

Grafting of Gallic Acid onto a Bioactive Ti6Al4V Alloy: A Physico-Chemical Characterization

Original

Grafting of Gallic Acid onto a Bioactive Ti6Al4V Alloy: A Physico-Chemical Characterization / Cazzola, Martina; Ferraris, Sara; Prenesti, Enrico; Casalegno, Valentina; Spriano, Silvia. - In: COATINGS. - ISSN 2079-6412. - ELETTRONICO. - 9:5(2019). [10.3390/coatings9050302]

Availability:

This version is available at: 11583/2979563 since: 2023-06-26T09:01:29Z

Publisher:

MDPI

Published

DOI:10.3390/coatings9050302

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

A Risk-based Path Planning Strategy to Compute Optimum Risk Path for Unmanned Aircraft Systems over Populated Areas

Stefano Primatesta¹, Matteo Scanavino¹, Giorgio Guglieri¹, Alessandro Rizzo²

Abstract—The large diffusion of Unmanned Aircraft Systems (UAS) requires a suitable strategy to design safe flight missions. In this paper, we propose a novel path planning strategy to compute optimum risk path for UAS over populated areas.

The proposed strategy is based on a variant of the RRT* (Rapidly-exploring Random Tree "Star") algorithm, performing a risk assessment during the path planning phase. Like other RRT-based algorithms, the proposed path planning explores the state space by constructing a graph. Each time a new node is added to the graph, the algorithm estimates the risk level involved by the new node, evaluating the flight direction and velocity of the UAS placed in the analyzed node.

The risk level quantifies the risk of flying over a specific location and it is defined using a probabilistic risk assessment approach taking into account the drone parameters and environmental characteristics.

Then, the proposed algorithm computes an asymptotically optimal path by minimizing the overall risk and flight time.

Simulation results in realistic environments corroborate the proposed approach proving how the proposed risk-based path planning is able to compute an effective and safe path in urban areas.

I. INTRODUCTION

In the last years Unmanned Aircraft Systems (UAS) have been widely used in many applications both for commercial and personal use [1].

However, UAS are at the beginning of their expansion. In fact, the technological progress and the consequently increase of the autonomy level will enable the use of UAS in more and more applications [2]. Specifically, UAS will be widely used in urban areas [3]. In fact, thanks to their flexibility and low cost, UAS are the ideal platform to sense cities [4]. For this reason, a UAS is considered as a Flying IoT system and the term *Internet of Drone Things* (IoDT) has been coined in [5].

However, urban areas are complex environments and the integration of UAS in cities poses important challenges such as public safety, cybersecurity and privacy [6]. In particular, urban areas have a high population density and a possible crash of the UAS on the ground may involve people and, in the worst case, casualties. For this reason, the flight over populated areas is strongly restricted by National Aviation Authorities (NAA), such as ENAC (Ente Nazionale per l'Aviazione Civile) in Italy and FAA (Federal Aviation Administration) in the U.S.. Generally, the flying

over people is permitted only with particular conditions in order to guarantee an appropriate level of safety.

In Europe, EASA (European Union Aviation Safety Agency) is working on an European regulation for UAS [7], [8], in which the specific scenario with the flight over people is taken into account. EASA proposed the definition of three risk-based categories: open, specific and certified. For the *open* category only UAS with a mass lower than 0.250 kg and a speed lower than 19 m/s can fly over uninvolved people, but not over crowd. For the *specific* category, the flight over people is permitted only after a risk analysis using the SORA (Specific Operation Risk Assessment) guidelines proposed by JARUS (Joint Authorities for Rulemaking on Unmanned Systems) [9]. SORA is a qualitative approach and assesses the risk of a flight operation with a multi-step procedure, defining if an adequate level of safety is guaranteed and recommending a set of safety requirements for the UAS and the operation.

In the literature many alternative risk assessment approaches have been proposed [10]. One of the most popular methods is the probabilistic risk assessment used in [11], [12], [13], where the risk is defined as the probability to cause a fatality per flight hour. The probabilistic risk assessment is a quantitative and high-fidelity approach and can be used to quantify the risk of a specific flight mission [14]. Moreover, according to [15], the probabilistic risk assessment seems to be in agreement with SORA. The probabilistic risk assessment is also used in [16] to compute a ground risk-based map quantifying the risk of flying over an urban area.

Another interesting study is presented in [17], where the authors propose a risk assessment framework developed to provide real-time safety evaluation and tracking to be used in a UAS Traffic Management (UTM). Instead, a method to model and assess the risk of UAS is proposed in [18], taking into account both ground impacts and mid-air collisions.

