

Real-Time Path Planning Considering Static and Dynamic Obstacles

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

This paper introduces a real-time path planning strategy that effectively navigates around both static and dynamic obstacles. The approach combines the principles of Generalised Explicit Vector (GENEX) and Inverse Proportional Navigation (IPN), established algorithms in missile guidance known for their advantageous features such as low computation and closed-form expressions. Leveraging these attributes, the proposed strategy addresses path planning challenges involving static and dynamic obstacles. The performance of the combined algorithm is assessed through a comprehensive simulation study in both 2D and 3D scenarios, considering multiple static and dynamic obstacles.

I. Nomenclature

<i>GENEX</i>	=	generalized vector explicit guidance
<i>GMP</i>	=	gap middle point
<i>IPN</i>	=	Inverse proportional Navigation
<i>DDP</i>	=	Desired destination point
<i>UAV</i>	=	Unmanned Aerial Vehicle
<i>LOS</i>	=	Line of sight
<i>CAS</i>	=	Collision Avoidance System

II. Introduction

THE use of Unmanned Aerial Vehicles (UAVs) has experienced a rapid increase in recent times. However, the growing prevalence of UAVs brings with it a rise in potential hazards. Challenges arise in tasks conducted in civilian zones or shared airspace, given legal constraints and the risk of collisions. To enhance adaptability in such missions, UAVs need to skillfully navigate obstacles to reach their target or achieve a specific objective. This highlights the crucial need to incorporate a Collision Avoidance System (CAS) into their design, aiming to address and minimize these risks. Moreover, as the majority of UAV operations occur in outdoor settings with unpredictable and ever-changing conditions, the collision avoidance system must possess real-time capabilities and the ability to make on-the-spot calculations. This not only ensures the successful completion of the mission but also prioritizes the safety of both the UAV and its surrounding environment.

Numerous research endeavours in the literature have delved into addressing the collision avoidance problem, showcasing a variety of categories of methods. Geometric methods, a widely employed category, operate by calculating the distance between the UAV and obstacles based on their positions and velocities [1]. For instance, in [2], the collision cone approach, a prevalent geometric method, is applied to prevent collisions between two aircraft navigating a 3D environment. Similarly, [3] employs this approach for UAVs engaged in formation flight, and [4] utilizes it for pairwise non-cooperative aircraft scenarios. Another category of methods involves the potential field approach, which utilizes attractive and repulsive potentials to guide a UAV toward a target or repel it from obstacles [1]. In [5], this method is employed to navigate a mobile robot through a cluttered environment, while [6] adjusts real-time trajectories in 3D space using cooperative UAVs to avoid collisions. Optimization-based methods are focused on determining an optimal

*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: josepsubies@gmail.com

[†]Lecturer in Autonomous systems, 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.

or near-optimal path to a desired destination. For instance, in [7], an A* dynamic algorithm is developed for multiple ship encounters, while [8] proposes an advanced formation flying approach using the rapidly exploring random tree (RRT) algorithm to define waypoint nodes. Probabilistic modelling methods play a role in decision-making within dynamic environments, adapting strategies as threat information becomes available. In [9], a cost function is optimized to generate avoidance strategies.

The features of these methods are compared in Table 1 [1, 10]. A notable trade-off is observed between online efficiency and the generation of optimal paths. This is primarily due to the significant computational time typically required for a guidance algorithm to compute an optimal path. Consequently, optimal collision avoidance methods, such as optimization-based and probabilistic modelling, often conduct this computation offline. Conversely, geometric and potential field methods are characterized by quicker, computationally efficient processes that are easily implementable. Hence, for real-time computations, geometric and potential field methods prove more suitable compared to optimization-based and probabilistic modelling approaches. However, it's essential to carefully consider the potential field methods' local minima problem when opting for this approach over others. It is evident that methods such as geometric and

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	✗	✗	✗	✗	✗	✓	✓

potential fields exhibit the features outlined in the table but fall short in generating an optimal trajectory. Conversely, optimization-based methods and probabilistic modelling excel in producing optimal trajectories but may lack in other aspects. Emphasizing the importance of optimality is crucial, as it can enhance operation time or range for energy-limited platforms like UAVs. The Generalized Explicit Vector Guidance (GENEX) obstacle avoidance scheme [11] is noteworthy in addressing this trade-off between optimality and online efficiency. GENEX recognized as an optimal guidance algorithm employed in missile engagement problems [12, 13], meets all requirements. It simultaneously achieves design specifications regarding miss distance and the final UAV-target relative orientation. As previously mentioned, optimal path generation is vital for UAVs, considering their limited energy capacity. Furthermore, the guidance command offered by GENEX is straightforward to implement, computationally inexpensive, and calculated in closed form.

The methods discussed thus far are well-suited for avoiding static obstacles, as indicated by the features outlined in Table 1. However, finding an algorithm with these features that can effectively navigate dynamic obstacles poses a challenge. Fortunately, the Inverse PN or IPN algorithm emerges as a solution, representing a variant of PN specifically designed for interception or collision avoidance scenarios. In the IPN approach, both the interceptor and evader rotate at a proportional rate to, and in the opposite direction of, the Line of Sight (LOS) rate, as detailed in [14]. It is worth emphasizing that PN and IPN guidance laws can achieve optimality when the appropriate proportional gain is selected, as highlighted in [15]. In [16], a PN-based solution is proposed for neutralizing non-cooperative UAVs through high-speed interception using another UAV, presented in a clear and understandable manner. Regarding collision avoidance, [17] investigates PN for 3D collision avoidance, while [18] introduces an optimized 2D version of the same approach. The implementation of an algorithm to reduce collisions using the previously described IPN philosophy is detailed in [14]. Similarly, [19] presents and tests IPN for collision avoidance in high-density traffic. Notably, both [14] and [19] utilize the Dubins path (minimum time path) to reach the destination once the collision avoidance manoeuvre is complete. Lastly, [20] considers speed control and adherence to International Civil Aviation Organisation (ICAO) rules in the design of the IPN guidance law.

In this paper, we present a path planning and collision avoidance strategy where the GENEX and IPN are integrated to obtain an optimal path. We have evaluated the performance of this combined algorithm by a realistic simulation study in 2D and 3D scenario of multiple static and dynamic obstacles.

III. Problem Formulation

The problem formulation is presented here, wherein both static and dynamic obstacles are considered during path planning. The visual representation of this problem is depicted in Fig. 1. In this illustration, the UAV, initially positioned at the 'Start' point, is required to navigate from this point to the 'End.' The presence of static obstacles is denoted by 'Obs,' while dynamic obstacles **D** (primarily drones but potentially any moving objects with velocity) are also present. The UAV's objective is to avoid these obstacles and successfully reach the 'End' point.

It is assumed that the locations of the static obstacles are known and are approximated by circles with varying radii. The dynamic obstacles are in motion with constant heading and velocity, as indicated by the blue arrow. Additionally, it is assumed that the UAV is equipped with sensors capable of measuring the motion of dynamic obstacles.

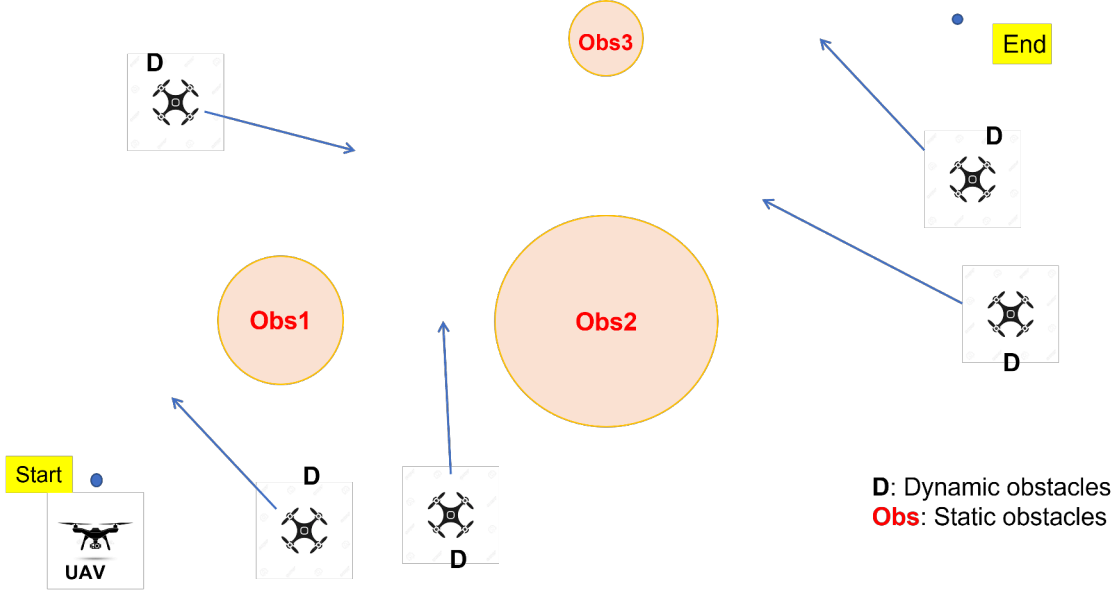


Fig. 1 Problem scenario with static and dynamic obstacles.

IV. Implementation

This section introduces the envisaged path planning strategy, designed to concurrently navigate around static and dynamic obstacles. The integration of the GENEX algorithm [12, 21] and the IPN algorithm is employed for the dual purposes of path planning and obstacle avoidance. It's noteworthy that the authors previously proposed a path planning approach for static obstacles utilizing GENEX and barrier functions, as outlined in [11]. For reference, a succinct overview of the static obstacle avoidance strategy is presented here.

