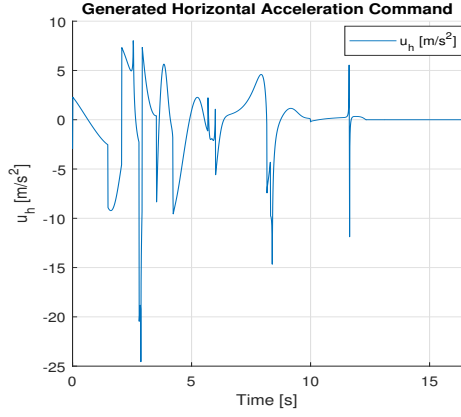


Fig. 22 Generated Horizontal Acceleration Command: u_h [m/s^2]

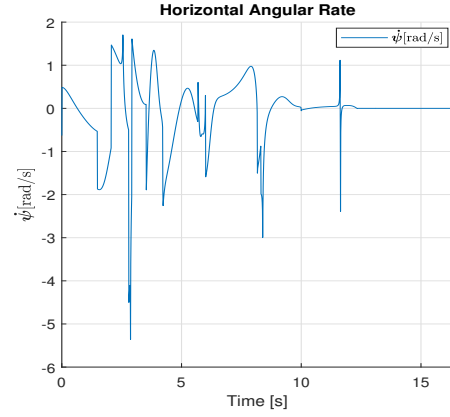


Fig. 23 Horizontal Angular Rate: $\dot{\psi}$ [rad/s]

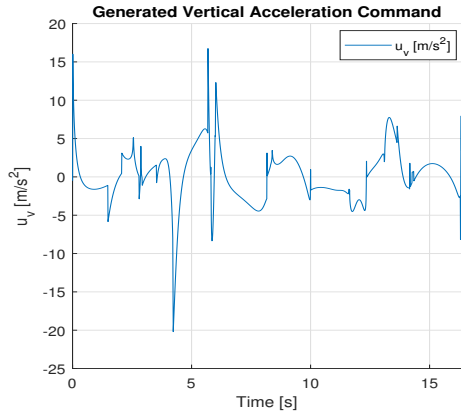


Fig. 24 Generated Vertical Acceleration Command: u_v [m/s^2]

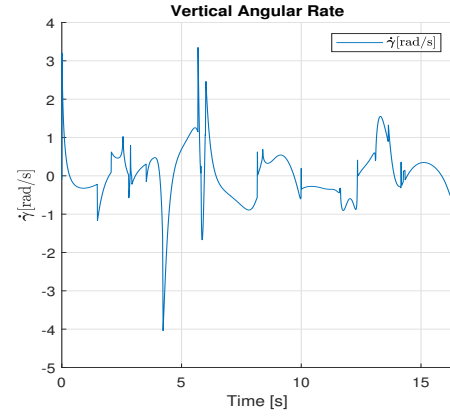


Fig. 25 Vertical Angular Rate: $\dot{\gamma}$ [rad/s]

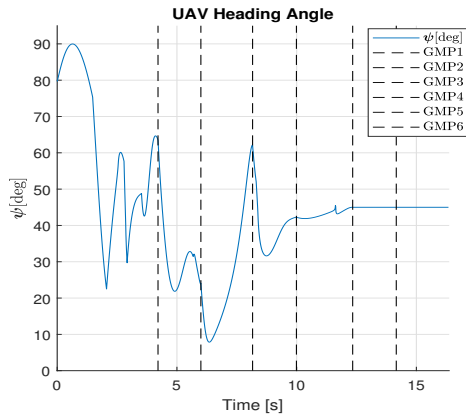


Fig. 26 UAV Heading Angle: ψ [deg]

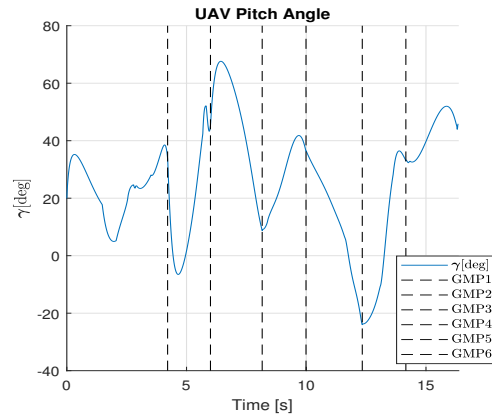


Fig. 27 UAV Pitch Angle: γ [deg]

Figures 26 and 27 show the UAV heading and pitch angle during the flight. It can be observed that the UAV reaches the DDP with a heading and flightpath angle each of 45 deg, which satisfies the constraint.

