



**HAL**  
open science

# Toward Dynamic Path Planning of Unmanned Aerial Vehicles (UAVs) in Windy Conditions 1

Ali Haidar Ahmad, Oussama Zahwe, Abbass Nasser, Benoit Clement

► **To cite this version:**

Ali Haidar Ahmad, Oussama Zahwe, Abbass Nasser, Benoit Clement. Toward Dynamic Path Planning of Unmanned Aerial Vehicles (UAVs) in Windy Conditions 1. International Conference on Control, Mechatronics and Automation, IEEE, Nov 2024, London, United Kingdom. hal-04778257

**HAL Id: hal-04778257**

**<https://hal.science/hal-04778257v1>**

Submitted on 12 Nov 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Toward Dynamic Path Planning of Unmanned Aerial Vehicles (UAVs) in Windy Conditions

1<sup>st</sup> Ali Haidar Ahmad  
Crossing IRL CNRS 2010  
ENSTA Bretagne  
Brest, France

ali.haidar@ensta-bretagne.org

2<sup>nd</sup> Oussama Zahwe  
Vectrawave Device SAS  
MMIC Dept.  
Lannion, France

oussama.zahwe@vectrawave.com

3<sup>rd</sup> Abbass Nasser  
ICCS-Lab, Computer Science Department  
Holy-Spirit University of Kaslik (USEK), Kaslik, Lebanon  
Beirut, Lebanon

abbass.nasser@usek.edu.lb

4<sup>th</sup> Benoit Clement  
Crossing IRL CNRS 2010  
ENSTA Bretagne  
Adelaide, Australia  
benoit.clement@ensta-bretagne.fr

**Abstract**—Unmanned Aerial Vehicle (UAV) path planning optimizes trajectory for efficient target reach with minimal energy consumption. This study enhances the Target Interception method (TI), to address wind-related challenges. We introduce two adaptations: TI Dynamic Path and TI External Influences. Implementing these methods in a dynamic wind environment yields results demonstrating enhanced reliability and efficiency. This research marks a significant stride in developing more effective UAV path planning algorithms.

**Index Terms**—UAV, Obstacle Avoidance, Path planning algorithms, Optimization.

## I. INTRODUCTION

Nowadays, the use of Unmanned Aerial Vehicles (UAVs) is no longer limited to military purposes [1] or environmental monitoring [2]. UAVs are now utilized in various domains, such as delivering products, as seen in Amazon’s service called Prime Air [3], or delivering food [4].

The main goal of the researchers is to find the most suitable path to ensure safe navigation of the drones, reaching their destinations using the shortest and most time-efficient flight paths. UAV trajectory planning has seen a proliferation of combined methodologies aimed at optimizing trajectory planning. The integration of various techniques has led to substantial advances in terms of efficiency, accuracy, and adaptability. In particular, the integration of multiple methods has shown promise in addressing the complexities of real-world scenarios. In this context, our research lies at the intersection of this evolving landscape. Building on our previous work [5], where path planning is divided into two categories:

Global path planning (GPP) and Local path planning (LPP). GPP aims to identify the optimal route for a UAV to travel from its starting point to its destination within a given environment, while avoiding obstacles and complying with any constraints, such as restricted airspace or minimum altitude requirements. Four subgroups of GPP: optimization methods, geometric methods, grid methods, and differential games [5].

We are set to further enhance UAV trajectory planning by introducing two new approaches. The first approach involves periodic reassessment of the shortest path. If the calculated path differs from the current trajectory, the drone will adjust its route accordingly. The second approach incorporates real-time wind dynamics into our methodology. By considering the effects of wind, we aim to enhance the accuracy and robustness of trajectory planning, taking a critical environmental factor into account. This new approach is categorized as a hybrid method, representing a new subsection within GPP, as illustrated in Fig 1. This paper is organized as follows: In Section 2, we provide an overview of related work, discussing existing literature and studies related to UAV path planning in windy conditions and also present our new proposal. Section 3 contains the implementations that have been done and the results, which highlight the advantages and limitations of each method. Section 4 concludes the paper.