A. Static obstacle avoidance strategy

To navigate around obstacles, a secure waypoint for the UAV is determined approximately at the midpoint between obstacles Obs_1 and Obs_2 , as illustrated in Figure 2 (assuming sufficient clearance for passage). This midpoint is referred to as the GAP_{2D} (shown as GAP in the figure) middle point (GMP_{2D} , shown as GMP in the figure), indicated by a blue dot in Figure 2 for a scenario involving two obstacles. The respective radii of obstacles Obs_1 and Obs_2 are denoted as R_1 and R_2 ($R_1 \neq R_2$). Owing to the distinct radii, the GMP is not positioned at the geometric center of the two circles' centers ($C_1(x_{c1}, y_{c1})$ and $C_2(x_{c2}, y_{c2})$).

Furthermore, the desired heading at the GMPs is computed using the geometric relationships outlined in Fig. 3. The distance between the centers of both obstacles is determined utilizing the Pythagorean theorem.

$$dist_{2D} = \sqrt{(x_{c1} - x_{c2})^2 + (y_{c1} - y_{c2})^2} \quad (1)$$

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

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

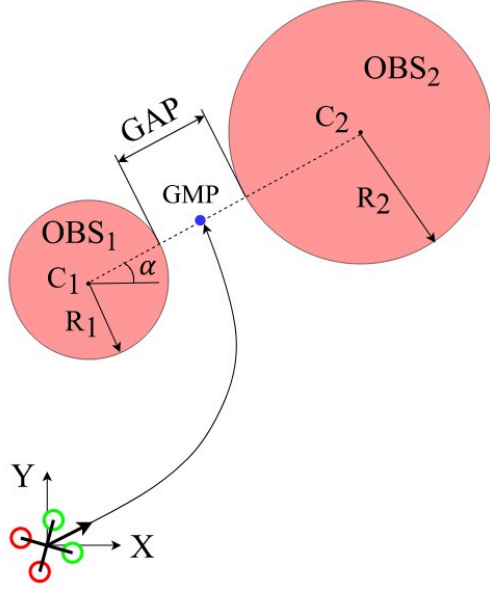


Fig. 2 GAP middle point (GMP) diagram. GENEX needs this information to avoid the static obstacles [11]

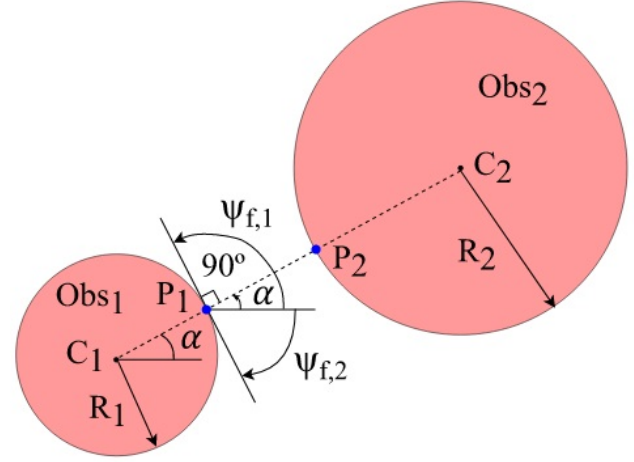


Fig. 3 Problem scenario with static and dynamic obstacles [11]

Therefore,

$$R_1 + R_2 + GAP_{2D} = \sqrt{(x_{c1} - x_{c2})^2 + (y_{c1} - y_{c2})^2} \quad (3)$$

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

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

$$GMP_{2D} = (x_{c1}, y_{c1}) + \left(R_1 + \frac{GAP}{2} \right) (\cos(\alpha), \sin(\alpha)) \quad (5)$$

where,

$$\cos(\alpha) = \frac{x_{c2} - x_{c1}}{dist_{2D}} \quad \sin(\alpha) = \frac{y_{c2} - y_{c1}}{dist_{2D}} \quad (6)$$

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

$$GMP_{2D} = (x_{ci}, y_{ci}) + \left(R_i + \frac{GAP_{2D}}{2} \right) (\cos(\alpha), \sin(\alpha)) \quad (7)$$

where,

$$\cos(\alpha) = \frac{x_{cj} - x_{ci}}{dist_{2D}} \quad \sin(\alpha) = \frac{y_{cj} - y_{ci}}{dist_{2D}} \quad (8)$$

In addition to determining the GMP's location, it is essential to compute the orientation of the final velocity vector V_f , i.e., the UAV velocity heading (ψ_f) at the GMP, commonly known as the UAV final heading angle. While there are multiple possible values for ψ_f to choose from, we opt for the one that is perpendicular to the line connecting the centers of the obstacles. This choice facilitates the UAV's natural avoidance of obstacles without requiring additional accelerations to ensure its safety.

The computation of ψ_f is illustrated in Figure 3. Initially, a straight line is drawn to connect the two centers (C_1 and C_2), intersecting both circles at points P_1 and P_2 (depicted as blue dots in Figure 3). Subsequently, a circle tangent to one of these points (specifically P_1 in this instance) is determined. Finally, ψ_f is defined as the tangent angle concerning the horizontal. It's noteworthy that there are two possible values for ψ_f , and they are defined as follows.

$$\psi_{f,1} = \alpha + 90^\circ \quad \psi_{f,2} = \alpha - 90^\circ \quad (9)$$

Where α can be defined using Equation 8 as

$$\alpha = \arctan \left(\frac{y_{c2} - y_{c1}}{x_{c2} - x_{c1}} \right) \quad (10)$$

The choice of $\psi_{f,1}$ is made as it corresponds to a more natural trajectory with respect to the UAV's starting position. With the waypoint GMP_{2D} and $\psi_{f,1}$ identified, these parameters serve as inputs for the GENEX algorithm. Subsequently, the guidance command for the UAV is computed using GENEX to effectively navigate and avoid static obstacles. Further details can be found in [11].

In this paper, we extend our previous to include the dynamic obstacles and explain how IPN along with GENEX can perform efficiently though simulation study. A brief overview of IPN is given as follows.

B. Inverse Proportional Navigation

As mentioned earlier, Proportional Navigation (PN) is widely employed in addressing missile guidance challenges. However, there is an extension of PN known as Inverse Proportional Navigation (IPN), specifically designed for collision avoidance [22]. While PN aims to bring the interceptor and evader on a collision course, IPN operates in the opposite manner. By applying an angular rate to adjust the Line-of-Sight (LOS) rate, IPN increases the relative distance rate between the UAV and the obstacle, diverting them away from a collision course and ensuring a successful collision avoidance manoeuvre. To be more precise, the guidance law generates an acceleration command that causes the UAV to rotate at a rate proportional to and in the opposite direction of the LOS rate [23]. In a 2D space, the acceleration command produced by the guidance law is defined as follows.

$$u = -Nv_c\dot{\sigma} \quad (11)$$

In this context, N represents the proportional gain, v_c denotes the closing velocity, and $\dot{\sigma}$ stands for the Line-of-Sight (LOS) rate. Concerning the three-dimensional (3D) space, the acceleration commands in the horizontal and vertical directions, generated by the guidance law, are expressed as:

$$u_h = -Nv_c\sigma_h \quad u_v = -Nv_c\dot{\sigma}_v \quad (12)$$

The commands generated are identical to those of PN but with a negative sign. Thus, by replacing the closing velocity (v_c) between the interceptor and evader with the negative relative distance rate ($-\dot{R}$) between them, the Inverse Proportional Navigation (IPN) acceleration commands take the following forms in 2D and 3D space, respectively:

$$u = N\dot{R}_{2D}\dot{\sigma} \quad (13)$$

$$u_h = N\dot{R}_{3D}\dot{\sigma}_h \quad u_v = N\dot{R}_{3D}\dot{\sigma}_v \quad (14)$$

These acceleration commands are used to compute the angular rates. These angular rates are defined in 2D and 3D space respectively as:

$$\dot{\psi}_{2D} = \frac{d\psi_{2D}}{dt} = \frac{u}{v} \quad (15)$$

$$\dot{\psi}_{3D} = \frac{d\psi_{3D}}{dt} = \frac{u_h}{v \cos \gamma} \quad \dot{\gamma}_{3D} = \frac{d\gamma_{3D}}{dt} = \frac{u_v}{v} \quad (16)$$

A summary of the inverse proportional navigation sufficient conditions has been performed. In 2D space, the collision avoidance manoeuvre is guaranteed if:

- The relative distance rate is positive: $\dot{R} > 0$. Thus, the UAV and the obstacle are moving away.
- The Line-Of-Sight rate is different from zero: $\dot{\sigma} \neq 0$. Thus, σ does not remain constant.

In 3D space, the collision avoidance manoeuvre is guaranteed if:

- The relative distance rate is positive: $\dot{R} > 0$. Thus, the UAV and the obstacle are moving away.
- The horizontal and/or vertical Line-Of-Sight rate are different from zero: $\dot{\sigma}_h \neq 0$ and/or $\dot{\sigma}_v \neq 0$. Thus, σ_h and/or σ_v do not remain constant.

In three-dimensional (3D) space, the occurrence of a UAV collision avoidance manoeuvre in a specific plane is indicated when only one of the Line-of-Sight (LOS) rates is non-zero. The subsequent subsections elaborate on the computation of essential parameters required for calculating the acceleration command in both 2D and 3D space: $\dot{R}_{2D}$, $\dot{R}_{3D}$, $\dot{\sigma}$, $\dot{\sigma}_h$, and $\dot{\sigma}_v$.