D. Real-Time Performance

The real-time performance of the design algorithm is tested in this section. During the development of the project, an ACER Aspire V3 – 575 Series laptop was used to run the simulations and test the real-time performance of the design algorithms. The laptop specifications can be found in Table 5. The operating system installed on this laptop is Windows. The operating system specifications can be found in Table 6.

Table 5 Laptop Specifications

Brand	ACER
Model	Aspire V-575 Series
Processor	Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz 2.60 GHz
Instaled RAM	16,0 GB (15,9 GB usable)
Type of Operating System	64 bits

Table 6 Operating System Specifications

Brand	Windows
Edition	Windows 10 Home
Version	21H2

To check the real-time performance, the amount of time that it takes the algorithm to compute the acceleration commands and the algorithm loop is analyzed. To do so, these computational times have been stored for each implemented simulation. This operation was repeated a hundred times to achieve more reliable data, and then the final mean was computed for each simulation. Finally, each implemented simulation plots the mean of both time values as a bar graph in Figure 28.

The first simulation corresponds to the two-obstacle scenario, and the second one to the multiple-obstacle scenario. As shown in the vertical axis label, the time unit used in this figure is microseconds. The percentage of the acceleration command computational time over the algorithm loop computational time is also shown for each implemented simulation.

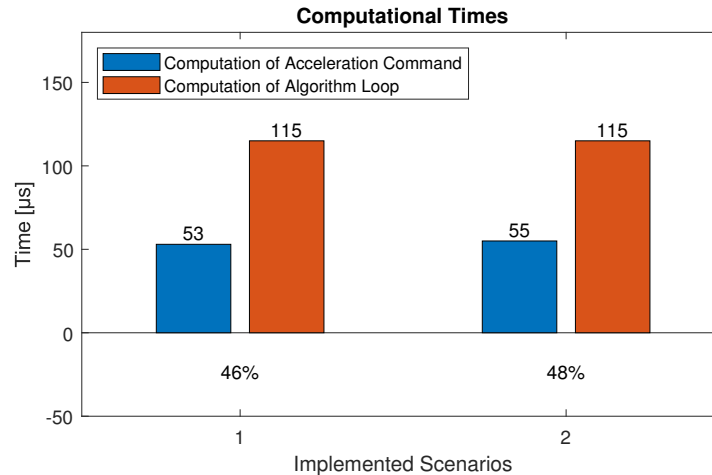


Fig. 28 Acceleration Command and Algorithm Loop Computational Times

It can be seen that the algorithm does not suffer from the number of obstacles since both algorithm loops have the same computational time for the different implementations. The computational time of the acceleration command differs so little (just $2\mu s$) that it seems to be pointing to a hardware issue. Finally, it needs to be mentioned that the

results are very promising if working with software, as the computational times are so fast that they are not even in the range of milliseconds. Therefore, if working with the software, they are more than suitable for real-time applications.

However, if these algorithms are going to be implemented in hardware, a deeper analysis needs to be made. In hardware, these algorithms will be executed inside a flight controller like the Pixhawk 6X in C/C++ instead of the MATLAB programming language. Although C/C++ are way faster than MATLAB, the obtained computational times will increase; in fact, they may even increase by an order of magnitude due to limitations in flight controller processors. However, the obtained computational times are so ridiculously short that even if the computational times increase by order of magnitude, the designed algorithms will still be more than suitable for real-time applications.

VII. Conclusion

The obstacles are successfully avoided using GENEX. The trajectory generated is optimal and shaped to satisfy the constraints at desired destination point (DDP). Closed-form expression of guidance command helps to implement it in real-time. Using guidance algorithms like the one designed in this paper, it is possible to extend UAV applications to civilian areas and operate UAVs alongside other manned aerial vehicles in the common airspace. In this study, we presented a two-dimensional scenario, and we will focus on extending this study to three-dimension and dynamic obstacles.