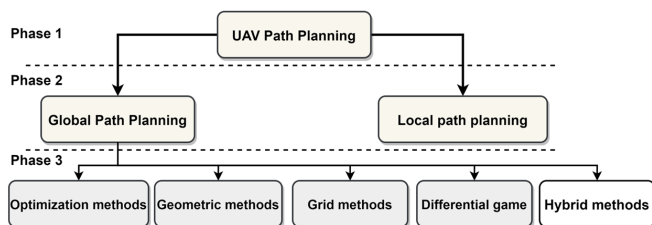


Fig. 1. The UAV Path Planning Classification Categories

## II. RELATED WORK

Many researchers have been actively exploring the impact of wind on drone movement, suggesting new approaches. In [6], the author introduces a cost-effective approach that allows drones to follow designated routes, maintaining stable camera focus on targets without the need for costly wind sensors. This innovation enhances UAV navigation, addressing wind effects

with improved efficiency and affordability. Furthermore, researchers are refining established methods by incorporating the wind factor, emphasizing the increasing acknowledgment of wind's influence on drone navigation. These endeavors reflect the continuous quest for more reliable flight strategies. In [7], the author introduces wind-aware trajectory planning, integrating wind formations and quadcopter kinematics. They formulate A\* cost functions considering distance and wind data. A\* has a major limitation in generating paths in dynamic environments, such as those containing wind [8]. In addition, the study proposes a collision verification method applied to outdoor terrain scenarios with different wind types. In a recent study [9], the author proposes an efficient particle swarm optimization (PSO) metaheuristic algorithm to reduce energy consumption, which is known as a non-computationally intensive algorithm [10], assuming that the UAV operates within a two-dimensional plane and flies at a fixed altitude. They also utilize the Voronoi Diagram [9], to simplify the urban environment, accounting for no-fly zones. A path cost function is formulated by considering total energy consumption. This approach [9] aims for a near-optimal solution while significantly reducing computational time, offering a practical and efficient method for UAV path planning in complex urban settings. A new technique for outdoor quadcopter trajectory planning was proposed [7]. This work is an enhanced version of the well-known A\* algorithm cost functions, combining distance and wind data for optimized navigation. A collision avoidance technique helps to avoid obstacles, and an intelligent approach to graph creation takes into account the complexity of winds. The comparative analysis favors probabilistic route map (PRM) graphs, demonstrating their effectiveness. Probabilistic Route Map (PRM) is an algorithm used in robotics to find and improve the route from the robot's starting point to its intended destination [11]. The study recommends the hybrid A\* angle for conflicting wind scenarios and emphasizes the advantages of on-board processing in UAVs, specifically probabilistic roadmapping. This research establishes foundations for wind-optimized global planning, enhancing quadcopter navigation amid real-time wind changes and limited communication resources.

in order to calculate how the wind will affect an aircraft, the authors in [12] assume that the aircraft flies at a constant flight level and at a constant true airspeed, and under this assumption they use the following equation of motion (1):

$$\begin{aligned}\dot{x}(t) &= V_a \cos(\theta(t)) + W_x(x, y) \\ \dot{y}(t) &= V_a \sin(\theta(t)) + W_y(x, y)\end{aligned}\quad (1)$$

with  $(x, y)$  the aircraft position,  $\theta$  the heading angle,  $V_a$  the True Airspeed,  $W_x(x, y)$  the east component of the wind and,  $W_y(x, y)$  the north component of the wind.

#### A. Target Interception (TI)

Target interception is to collide with the target or what can be considered as a collide in case of using thresholds below which any distance can be considered as a collision. The target must be in motion or in displacement, then the drone must

calculate the future location of the target precisely, taking into account its speed and heading angle, or predict it based on other features. For example, using the linear regression to predict the future positions based on past position. The second step is to generate a feasible and safe trajectory from current position of the mobile robot to the target point [13]. In our previous research [5], We tackled the challenge of docking a drone onto a mobile station in a 2D dynamic environment with wind effects and without obstacles. We consider this use case as a cooperative scenario, meaning that neither the landing station nor the drone would take any premeditated actions to prevent the goal from being reached. To meet this challenge, we applied two Differential games methods: Homicidal chauffeur (HC) and target interception (TI). To detect any external influences on the drone, we compare the actual location with the assumed location using constant speed and heading to ensure the drone is on the right trajectory. If external influences are detected, the drone will generate a new path based on the new coordinates and new heading angles of both drone and landing station. This process is described in detail in Algorithm 1.