1. Relative Distance Rate ($\dot{R}$) Computation

The initial parameter to be determined is the relative distance rate ($\dot{R}$) between the UAV and the dynamic obstacle, and it is calculated for both the 2D and 3D scenarios. Initially, the 2D relative distance rate is computed. In Figure 4, a 2D schematic of the problem illustrates the UAV velocity vector v and its components $[v_x, v_y]$ using blue arrows. Simultaneously, the dynamic obstacle velocity vector v_{obs} and its components $[v_{obs,x}, v_{obs,y}]$ are represented with red arrows.

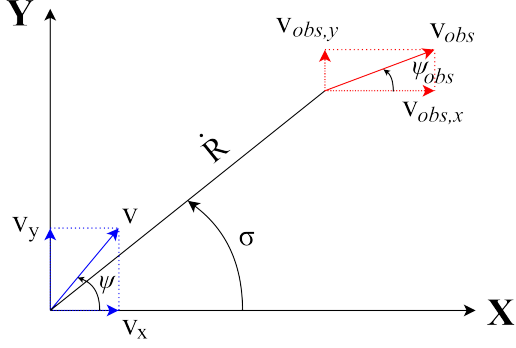


Fig. 4 Relative Distance Rate Schematics in 2D

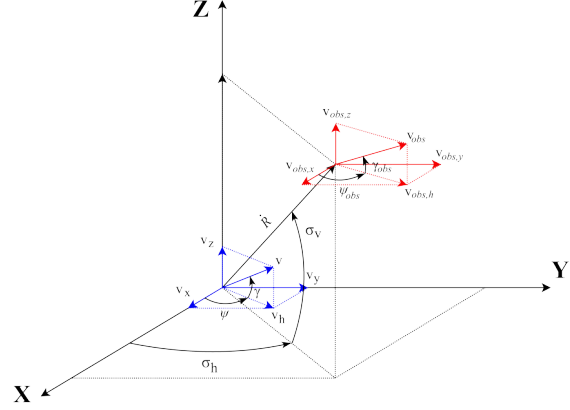


Fig. 5 Relative Distance Rate Schematics in 3D

Using this schematic diagram in Figure 4, the 2D relative distance rate ($\dot{R}_{2D}$; for convenience R is written as R_{2D}) is defined using the following equation:

$$\dot{R}_{2D} = v_{obs} \cos(\psi_{obs} - \sigma) - v \cos(\psi - \sigma) \quad (17)$$

For the calculation of the 3D relative distance rate, it is necessary to take the vertical plane into account. In Figure 5, a 3D schematic of the problem is presented, illustrating the UAV velocity vector v , its components $[v_x, v_y, v_z]$, and the UAV horizontal velocity v_h through blue arrows. Meanwhile, the dynamic obstacle velocity vector v_{obs} , its components $[v_{obs,x}, v_{obs,y}, v_{obs,z}]$, and its horizontal velocity $v_{obs,h}$ are represented with red arrows. Using this schematic diagram in Figure 5, the 3D relative distance rate ($\dot{R}_{3D}$; for convenience R is written as R_{3D}) is defined by the following equation:

$$\dot{R}_{3D} = v_{obs} \cos(\psi_{obs} - \sigma_h) \cos(\gamma_{obs} - \sigma_v) - v \cos(\psi - \sigma_h) \cos(\gamma - \sigma_v) \quad (18)$$

2. Line-Of-Sight Rate ($\dot{\sigma}$) Computation

This section focuses on deriving the Line-of-Sight (LOS) rate for both 2D and 3D scenarios. Initially, the computation of the 2D LOS rate, denoted as $\dot{\sigma}$, is addressed. By utilizing the schematic diagram presented in Figure 4, the derivation of the 2D LOS rate between the UAV and the obstacle is as follows:

$$\dot{\sigma} = \frac{v_{obs} \sin(\psi_{obs} - \sigma) - v \sin(\psi - \sigma)}{R_{2D}} \quad (19)$$

Where, R_{2D} is the 2D relative distance between the UAV and the obstacle. σ is the 2D LOS angle between the UAV and the obstacle. These parameters are defined as follows:

$$R_{2D} = \sqrt{(x_{obs} - x_x)^2 + (y_{obs} - y_y)^2} \quad (20)$$

$$\sigma = \arctan\left(\frac{y_{obs} - y_y}{x_{obs} - x_x}\right) \quad (21)$$

In the context of the 3D scenario, the consideration of the vertical plane becomes essential for computation. Consequently, two distinct Line-of-Sight (LOS) rates emerge—one for the horizontal plane ($\dot{\sigma}_h$) and another for the vertical plane ($\dot{\sigma}_v$). By employing the schematic diagram in Figure 5, the derivation of both LOS rates is as follows:

$$\dot{\sigma}_h = \frac{v_{obs} \cos(\gamma_{obs} - \sigma_v) \sin(\psi_{obs} - \sigma_h) - v \cos(\gamma - \sigma_v) \sin(\psi - \sigma_h)}{R_{3D}} \quad (22)$$

$$\dot{\sigma}_v = \frac{v_{obs} \cos(\psi_{obs} - \sigma_h) \sin(\gamma_{obs} - \sigma_v) - v \cos(\psi - \sigma_h) \sin(\gamma - \sigma_v)}{R_{3D}} \quad (23)$$

In the given context, R_{3D} represents the 3D relative distance between the UAV and the obstacle in three-dimensional space. Additionally, σ_h and σ_v denote the 3D horizontal and vertical Line-of-Sight (LOS) angles, respectively. These parameters are defined as follows:

$$R_{3D} = \sqrt{(x_{obs} - x_x)^2 + (y_{obs} - x_y)^2 + (z_{obs} - x_z)^2} \quad (24)$$

$$\sigma_h = \arctan\left(\frac{y_{obs} - x_y}{x_{obs} - x_x}\right) \quad (25)$$

$$\sigma_v = \arctan\left(\frac{z_{obs} - x_z}{R_h}\right) = \arctan\left(\frac{z_{obs} - x_z}{\sqrt{(x_{obs} - x_x)^2 + (y_{obs} - x_y)^2}}\right) \quad (26)$$

Where R_h is the horizontal projection of the 3D relative distance R_{3D} . R_h is equal to the 2D relative distance R_{2D} .

C. GENEX and IPN Combination Algorithm

The algorithmic flowchart for the combination of GENEX and IPN is depicted in Figure 6, providing a clear overview of the entire algorithm structure. Subsequent subsections offer a comprehensive analysis of the 2D and 3D algorithms resulting from the combination of GENEX and IPN, utilizing the corresponding pseudocodes.

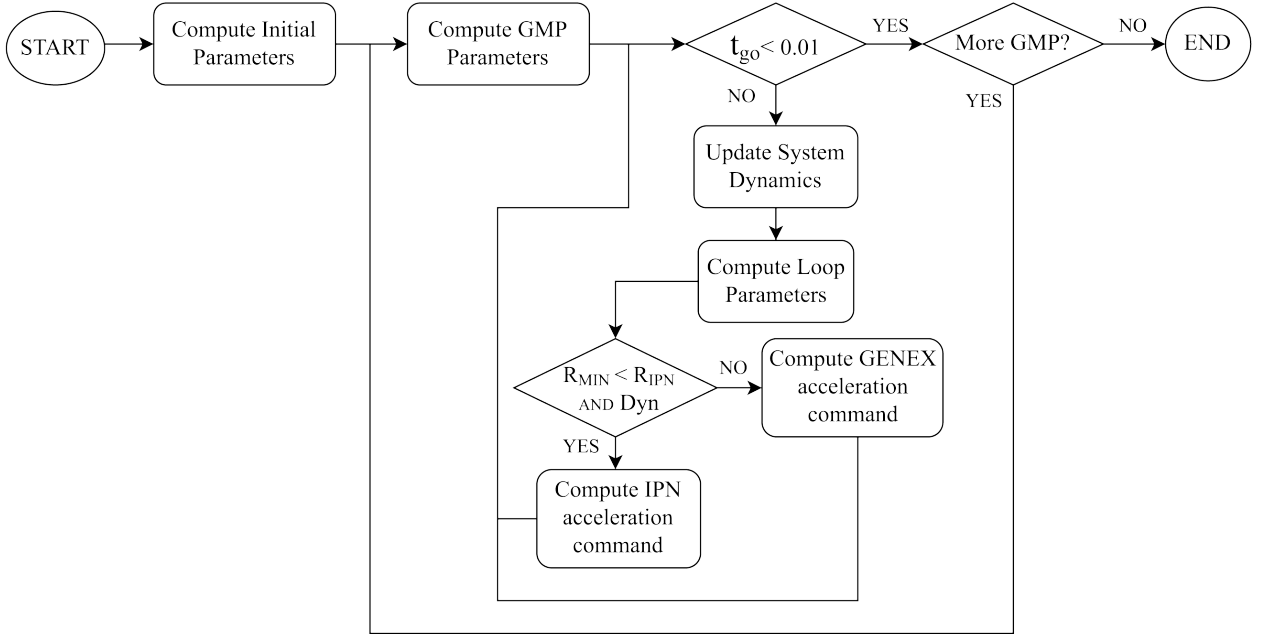


Fig. 6 GENEX and IPN Combination Algorithm Flowchart

1. Implementation in 2D

The GENEX and IPN combination algorithm (2D main pseudocode) is presented in Algorithm 1.