References

- [1] Wright, A., "Drones Offer Risks, Underwriting Challenges," 1 2015. URL <https://riskandinsurance.com/drones-offer-risks-underwriting-challenges/>.
- [2] Mujumdar, A., and Padhi, R., "Evolving philosophies on autonomous obstacle/collision avoidance of unmanned aerial vehicles," *Journal of Aerospace Computing, Information and Communication*, Vol. 8, 2011, pp. 17–41. <https://doi.org/10.2514/1.49985>.
- [3] Yasin, J. N., Mohamed, S. A., Haghbayan, M. H., Heikkonen, J., Tenhunen, H., and Plosila, J., "Unmanned Aerial Vehicles (UAVs): Collision Avoidance Systems and Approaches," *IEEE Access*, Vol. 8, 2020, pp. 105139–105155. <https://doi.org/10.1109/ACCESS.2020.3000064>.
- [4] Goss, J., Rajvanshi, R., and Subbarao, K., "Aircraft conflict detection and resolution using mixed geometric and collision cone approaches," 2004, pp. 670–689. <https://doi.org/10.2514/6.2004-4879>.
- [5] Seo, J., Kim, Y., Kim, S., and Tsourdos, A., "Collision Avoidance Strategies for Unmanned Aerial Vehicles in Formation Flight," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 53, 2017, pp. 2718–2734. <https://doi.org/10.1109/TAES.2017.2714898>.
- [6] Carbone, C., Ciniglio, U., Corrado, F., and Luongo, S., "A novel 3D geometric algorithm for aircraft autonomous collision avoidance," Institute of Electrical and Electronics Engineers Inc., 2006, pp. 1580–1585. <https://doi.org/10.1109/cdc.2006.376742>.
- [7] Azzabi, A., and Nouri, K., "Path planning for autonomous mobile robot using the Potential Field method," Institute of Electrical and Electronics Engineers Inc., 2017, pp. 389–394. <https://doi.org/10.1109/ASET.2017.7983725>.
- [8] Yu, T., Tang, J., Bai, L., and Lao, S., "Collision avoidance for cooperative UAVs with rolling optimization algorithm based on predictive state space," *Applied Sciences (Switzerland)*, Vol. 7, 2017. <https://doi.org/10.3390/app7040329>.
- [9] He, Z., Liu, C., Chu, X., Negenborn, R. R., and Wu, Q., "Dynamic anti-collision A-star algorithm for multi-ship encounter situations," *Applied Ocean Research*, Vol. 118, 2022. <https://doi.org/10.1016/j.apor.2021.102995>.
- [10] Baberg, F., and Ogren, P., "Formation Obstacle Avoidance using RRT and Constraint Based Programming," Institute of Electrical and Electronics Engineers Inc., 2017. <https://doi.org/10.1109/SSRR.2017.8088131>.
- [11] Temizer, S., Kochenderfer, M. J., Kaelbling, L. P., Lozano-Pérez, T., and Kuchar, J. K., "Collision Avoidance for Unmanned Aircraft using Markov Decision Processes," American Institute of Aeronautics and Astronautics, 2010. URL <http://hdl.handle.net/1721.1/61341>.
- [12] Ohlmeyer, E. J., and Phillips, C. A., "Generalized vector explicit guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 29, 2006, pp. 261–268. <https://doi.org/10.2514/1.14956>.
- [13] Mondal, S., and Padhi, R., "Generalized explicit guidance with optimal time-to-go and realistic final velocity," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 233, No. 13, 2019, pp. 4926–4942.

- [14] Nesterov, Y., *Lectures on Convex Optimization*, 2nd ed., 2018.
- [15] Mondal, S., and Padhi, R., “State and input constrained missile guidance using spectral model predictive static programming,” American Institute of Aeronautics and Astronautics Inc, AIAA, 2018. <https://doi.org/10.2514/6.2018-1584>.



DSSPACE

<https://dspace.org/>

Real-time collision avoidance trajectory planner using generalized vector explicit guidance

Subies Hueso, Josep; Mondal, Sabyasachi; Tsourdos, Antonios; Chadwick, Andrew

2023-01-19

American Institute of Aeronautics and Astronautics (AIAA)

<https://doi.org/10.2514/6.2023-1734>

Subies Hueso J, Mondal S, Tsourdos A, Chadwick A. (2023) Real-time collision avoidance trajectory planner using generalized vector explicit guidance. In: AIAA SciTech Forum 2023, 23-27 January 2023, National Harbor, Maryland, USA. Paper number AIAA 2023-1734