Based on our previous results, we extend our work with proposing two new ideas considered as an enhanced version of the well-known differential game method, TI.

The first novel approach is TI Dynamic Path Adaptation (TI DPA). the drone recalculates the shortest path to intercept the moving landing station every 1 second using equations (1, 2). This method exempts the drone from calculating the wind effect at each step, unlike our previous proposal, which required such a calculation. However, it requires more complex calculations, as explained in Algorithm 2.

Second novel approach is TI External Influences Adaptation (TI EIA). The drone calculates the original path using the same method as the original TI method. The novelty of this approach lies in its ability to detect and measure external influences. If external influences are detected, the drone generates a new path, taking into consideration the strength of these influences Algorithm 3.

### III. IMPLEMENTATION AND RESULTS

In order to compare our new proposal with our previous research [5], we extend our study to dynamic environments with mobile destinations. We present two improved versions of the target interception methods: TI DPA and TI EIA. Both methods are specifically adapted to scenarios where a drone is tasked with intercepting a moving landing station. To establish a comprehensive evaluation, we apply these methods in the context of our research, allowing us to make meaningful comparisons with our previous work.

This implementation aims to mimic the implementation of our work [5], which addresses the challenge of docking a drone to a mobile station in a dynamic 2D environment with wind effects and no obstacles. The use case is considered to be a cooperative scenario, where neither the landing station nor the drone will undertake any premeditated actions to prevent the goal from being achieved. In our previous implementation,

we used a Python script to generate paths based on differential game algorithms (HC, TI). We also used JSBSIM FDM [14], to simulate the wind effects, with an altitude-hold autopilot to simplify the implementation to a 2D environment. In our current work, we choose to calculate the wind effect on the drone using (1) instead of JSBSIM, in order to improve calculation speed. The transition was essential to mitigate the substantial computing power required by the simulations, especially for TI DPA, where a lot of calculations had to be made every second to re-evaluate the shortest route. To calculate the UAV heading angle regarding its limited turn radius, we use (2) as defined in [15]. Where  $V$  denote the speed of UAV, and  $\theta$  denote its heading angle, counted clockwise from the Y axis. Also,  $R$  defines the minimum radius of curvature of UAV's trajectory.

$$\dot{\theta} = \frac{V}{R} \cdot \phi \quad |\phi| \leq 1 \quad (2)$$

Where  $\phi$  is the ratio of minimum radius of curvature to actual radius of curvature.

### A. Analyzing TI DPA Performance

In this paper three scenarios were implemented. In the first scenario, we compared the regular TI with TI DPA. We chose to exclude TI EIA from this comparison because it essentially mirrors the regular TI in windless conditions. This decision was crucial as it allowed us to concentrate our efforts on the unique qualities of TI and TI DPA. In addition to measuring their effectiveness in dynamic environments with moving destinations, our study also explored the execution time for each method. The execution time was determined using Python's `time.time()` function, which records the start and end times of the script, with the difference representing the total execution time. While this method provides a straightforward way to measure elapsed time, it can be influenced by factors such as system load, background processes, and available system resources, leading to potential variability between runs. To ensure precision, the script was executed multiple times, and the resulting execution times were found to be very consistent. Additionally, since the study compares two methods, both methods were timed under identical conditions on the same machine, with each method's execution time measured multiple times. This thorough approach not only gave us detailed insights into how these methods differ in performance but also provided valuable information about their computational efficiency. By taking into account both accuracy and execution time. We use a fixed starting location for the landing station during the implementations ( $X=250, Y=340$ ), which is 422 meters away from the UAV. The heading angles are varied as follows: 90, 60, 45, 30, 0, 120 and 100. The implementation results show that both methods produce nearly identical paths, especially in scenarios without wind effects. However, there is a noticeable variation in the total execution times across the 8 tests. These results are presented in Table I. The results are as expected since TI DPA recalculates the most suitable path every second, a feature unnecessary in scenarios without external influences affecting the drone.