Algorithm 1 GENEX and IPN Combination 2D 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_GENEX_Parameters
7:     for  $n \leftarrow 1, n_{obs,dyn}$  do
8:       Calculate_IPN_Parameters
9:     end for
10:     $[R_{min}, j] \leftarrow \min(R)$ 
11:    if  $R_{min} < R_{IPN}$  AND  $j > n_{obs,stc}$  then ▷ Check if  $j$  belongs to a Dyn.Obs.
12:      Calculate_IPN_Acceleration_Command
13:    else
14:      Calculate_GENEX_Acceleration_Command
15:    end if
16:  end while
17: end for
```

The first thing it does is initialising the parameters. The function *Initialisation* initialises global parameters, constraints, obstacles, system initial conditions, GMPs, and desired heading at GMPs etc. Following this, the main algorithm enters the outer loop (denoted by the “for” loop), wherein it sequentially iterates through each GMP, encompassing the DDP designated as the final GMP (‘End’). For each GMP, the algorithm computes its initial parameters within the function *Calculate_GMP_Initial_Parameters*. This function determines the initial relative parameters between the UAV and the GMP, including the relative position R_{GMP} and its components $R_{x,GMP}$ and $R_{y,GMP}$, the heading angle error ψ_{he} , the closing velocity V_c , and the time-to-go t_{go} . Ultimately, the initial acceleration command u is calculated. Next, the main algorithm gets inside the inner loop (*whileloop*), where it stays until the value of t_{go} gets smaller than 0.01, meaning that the UAV reaches the GMP. In this inner loop, first, the main algorithm updates the system model. The dynamic obstacles do not need any angular rate to update their model because they are non-cooperative and move in a straight line through the environment.

Following the update of the system model, the main algorithm initiates the calculation of essential parameters needed for avoiding static obstacles and reaching the DDP using the *Calculate_GENEX_Parameters* function. Next, the algorithm determines the parameters necessary to navigate around incoming dynamic obstacles. This encompasses the computation of the relative distance R and relative distance rate $\dot{R}$ between the UAV and dynamic obstacles, the Line-Of-Sight angle σ , the Line-Of-Sight rate $\dot{\sigma}$ between the UAV and dynamic obstacles, as well as the heading angles ψ for both the UAV and dynamic obstacles. Subsequently, the algorithm checks whether the nearest obstacle is within the user-defined threshold R_{IPN} and whether this obstacle is dynamic (Line 11 in Algorithm 1). If both conditions are satisfied, the algorithm proceeds to compute the Inverse Proportional Navigation (IPN) acceleration command to avoid the approaching dynamic obstacle using the *Calculate_IPN_Acceleration_Command* function. Otherwise, it calculates the Geometric Nonlinear EXploration (GENEX) acceleration command to navigate toward the DDP while avoiding static obstacles, utilizing the *Calculate_GENEX_Acceleration_Command* function.

2. Implementation in 3D

The main algorithm for combining 3D GENEX and IPN (Algorithm 2) follows a structure similar to its 2D counterpart. The key difference lies in the 3D version, which introduces additional parameters due to the consideration of the vertical plane. Therefore, this section focuses on clarifying these parameters introduced by the inclusion of the third dimension, avoiding unnecessary repetition.

Algorithm 2 GENEX and IPN Combination 3D 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_GENEX_Parameters
7:     for  $n \leftarrow 1, n_{obs,dyn}$  do
8:       Calculate_IPN_Parameters
9:     end for
10:     $[R_{min}, j] \leftarrow \min(R)$ 
11:    if  $R_{min} < R_{IPN}$  AND  $j > n_{obs,stc}$  then ▷ Check if  $j$  belongs to a Dyn.Obs.
12:      Calculate_IPN_Acceleration_Command
13:    else
14:      Calculate_GENEX_Acceleration_Command
15:    end if
16:  end while
17: end for
```

V. Simulation Results and Discussions

We have presented simulation results in this section. The path planning simulation results are provided for 2D and 3D. They are given in the following subsection.

A. 2D results

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

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

$$\psi = \arctan\left(\frac{v_y}{v_x}\right) \quad (28)$$

$$\dot{\psi} = \frac{d\psi}{dt} = \frac{u}{v} \quad (29)$$

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. GENEX + IPN for Multiple Obstacles

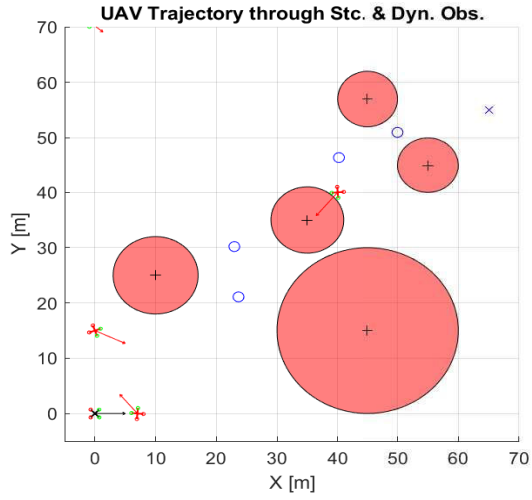
In this subsection, an analysis of the simulation results for the 2D combination algorithm of GENEX and IPN is conducted, considering scenarios with multiple static and dynamic obstacles. The initial conditions are presented in Tables 2 and 3. The dynamic obstacles are non-cooperative, moving in a straight line with constant heading and velocity. The states of the obstacles include x_{obs} , y_{obs} , r , v_{obs} , and ψ_{obs} . For this simulation study, five static obstacles ($n_{obs,stc} = 5$), seven dynamic obstacles ($n_{obs,dyn} = 7$), and $R_{IPN} = 5m$ have been considered.

Table 2 Initial Conditions: UAV and Static Obstacles

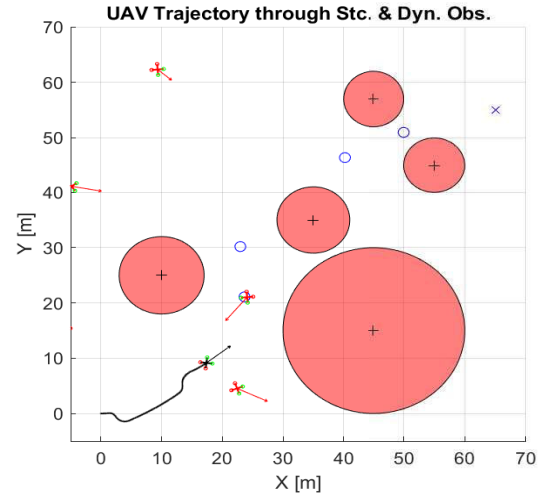
Unit	UAV		Static Obstacles					
m	x_x	0	x_{obs}	45	35	45	10	55
m	x_y	0	y_{obs}	57	35	15	25	45
m	-	-	r	5	6	15	7	5
m/s	v	10	v_{obs}	-	-	-	-	-
deg	ψ	0	ψ_{obs}	-	-	-	-	-

Table 3 Initial Conditions: Dynamic Obstacles and DDPs

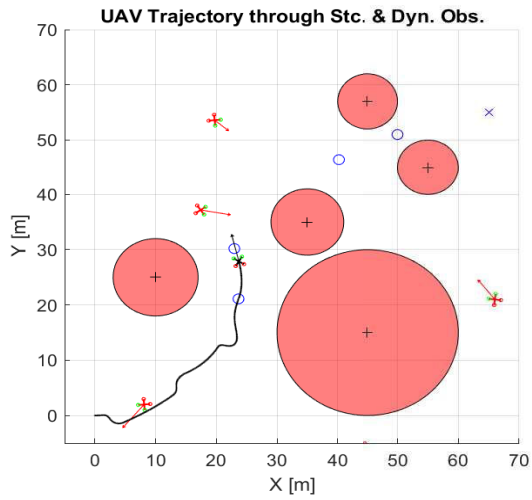
Unit	Dynamic Obstacles								DDP	
m	x_{obs}	7	0	40	-27	-1	91	-7	x_{DDP}	65
m	y_{obs}	0	15	40	45	71	-11	117	y_{DDP}	55
m	r	-	-	-	-	-	-	-	-	-
m/s	v_{obs}	9	11	11	10	6	9	9	-	-
deg	ψ_{obs}	130	-25	-130	-10	-40	128	-45	$\psi_{f,DDP}$	0