### B. Landing station angle adjustment in wind

In the second implementation we use the same coordination of landing station ( $X=250, Y=340$ ) at  $t_0$ . Additionally, we maintained a steady wind speed of 5 m/s, which hit the drone at a 25-degree angle from  $t=10s$  onward. Meanwhile, we implemented with different landing station heading angles in order to test the efficiency and limitations of the three methods in windy environments. Fig. 2, 3, and 4 show the trajectories generated when the Landing Station (LS) orientation angle is set to  $90^\circ$ , with the TI EIA providing the most efficient trajectory when the UAV is affected by wind.

The results presented in Tables II and Table III show that TI EIA is the best method, in terms of both path quality and execution time in the presence of wind. However, it's important to note a significant issue: The TI EIA may not reach the landing point because it relies on initial wind conditions without adapting to subsequent changes. With this speed advantage comes the risk that the drone may miss its destination in the event of unexpected wind variations. Nevertheless, the other two methods, TI and TI DPA, had their own issues. TI failed to reach the landing station when the heading angle was set to 100 or 120 degrees. TI DPA failed at 100 degrees but successfully reached the landing station at 120 degrees. Despite this, TI DPA has made progress in terms of execution time, and the gap between the methods narrowed in this implementation, as TI had to readjust its path each time the drone faced significant wind changes after  $t = 10s$ .

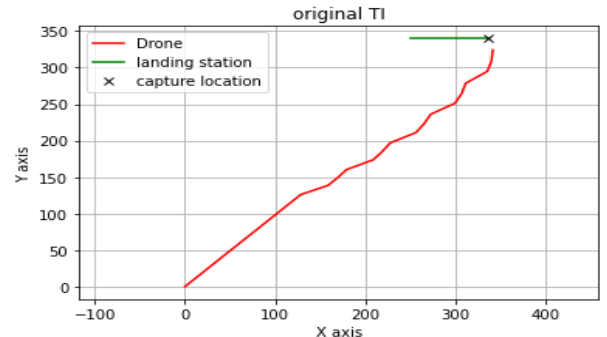


Fig. 2. TI Trajectory at LS Heading  $90^\circ$  and WS 5 m/s.

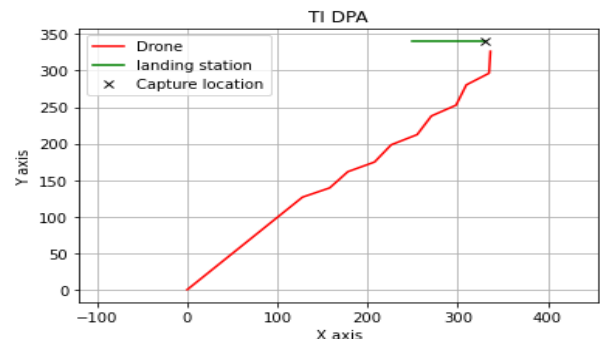


Fig. 3. TI DPA Trajectory at LS Heading  $90^\circ$  and WS 5 m/s.

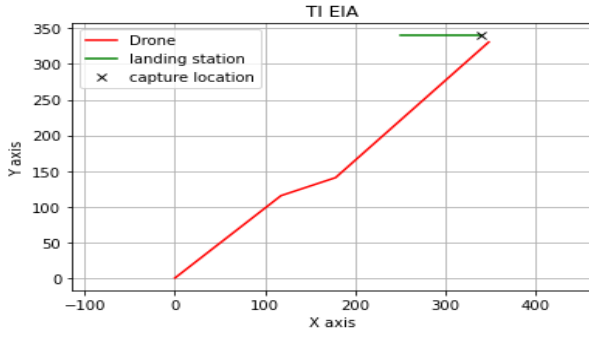


Fig. 4. TI EIA Trajectory at LS Heading 90° and WS 5 m/s.