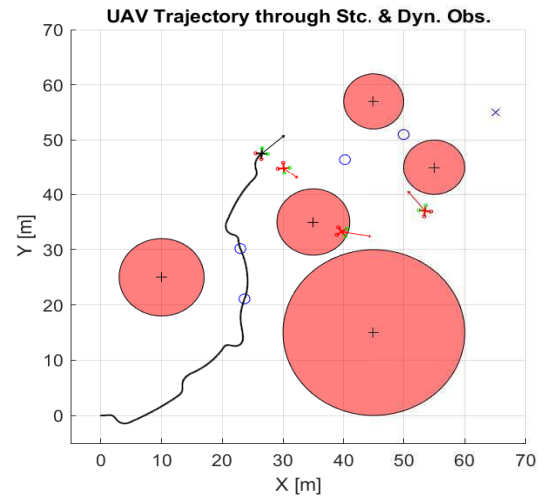
(a) $t = 0$ s



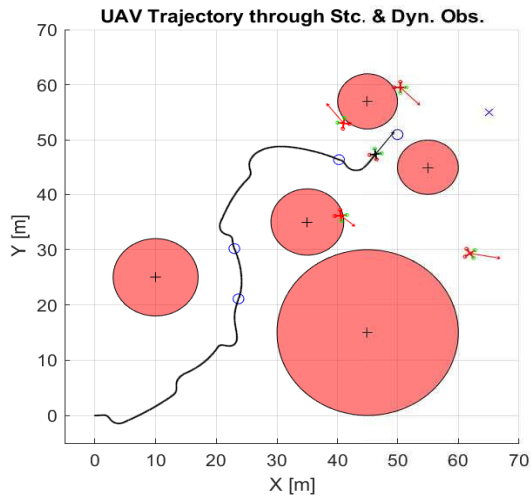
(b) $t = 2.26$ s



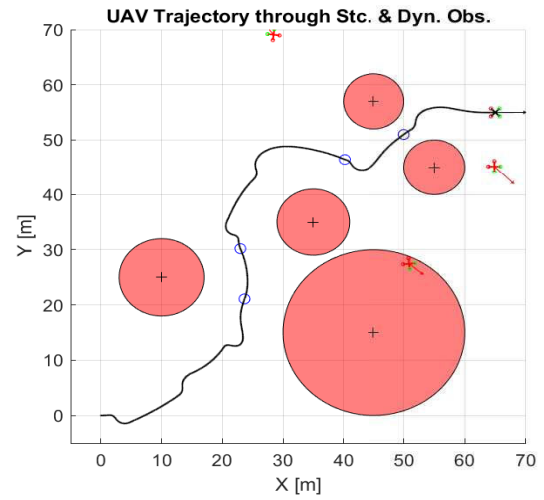
(c) $t = 4.52$ s



(d) $t = 6.78$ s



(e) $t = 9.04$ s



(f) $t = 11.29$ s

Fig. 7 GENEX + IPN Combination 2D - UAV Trajectory through Multiple Static and Dynamic Obstacles

In Figure 7, six frames capture the UAV's 2D trajectory as it navigates through five static obstacles (depicted as red circles) and avoids seven dynamic obstacles (represented by red UAV icons). As intended, the UAV successfully reaches the DDP marked by a blue cross, effectively circumventing both static and dynamic obstacles. To manoeuvre around the static obstacles, the UAV follows a path between them, guided by the GMPs denoted by small blue circumferences.

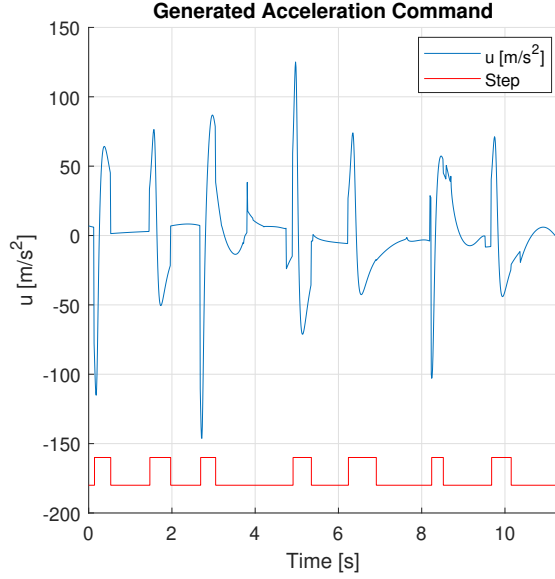


Fig. 8 Generated Acceleration Command: u [m/s²]
Step=-160 →IPN, Step=-180 →GENEX

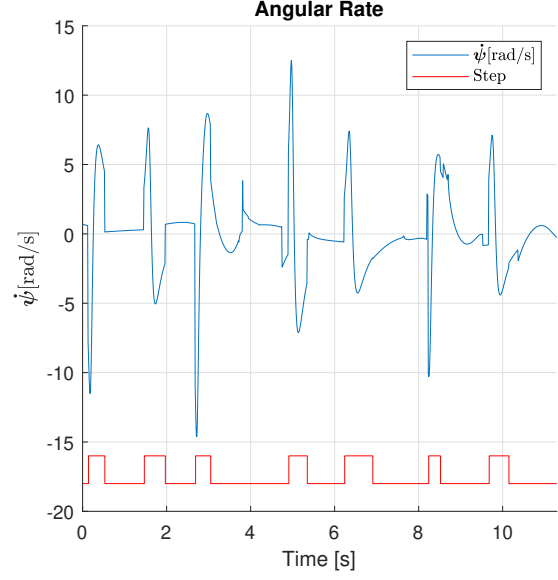


Fig. 9 Angular Rate: $\dot{\psi}$ [rad/s]
Step=-16 →IPN, Step=-18 →GENEX

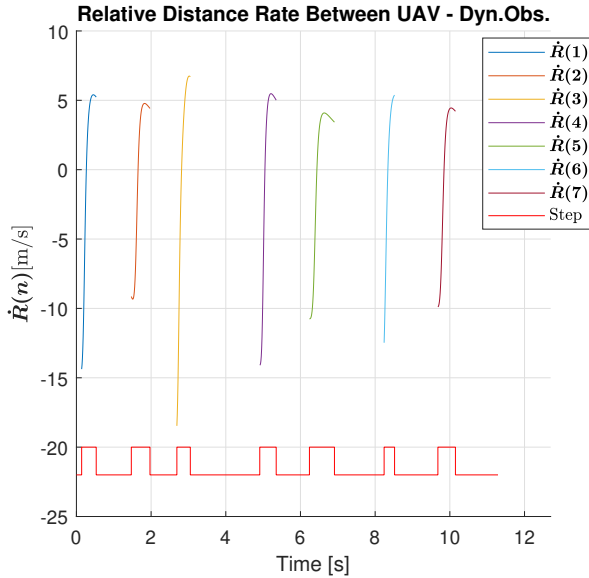


Fig. 10 Relative Distance Rate Between UAV and Dyn. Obs.: $\dot{R}(n)$ [m/s]
Step=-20 →IPN, Step=-22 →GENEX

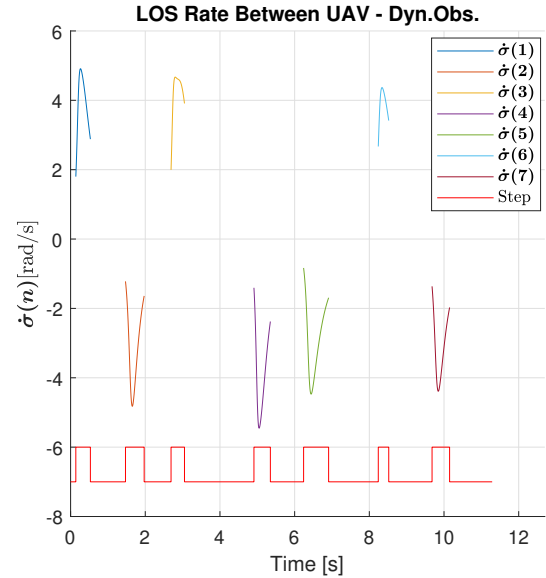


Fig. 11 Line-Of-Sight Rate Between UAV and Dyn. Obs.: $\dot{\sigma}(n)$ [rad/s]
Step=-6 →IPN, Step=-7 →GENEX

In Figure 8, the generated acceleration command u is displayed. When the Inverse Proportional Navigation (IPN)

command is activated, the acceleration command attains relatively high values compared to when the GENEX command is in effect. It reaches a maximum value of $-146.22m/s^2$ at $t = 2.72 s$. In Figure 9, the angular rate $\dot{\psi}$ is plotted. Figure 10 illustrates the relative distance rate between the UAV and multiple dynamic obstacles, denoted as $\dot{R}(n)$ (where $n = 1, 2, \dots, n_{obs,dyn}$). For a successful collision avoidance manoeuvre with each dynamic obstacle, $\dot{R}(n)$ must exhibit positive values. Figure 10 confirms this requirement, revealing a sudden increase in $\dot{R}(n)$ toward positive values for each dynamic obstacle when the IPN command is activated. Conversely, in Figure 11, the Line-Of-Sight rate between the UAV and multiple dynamic obstacles, denoted as $\dot{\sigma}(n)$, is presented. To ensure obstacle avoidance, $\dot{\sigma}(n)$ should not remain equal to zero. As intended, Figure 11 indicates that when the IPN command is activated, $\dot{\sigma}(n)$ assumes non-zero values for each dynamic obstacle.