**Input:** Initialize landing station start position ( $L_x$ ,  $L_y$ ), heading angle ( $L_h$ ) and speed ( $L_s$ ).  
**Input:** Initialize drone start position ( $D_x$ ,  $D_y$ ), drone speed ( $D_s$ ), drone heading angle ( $D_h=0$ ) and drone limited turn radius ( $L_{tr}$ ) (0.1).  
**Input:** Initialize an array  $\xi_i$  with values [Drone coordination, Drone heading angle, landing station coordination]  
 $\xi_i \leftarrow [D_x, D_y, D_h, L_x, L_y]$ ;

**Input:** Initialize an empty array  $arr\_L$   
**for**  $n = 1, \dots, 10000$  **do**  
 $x_n \leftarrow W_n \sin(\theta)$ ;  
 $y_n \leftarrow W_n \cos(\theta)$ ;  
Append ( $x_n, y_n$ ) to  $arr\_L$ ;  
**end**

**Input:** Function  $distance(x_0, y_0, x_1, y_1)$ : **return**  
 $\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$ ;

**Input:** Function  $angle(X)$ : **return** angle between Drone and landing station;

**Input:** Function  $landing(arr\_L)$   
 $i \leftarrow 0$ ;  
Initialize an empty array  $Result\_array$ ;  
**while**  $i < len(arr\_L)$  **do**  
 $L_x \leftarrow L_x +$   
 $L_s \cos(\text{angle}(D_x, D_y, arr\_L[0][i], arr\_L[1][i]))$ ;  
 $L_y \leftarrow L_y +$   
 $L_s \sin(\text{angle}(D_x, D_y, arr\_L[0][i], arr\_L[1][i]))$ ;  
**if**  $distance(D_x, D_y, arr\_L[x], arr\_L[y]) < D_s$  **then**  
Append  $i$  distance to  $Result\_array$ ;  
 $i \leftarrow i + 1$ ;  
**end**  
 $i \leftarrow i + 1$ ;  
**return**  $\min(Result\_array)$ ;  
**end**

**Input:** Function  $run(\xi_i, \Phi_i, \Psi_i)$   
 $\theta_{dot} \leftarrow \frac{speed}{L_s}$ ;  
 $\dot{x}_D \leftarrow x + D_s \times \sin(\Phi_i) + \theta_{dot}$ ;  
 $\dot{y}_D \leftarrow y + D_s \times \cos(\Phi_i) + \theta_{dot}$ ;  
 $\dot{x}_L \leftarrow x + L_s \times \sin(\Phi_i)$ ;  
 $\dot{y}_L \leftarrow y + L_s \times \cos(\Phi_i)$ ;  
**return**  $\dot{x}_D \ \dot{y}_D \ \dot{x}_L \ \dot{y}_L$ ;

**Function**  $start(\xi_i, angle, L_h)$   
 $nb\_seconds \leftarrow 0$ ;  
 $X \leftarrow [\xi_i]$ ;  
 $\phi_i \leftarrow \text{angle}(X)$ ;  
**while true do**  
 $next\_step \leftarrow run(X[-1], \phi_i, L_h)$ ;  
 $X.append(next\_step)$ ;  
**if**  $50 < nb\_seconds$  **then**  
 $D_x += wind\_effect\_x$ ;  
 $D_y += wind\_effect\_y$ ;  
 $nb\_seconds += 1$ ;  
**end**  
**if**  $distance(X[-1,0], X[-1,1], X[-1,3], X[-1,4]) < D_s$  **then**  
**Print**("Landing station has been reached after ",  $nb\_seconds$ , " seconds");  
**Break**;  
**end**  
**if**  $nb\_seconds \geq 10000$  **then**  
**Print**("Limit time has been reached");  
**Break**;  
**end**  
**end**  
**if**  $D_x \neq predicted\_Dx[i]$  **and**  $D_y \neq predicted\_Dy[i]$  **then**  
Calculate new  $\Phi_i$   
 $i \leftarrow i + 1$   
**end**  
**else**  
 $i \leftarrow i + 1$   
**end**  
**end**  
**return**  $X$ ;

**Input:**  $\xi_i$ : Initial state,  $angle$ : angle method,  $L_h$ : Heading angle  
**Output:**  $X$ : Trajectory path  
 $X \leftarrow runSim(\xi_i, angle, Eheading)$ ;

### Algorithm 1: Target Interception

TABLE I  
EXECUTION TIMES IN SECONDS: TI DPA VS. TRADITIONAL TI (WINDLESS SCENARIOS)

Table method	sum of execution time	average of execution time
TI	1.308	0.163
TI DPA	20.198	2.524

TABLE II  
UAV DISTANCE WITH CHANGING LANDING POSITION IN WIND

Distances traveled by the UAV in meters				
LS	Method	TI	TI DPA	TI EIA
	90	540.9	502.7	492
	60	547	543.5	536
	45	553.5	550.4	536.3
	30	641.3	552.5	Failed
	0	587.2	544.5	Failed
	120	Failed	Failed	460
	100	Failed	500.1	477.4

⋮

```

Function start (xi, angle, Lh)
  nb_seconds ← 0;
  X ← [xi];
  while true do
    phi ← angle (X);
    next_step ← run (X[-1], phi, Lh);
    X.append(next_step);
    if 50 < nb_seconds then
      Dx += wind_effect_x;
      Dy += wind_effect_y;
      nb_seconds += 1;
    end
    if distance (X[-1,0], X[-1,1], X[-1,3], X[-1,4]) <
      Ds then
      Print("Landing station has been reached after ",
        nb_seconds, " seconds");
      Break;
    end
    if nb_seconds ≥ 10000 then
      Print("Limit time has been reached");
      Break;
    end
  end
return X;

```

**Input:** *xi*: Initial state, *angle*: angle method, *Lh*: Heading angle

**Output:** *X*: Trajectory path  
 $X \leftarrow \text{runSim}(xi, angle, Eheading);$

**Algorithm 2:** Target Interception Dynamic Path Adaptation (TI DPA)

TABLE III

EXECUTION TIMES IN SECONDS FOR THE THREE METHODS DURING THE SECOND IMPLEMENTATION

		Execution time in seconds		
LS	Method	TI	TI DPA	TI EIA
	90	0.23	0.33	0.17
	60	0.24	0.35	0.17
	45	0.23	0.34	0.18
	30	0.26	0.33	0
	0	0.26	0.35	0
	120	0	0	0.2
	100	0	0.45	0.17

TABLE IV

UAV DISTANCE WITH VARIED WIND SPEED (353 METER TO LANDING STATION)

Method	Wind Speed variation		
	5m/s	7m/s	9 m/s
TI	437.17	449.27	17251.27
TI DPA	429.36	434.94	456.11
TI EIA	419.20	420.15	421.19