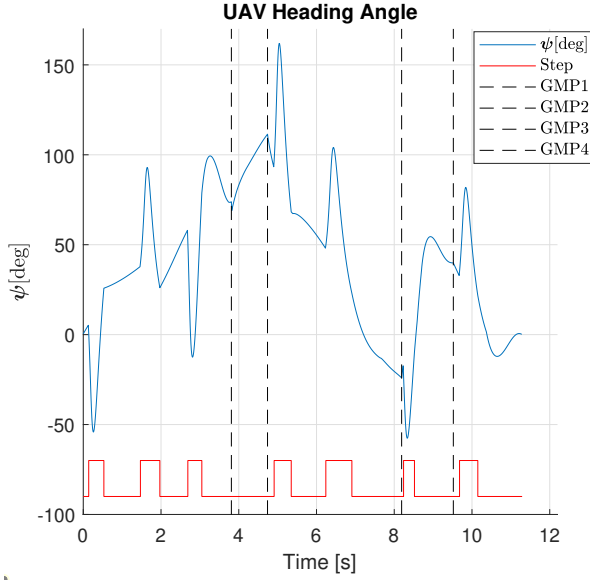


Fig. 12 UAV Heading Angle: ψ [deg]
Step=-60 → IPN, Step=-80 → GENEX

Desired heading	deg
$\psi_{f,GMP1}$	74.0546
$\psi_{f,GMP2}$	111.8014
$\psi_{f,GMP3}$	-24.4440
$\psi_{f,GMP4}$	39.8056
$\psi_{f,DDP}$	0

Table 4 Computed UAV Final Heading Angle:
 ψ_f (deg)

In Figure 12, the UAV heading angle extracted from the simulation is presented. This graph serves as a means to verify whether the UAV reaches the GMPs and the DDP with the intended final angles: $\psi_{f,GMP(n)}$ and $\psi_{f,DDP}$ as outlined in Table 4 (where $n = 1, 2, \dots, n_{obs}$). By comparing the extracted values (the black dotted vertical lines indicate the time at which the UAV reaches the GMPs and the corresponding angles achieved at those GMPs with the computed ones in Table 4, it becomes evident that the UAV attains the targets with the desired final heading angles. A noticeable change in the heading angle occurs when the Inverse Proportional Navigation (IPN) command is activated, reflecting the UAV altering its flight direction to evade incoming dynamic obstacles.

B. 3D scenario

1. UAV mathematical model

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

$$\mathbf{x} = \begin{bmatrix} x_x \\ x_y \\ x_z \end{bmatrix} \quad v = \frac{dx}{dt} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v \cos \gamma \cos \psi \\ v \cos \gamma \sin \psi \\ v \sin \gamma \end{bmatrix} \quad (30)$$

$$\psi = \arctan\left(\frac{v_y}{v_x}\right) \quad \gamma = \arctan\left(\frac{v_z}{v_h}\right) = \arctan\left(\frac{v_z}{\sqrt{v_x^2 + v_y^2}}\right) \quad (31)$$

$$\dot{\psi} = \frac{d\psi}{dt} = \frac{u_h}{v \cos \gamma} \quad , \quad \dot{\gamma} = \frac{d\gamma}{dt} = \frac{u_v}{v} \quad (32)$$

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.

2. GENEX + IPN 3D Multiple Obstacles

In this subsection, the analysis of the simulation results for the 3D GENEX and IPN combination algorithm considering multiple static and dynamic obstacles is performed. Tables 5 and 6 present the initial conditions and problem constants for this simulation.

Table 5 GENEX + IPN Combination 3D Multiple Obstacles - Initial Conditions

units	UAV		Static Obstacles								
[m]	x_x	0	x_{obs}	38	0	20	12	23	22	33	43
[m]	x_y	0	y_{obs}	38	20	9	28	24	23	33	43
[m]	x_z	0	z_{obs}	28	10	1	15	5	23	12	15
[m]	-	-	r	7	7	10	8	6	7	7	5
[m/s]	v	10	v_{obs}	-	-	-	-	-	-	-	-
[deg]	ψ	20	ψ_{obs}	-	-	-	-	-	-	-	-
[deg]	γ	20	γ_{obs}	-	-	-	-	-	-	-	-

units	Dynamic Obstacles							DDP	
[m]	x_{obs}	5	2	30	-28	171	35	x_{DDP}	47
[m]	y_{obs}	20	27	45	70	16.5	35	y_{DDP}	47
[m]	z_{obs}	3	20	24	14	32	60	z_{DDP}	28
[m]	r	-	-	-	-	-	-	-	-
[m/s]	v_{obs}	15	8	7.5	12.5	30	4.8	-	-
[deg]	ψ_{obs}	-80	-45	-115	-32	170	45	$\psi_{f,DDP}$	45
[deg]	γ_{obs}	0	-40	-30	5	-5	-66	$\gamma_{f,DDP}$	0

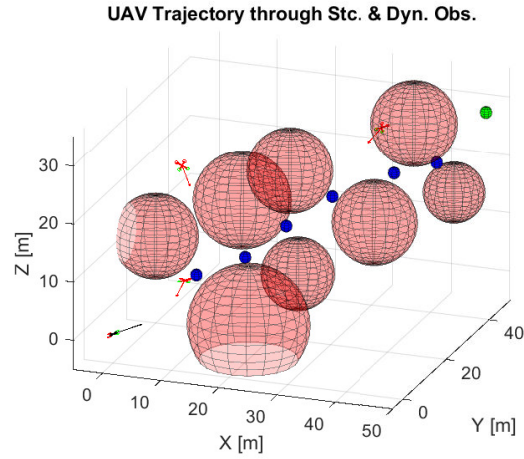
Table 6 GENEX + IPN Combination 3D Multiple Obstacles - Problem Constants

R_{IPN}	5 [m]
$n_{obs,stc}$	8
$n_{obs,dyn}$	6

In Figure 13, six frames capture the UAV's 3D trajectory as it navigates through eight static obstacles (depicted as red spheres) and avoids six dynamic obstacles (represented by red UAV icons). As intended, the UAV successfully reaches the DDP marked by a green sphere, effectively circumventing both static and dynamic obstacles. To navigate around static obstacles, the UAV follows a path between them, guided by the GMPs represented by blue spheres. Figures 14 and 16 depict the generated horizontal and vertical acceleration commands, denoted as u_h and u_v , respectively. Both acceleration commands reach relatively high values when the Inverse Proportional Navigation (IPN) command is activated compared to the GENEX command. The maximum values reached for u_h and u_v are -218.11 and -173.67 m/s^2 , respectively.

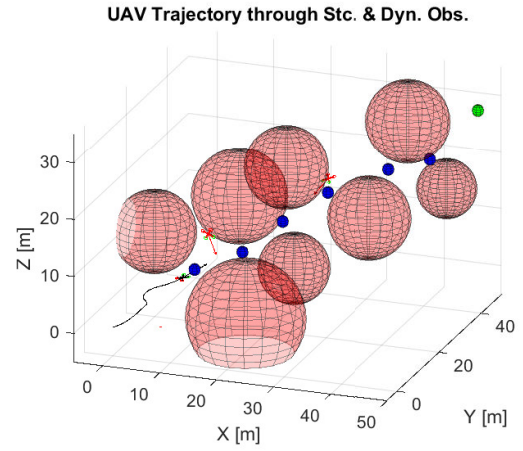
Figures 15 and 17 illustrate the horizontal and vertical angular rates $\dot{\psi}$ and $\dot{\gamma}$, respectively. Figure 17 is essentially a scaled-down version of Figure 16. While Figure 15 may appear similar to Figure 14, they are not identical since the

horizontal angular rate is influenced by the value of $\cos(\gamma)$. Figure 18 displays the relative distance rate between the UAV and multiple dynamic obstacles, denoted as $\dot{R}(n)$ (where $n = 1, 2, \dots, n_{obs,dyn}$). This graph is instrumental in verifying that $\dot{R}(n)$ assumes positive values, ensuring a successful collision avoidance manoeuvre. As intended, Figure 18 reveals a sudden increase in $\dot{R}(n)$ toward positive values for each dynamic obstacle once the Inverse Proportional Navigation (IPN) command is activated. The IPN command is activated twice to avoid the second dynamic obstacle, influenced by the distribution of the GMPs in this simulation.