⋮

```

Input: Function run (xi, Phi, Psi)
theta_dot ←  $\frac{\text{speed}}{Ls}$ ;
if nb = False then
   $x'_D \leftarrow x + Ds \times \sin(\text{Phi}) + \text{theta\_dot}$ ;
   $y'_D \leftarrow y + Ds \times \cos(\text{Phi}) + \text{theta\_dot}$ ;
   $x'_L \leftarrow x + LS \times \sin(\text{Phi})$ ;
   $y'_L \leftarrow y + Ls \times \cos(\text{Phi})$ ;
end
else
   $x'_D \leftarrow x + Ds \times \sin(\text{Phi}) + \text{theta\_dot} -$ 
    ExternalInfluencesPower[0];
   $y'_D \leftarrow y + Ds \times \cos(\text{Phi}) + \text{theta\_dot} -$ 
    ExternalInfluencesPower[1];
   $x'_L \leftarrow x + LS \times \sin(\text{Phi})$ ;
   $y'_L \leftarrow y + Ls \times \cos(\text{Phi})$ ;
end
return  $x'_D$   $y'_D$   $x'_L$   $y'_L$ ;
Function start (xi, angle, Lh)
  nb_seconds ← 0;
  X ← [xi];
  phi ← angle (X);
  while true do
    next_step ← run (X[-1], phi, Lh);
    X.append(next_step);
    if 50 < nb_seconds then
      Dx += wind_effect_x;
      Dy += wind_effect_y;
      nb_seconds += 1;
    end
    if distance (X[-1,0], X[-1,1], X[-1,3], X[-1,4]) <
      Ds then
      Print("Landing station has been reached after ",
        nb_seconds, " seconds");
      Break;
    end
    if nb_seconds ≥ 10000 then
      Print("Limit time has been reached");
      Break;
    end
  end
if  $D_x \neq \text{predicted\_Dx}[i]$  and  $D_y \neq \text{predicted\_Dy}[i]$  then
  External_Influences_Power[0] ←
   $D_x - \text{predicted\_Dx}[i]$ 
  External_Influences_Power[1] ←
   $D_y - \text{predicted\_Dy}[i]$ 
  External_Influences ← True
   $i \leftarrow i + 1$ 
end
else
   $i \leftarrow i + 1$ 
end
return X;

```

**Input:** *xi*: Initial state, *angle*: angle method, *Lh*: Heading angle

**Output:** *X*: Trajectory path  
 $X \leftarrow \text{runSim}(xi, angle, Eheading);$

**Algorithm 3:** Target Interception External Influences Adaptation (TI EIA)

### C. Wind and Distance Impact on UAV Path Planning

In the third implementation, our primary objective was to assess the impact of wind and the initial distance between

TABLE V  
UAV DISTANCE WITH VARIED WIND SPEED (707 METER TO LANDING STATION)

Method	Wind Speed variation		
	5m/s	7m/s	9 m/s
TI	16903.20	16855.90	failed
TI DPA	871.55	910.28	18210.56
TI EIA	838.64	839.88	841.21

the drone and the landing station on all three methods. To conduct this evaluation, we standardized the landing station's heading angle at  $90^\circ$  ( $\frac{\pi}{2}$  radians) and set the wind direction to hit the drone at an angle of  $\frac{\pi}{2}$  radians (90 degrees) with respect to the horizontal axis. Throughout this implementation, we systematically varied the wind speeds, experimenting with 5 m/s, 7 m/s, and 9 m/s to observe their effects on the performance of the methods. Additionally, we adjusted the initial distance between the drone and the landing station at  $t = 0$ . One configuration placed the landing station at coordinates (250, 250), resulting in a distance of 353 meters between the two, while the other configuration set the landing station at (500, 500), creating an initial distance of 707 meters. The results presented in Tables IV and V clearly indicate the significant impact of wind speed and direction on UAV path planning across all six tests. Among the methods assessed, TI EIA emerged as the most effective, followed by TI DPA in second place, and the regular TI in third place, with one failing to reach the landing station.

#### D. Results

The results show that Regular TI generated the least efficient path among the three, although it still remains a viable option. TI EIA, on the other hand, gave the best results in terms of path length and execution time. However, it occasionally failed because it relied heavily on wind power and direction to generate a new path. This dependency meant that as wind conditions changed throughout the flight, the algorithm struggled to adapt effectively. TI DPA showed promise in reaching the landing station in all tests, but in some cases, the method failed to generate an efficient path. It required higher execution times, especially in windless conditions, due to unnecessary computations. The overall pattern suggests that each method has the potential to perform exceptionally well depending on the specific characteristics present in the environment.

#### IV. CONCLUSIONS AND FUTURE WORKS

In this paper, we explored the Hybrid subcategory within global path planning for UAVs, building upon our previous research where we introduced the classification categories and their respective subcategories for UAV global path planning. We have implemented two new scenarios, designed as improved hybrid versions of the classic TI method. These methods were developed to handle dynamic tasks, specifically docking a drone to a mobile landing station while considering wind as a significant external influencing factor. The results

show that the proposed method is more reliable than the classical TI method, which was proven to be the most reliable among the methods we tested in our previous work. However, in this study, our newly proposed method demonstrated even greater reliability compared to TI. This highlights a significant advancement in our approach and underscores the progress we've achieved.