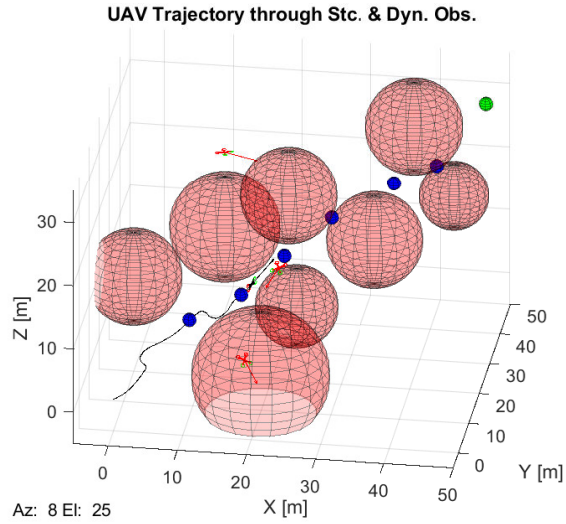
Az: 20 El: 20

(a) $t = 0 \text{ s}$



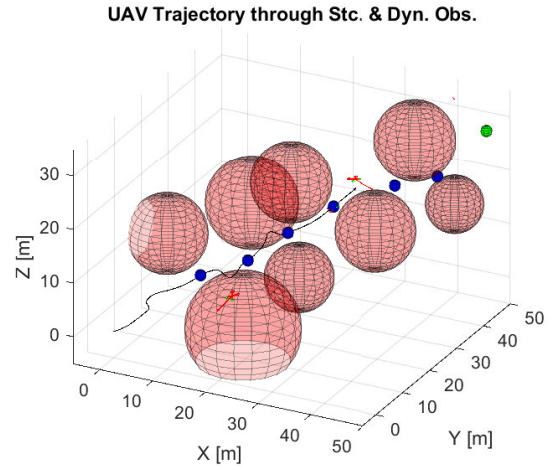
Az: 20 El: 20

(b) $t = 1.77 \text{ s}$



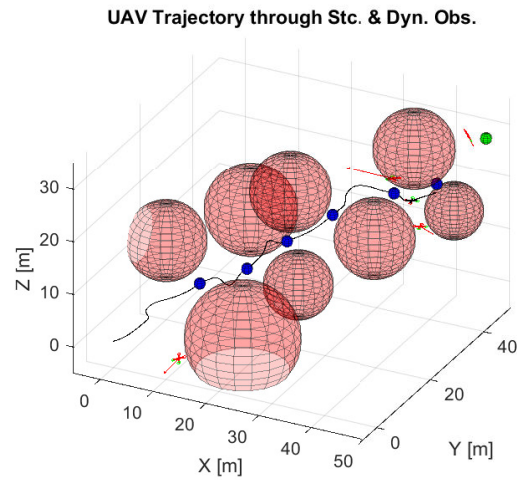
Az: 8 El: 25

(c) $t = 3.53 \text{ s}$



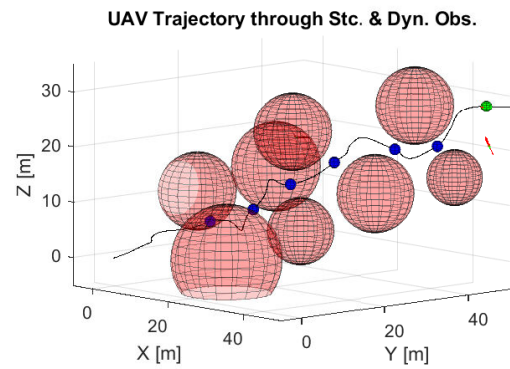
Az: 25 El: 25

(d) $t = 5.30 \text{ s}$



Az: 25 El: 25

(e) $t = 7.06 \text{ s}$



Az: 48 El: 8

(f) $t = 8.82 \text{ s}$

Fig. 13 GENEX + IPN Combination 3D - UAV Trajectory through Multiple Static and Dynamic Obstacles

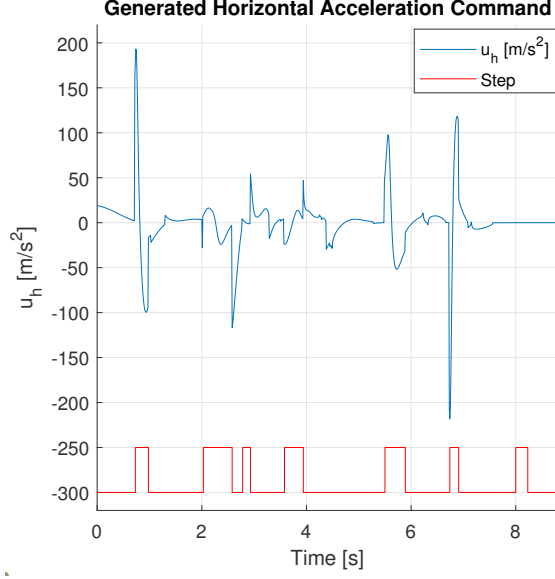


Fig. 14 Generated Horizontal Acceleration Command: u_h [m/s²]
Step=-250 →IPN, Step=-300 →GENEX

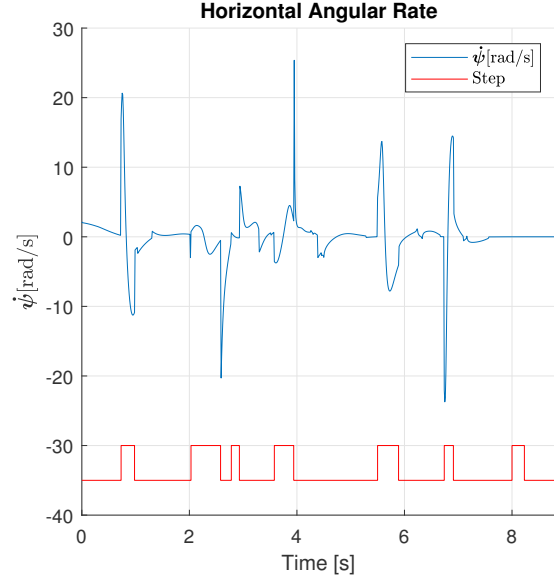


Fig. 15 Horizontal Angular Rate: $\dot{\psi}$ [rad/s]
Step=-30 →IPN, Step=-35 →GENEX

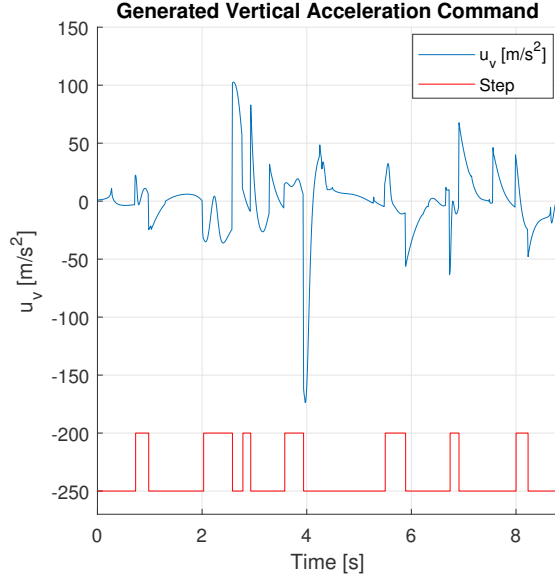


Fig. 16 Generated Vertical Acceleration Command: u_v [m/s²]
Step=-200 →IPN, Step=-250 →GENEX

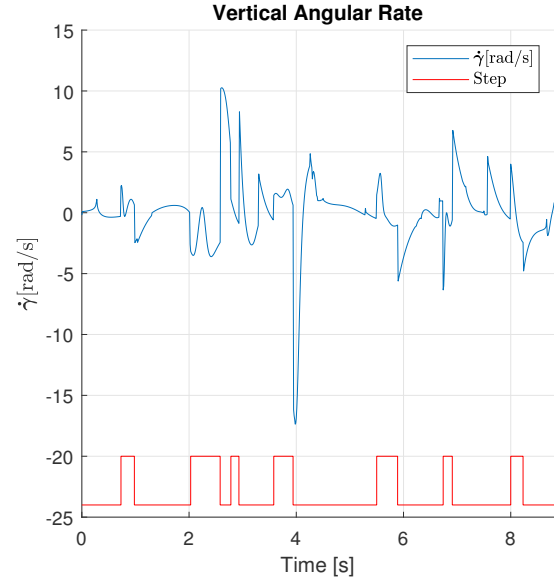


Fig. 17 Vertical Angular Rate: $\dot{\gamma}$ [rad/s]
Step=-20 →IPN, Step=-25 →GENEX

Figures 19 and 20 illustrate the horizontal and vertical Line-Of-Sight rates between the UAV and multiple dynamic obstacles, denoted as $\dot{\sigma}_h(n)$ and $\dot{\sigma}_v(n)$, respectively. For successful obstacle avoidance, $\dot{\sigma}_h(n)$ and $\dot{\sigma}_v(n)$ should not persistently remain equal to zero. As intended, Figures 19 and 20 demonstrate that when the Inverse Proportional Navigation (IPN) command is activated, $\dot{\sigma}_h(n)$ and $\dot{\sigma}_v(n)$ do not concurrently remain equal to zero. Notably, while $\dot{\sigma}_h(6)$ remains constant at 0 rad/s, $\dot{\sigma}_v(6)$ assumes non-zero values. This observation is attributed to the fact that to evade the sixth obstacle, the UAV executes a collision avoidance manoeuvre in the vertical plane.

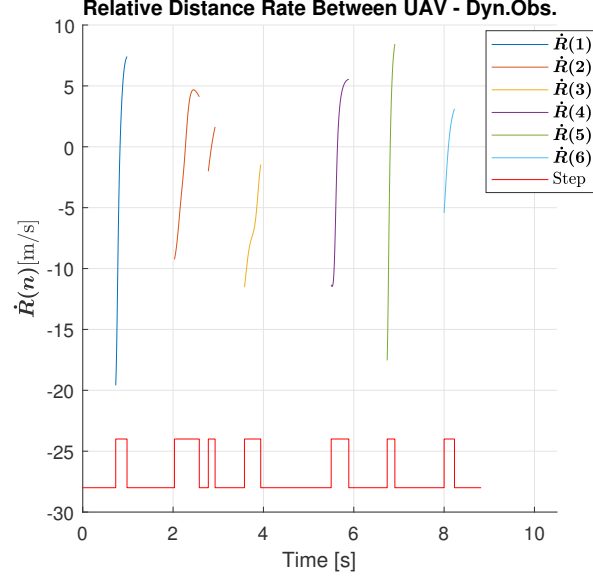


Fig. 18 Relative Distance Rate Between UAV and Dyn.
Obs.: $\dot{R}(n)$ [m/s]
Step=-24 $\rightarrow$ IPN, Step=-28 $\rightarrow$ GENEX