As future work, more validation is needed to assess the reliability and robustness of the proposed scenario in real-world settings. Furthermore, we plan to integrate these methods into machine learning models for UAV trajectory planning.

#### REFERENCES

- [1] Giordano Bruno Antoniazzi Ronconi, Thaís Jessinski Batista, and Victor Merola. The utilization of unmanned aerial vehicles (uav) for military action in foreign airspace. *UFRGSMUN: UFRGS Model United Nations Journal*, 2:137–180, 2014.
- [2] Saeid Asadzadeh, Wilson José de Oliveira, and Carlos Roberto de Souza Filho. UAV-based remote sensing for the petroleum industry and environmental monitoring: State-of-the-art and perspectives. 208:109633, 2022.
- [3] Seyed Mahdi Shavarani, Mazyar Ghadiri Nejad, Farhood Rismanchian, and Gokhan Izbirak. Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of amazon prime air in the city of san francisco. *The International Journal of Advanced Manufacturing Technology*, 95(9):3141–3153, 2018.
- [4] Asish Oommen Mathew, Abhishek Nath Jha, Anasuya K. Lingappa, and Pranshu Sinha. Attitude towards drone food delivery services—role of innovativeness, perceived risk, and green image. *Journal of Open Innovation: Technology, Market, and Complexity*, 7(2):144, 2021.
- [5] Ali Haidar Ahmad, Oussama Zahwe, Abbass Nasser, and Benoit Clement. Path planning algorithms for unmanned aerial vehicle: Classification, performance, and implementation. In *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pages 1–6, 2023.
- [6] Herath M. P. C. Jayaweera and Samer Hanoun. Path planning of unmanned aerial vehicles (uavs) in windy environments. *Drones*, 6(5), 2022.
- [7] Georgios A. Thanellas, Vassilis C. Moulitanitis, and Nikos A. Aspragathos. A spatially wind aware quadcopter (UAV) path planning approach. *IFAC-PapersOnLine*, 52(8):283–288, 2019.
- [8] Howie. Choset, K.M. Lynch, S. Hutchinson, G.A. Kantor, and W. Burgard. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Intelligent Robotics and Autonomous Agents series. MIT Press, 2005.
- [9] Y.Y. Chan, Kam K.H. Ng, C.K.M. Lee, Li-Ta Hsu, and K.L. Keung. Wind dynamic and energy-efficiency path planning for unmanned aerial vehicles in the lower-level airspace and urban air mobility context. *Sustainable Energy Technologies and Assessments*, 57:103202, 2023.
- [10] M. Ercan and Xiang Li. Particle swarm optimization and its hybrids. *International Journal of Computer and Communication Engineering*, 2:52–55, 2013.
- [11] Saleh Alarabi, Chaomin Luo, and Michael Santora. A prm approach to path planning with obstacle avoidance of an autonomous robot. In *2022 8th International Conference on Automation, Robotics and Applications (ICARA)*, pages 76–80, 2022.
- [12] Brunilde Girardet, Laurent Lapasset, Daniel Delahaye, Christophe Rabut, and Yohann Brenier. Generating optimal aircraft trajectories with respect to weather conditions. In *ISIATM 2013, 2nd International Conference on Interdisciplinary Science for Innovative Air Traffic Management*, Toulouse, France, July 2013.
- [13] Chendi Qu, Jianping He, Jialun Li, Chongrong Fang, and Yilin Mo. Moving target interception considering dynamic environment. In *2022 American Control Conference (ACC)*, pages 1194–1199, 2022.
- [14] Tomáš Vogelntanz. A survey of free software for the design, analysis, modelling, and simulation of an unmanned aerial vehicle. *Archives of Computational Methods in Engineering*, 23(3):449–514, 2016.
- [15] Pierre T. Kabamba and Anouck R. Girard. *Fundamentals of Aerospace Navigation and Guidance*. Cambridge Aerospace Series. Cambridge University Press, 2014.