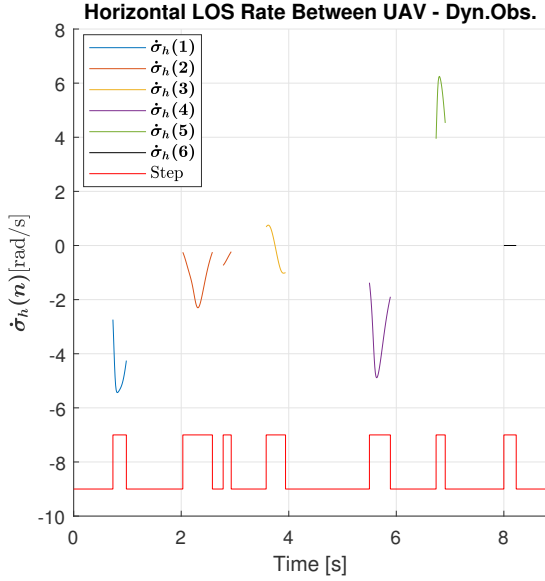


Fig. 19 Horizontal LOS Rate Between UAV and
Dynamic Obs.: $\dot{\sigma}_h(n)$ [rad/s]
Step=-7 $\rightarrow$ IPN, Step=-9 $\rightarrow$ GENEX

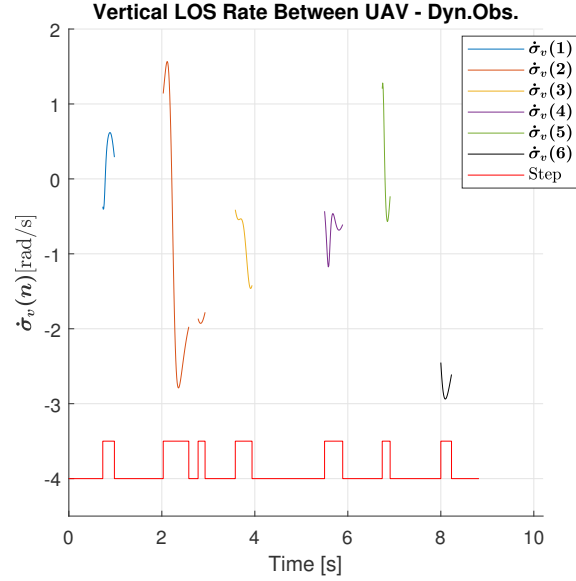


Fig. 20 Vertical LOS Rate Between UAV and
Dynamic Obs.: $\dot{\sigma}_v(n)$ [rad/s]
Step=-3.5 $\rightarrow$ IPN, Step=-4 $\rightarrow$ GENEX

Finally, in Figures 21 and 22, the extracted UAV heading and pitch angles from the simulation are displayed. These graphs serve as a means to verify whether the UAV reaches multiple GMPs and the DDP with the intended final angles: $\psi_{f,GMP(n)}$, $\psi_{f,DDP}$, $\gamma_{f,GMP(n)}$, and $\gamma_{f,DDP}$. The extraction procedure remains consistent with previous simulations. A comparison of the extracted values with the computed ones in Tables 7 and 8 reveals that the UAV successfully reached the targets with the desired final heading and pitch angles.

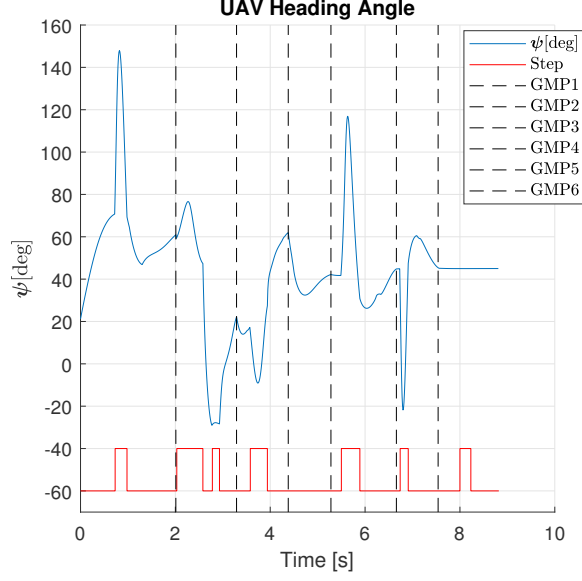


Fig. 21 UAV Heading Angle: ψ [deg]
Step=-40 →IPN, Step=-60 →GENEX

Angle	[deg]
$\psi_{f,GMP1}$	61.1892
$\psi_{f,GMP2}$	22.8337
$\psi_{f,GMP3}$	62.1027
$\psi_{f,GMP4}$	42.2739
$\psi_{f,GMP5}$	45
$\psi_{f,GMP6}$	45
$\psi_{f,DDP}$	45

Table 7 Computed UAV Final Heading Angle:
 ψ_f [deg]

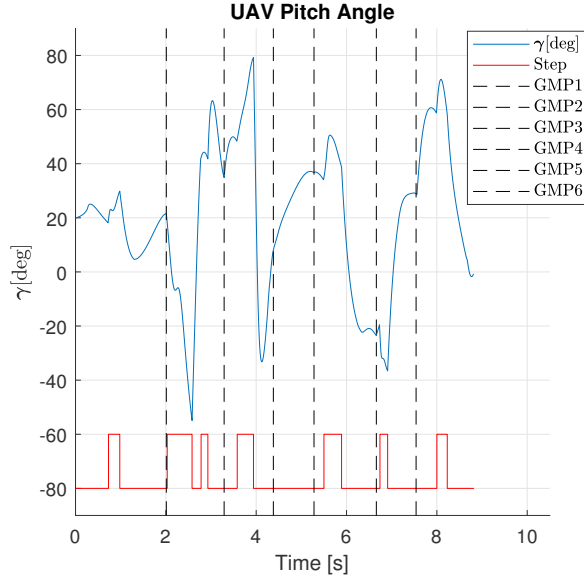


Fig. 22 UAV Pitch Angle: γ [deg]
Step=-60 →IPN, Step=-80 →GENEX

Angle	[deg]
$\gamma_{f,GMP1}$	21.5203
$\gamma_{f,GMP2}$	34.1827
$\gamma_{f,GMP3}$	8.8637
$\gamma_{f,GMP4}$	36.4974
$\gamma_{f,GMP5}$	-23.8408
$\gamma_{f,GMP6}$	28.5448
$\gamma_{f,DDP}$	0

Table 8 Computed UAV Final Pitch Angle: γ_f [deg]

VI. Conclusion

The suggested approach for path planning and obstacle avoidance, employing the combined GENEX and IPN, yields favourable outcomes. The algorithms automate control expressions in a closed form, enabling real-time execution onboard. GENEX effectively adheres to constraints at the GMPs while determining the optimal trajectory. Meanwhile, IPN initiates a reactive collision avoidance manoeuvre, with GENEX assuming control post-collision avoidance. Our testing encompassed 2D and 3D scenarios featuring various static and dynamic obstacles, reflecting real-world conditions. Consequently, the proposed strategy emerges as a promising solution for practical and instantaneous applications.

References

- [1] 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>.
- [2] 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>.
- [3] 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>.
- [4] 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>.
- [5] 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>.
- [6] 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>.
- [7] 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>.
- [8] 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>.
- [9] 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>.
- [10] 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>.
- [11] 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.
- [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] Fish, D., Olin, F. W., Holt, J., and Biaz, S., “An Inverse Proportional Navigation Algorithm for Collision Avoidance among Multiple Unmanned Aerial Vehicles,” *Computer Science*, 2012.
- [15] Heller, C., Yaesh, I., and Ben-Asher, J. Z., “Optimal Setting of the Proportional-Navigation gain,” *Journal of Guidance, Control, and Dynamics*, Vol. 36, 2013, pp. 888–891. <https://doi.org/10.2514/1.59226>.
- [16] Moreira, M., “Guidance, Navigation and Control for Interception of Non-Cooperative UAVs,” 2019. URL www.isr.tecnico.ulisboa.pt.
- [17] Han, S. C., Bang, H., and Yoo, C. S., “Proportional navigation-based collision avoidance for UAVs,” *International Journal of Control, Automation and Systems*, Vol. 7, 2009, pp. 553–565. <https://doi.org/10.1007/s12555-009-0407-1>.
- [18] Han, S. C., and Bang, H., “Proportional navigation-based optimal collision avoidance for UASs,” 2004, pp. 76–81. URL <https://www.researchgate.net/publication/228958279>.
- [19] George, J., and Ghose, D., “A reactive inverse PN algorithm for collision avoidance among multiple unmanned aerial vehicles,” 2009, pp. 3890–3895. <https://doi.org/10.1109/ACC.2009.5160241>.
- [20] D’Amato, E., Notaro, I., and Mattei, M., “Distributed Collision Avoidance for Unmanned Aerial Vehicles Integration in the Civil Airspace,” Institute of Electrical and Electronics Engineers Inc., 2018, pp. 94–102. <https://doi.org/10.1109/ICUAS.2018.8453432>.

- [21] 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, 2019, pp. 4926–4942. <https://doi.org/10.1177/0954410019834780>.
- [22] George, J., and Ghose, D., “A reactive inverse PN algorithm for collision avoidance among multiple unmanned aerial vehicles,” *2009 American Control Conference*, IEEE, 2009, pp. 3890–3895.
- [23] Fish, D., Westman, E., Holt, J., and Biaz, S., “An Inverse Proportional Navigation Algorithm for Collision Avoidance among Multiple Unmanned Aerial Vehicles CSSE12-06,” ????



DSSPACE

<https://dspace.org/>

Real-time path planning considering static and dynamic obstacles

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

2024-01-04

American Institute of Aeronautics and Astronautics (AIAA)

<https://doi.org/10.2514/6.2024-0875>

Subies Hueso J, Mondal S, Tsourdos A. (2024) Real-time path planning considering static and dynamic obstacles. In: AIAA SCITECH 2024 Forum, 8-12 January 2024, Orlando, USA. Paper number AIAA 2024-0875