In general, the risk assessment can be also used to provide safe flight missions. In [19], a risk-based path planning is presented using a bi-objective optimization method to compute a low-risk and time solution for UAS over populated areas. The same authors, in [20], propose an algorithm able to compute risk optimal UAS maneuvers estimating a probabilistic crash area. In [16], a two-stage risk-aware path planning is proposed to compute a safe path in dynamic risk-based maps by using a combination of two algorithms: riskA* and Borderland. In [21], a risk-based path planning is also used to seek for a safe landing area.

In our previous work [22], we have proposed a risk-aware path planning based on the RRT* (Rapidly-exploring

¹S. Primatesta, M. Scanavino and G. Guglieri are with the Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy. Corresponding author: Stefano Primatesta (e-mail: stefano.primatesta@polito.it)

²A. Rizzo is with the Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy.

Random Tree "Star") algorithm with the minimization of risk costs in a risk-based map. However, this path planning strategy evaluates risk costs of the risk-based map introduced in [23], where the risk is computed considering any flight direction and velocity. In fact, during the generation of the risk-based map, flight direction and velocity are not known and the estimated risk is defined considering a large impact area. As a consequence, the risk could be over-estimated or, in the worst case, under-estimated compared with the actual risk considering a specific flight direction and velocity. Thus, the path planning proposed in [22] does not use a realistic estimation of the risk.

In this paper, we propose an alternative path planning strategy able to compute safe routes for UAS over populated areas. The algorithm is based on the well-known RRT* algorithm [24] with the minimization of the overall risk and the flight time. The use of the flight time in the optimization is essential for two reasons: (i) the risk, expressed in casualties per flight hour (h^{-1}), is proportional with the flight time, and, (ii) often UAS have batteries with a limited time of autonomy, then, the flight time is a critical variable to be considered in the path planning. Unlike the risk-aware path planning method presented in [22], the proposed path planning approach does not minimize risk costs of a risk-based map, but it directly estimates the risk during the computation of the path. In fact, each time the path planning extends the graph with a new node, a risk assessment is performed by estimating the risk caused by the flight over the new node. This approach increases the complexity of the algorithm compared with [22], but it evaluates an efficient and realistic risk obtaining a more effective solution.

This paper is organized as follows. In Section II the ground risk assessment used by the path planning is described. In Section III the risk-based path planning is presented, detailing the proposed strategy and the pseudocode of the risk-based RRT* algorithm. Section IV reports numerical results. Our conclusions are drawn in Section V.

II. GROUND RISK ASSESSMENT

In this section, we explain the ground risk assessment used by the risk-based path planning algorithm.

The ground risk assessment used in this paper is based on the methodology presented in our previous work [23]. The risk is defined as the probability to cause a casualty when the UAS flies over a populated area and it is expressed in casualties per flight hour, a standard unit used in aviation. The risk is computed using a probabilistic risk assessment approach widely used in the literature [11], [12], [25], computed as a sequence of three conditional events: (i) the loss of control of the UAS with the consequent uncontrolled crash on the ground, (ii) the impact with at least one person, and, (iii) the impacted person suffers fatal injuries. Hence, the probability to have a casualty P_c is defined as

$$P_c(x) = P_{\text{event}} \cdot P_{\text{impact}}(x) \cdot P_{\text{fatality}}(x), \quad (1)$$

with x is the origin location in which the risk is computed. P_{event} is the probability that the UAS loses control with the

consequent uncontrolled descent and impact on the ground. The uncontrolled descent event may happen with different behaviors depending on the failure type and the UAS configuration (i.e. fixed-wing or multi-copter). In [23] four descent event types are considered: ballistic descent, uncontrolled glide, parachute descent and fly-away. The descent behavior is used to estimate the probabilistic impact area, the impact velocity, the impact angle and the kinetic energy at impact.

P_{impact} is the probability to impact with at least one person after the uncontrolled descent on the ground. It depends on the population density in the impact area and the area exposed to the crash.

P_{fatality} is the probability that the impact with a person causes a casualty. It is function of the kinetic energy at impact and of the sheltering factor in the impact area.

The risk is computed for each descent event using a common procedure:

- 1) The probabilistic impact area is computed using the mathematical model of the descent type. As a result, we obtain a geo-referenced two-dimensional Probability Density Function (2D-PDF). For simplicity the 2D-PDF is represented as a matrix in which each element represents a geo-referenced location with associated the probability to impact in the location, the estimated impact angle and the estimated impact velocity.
- 2) The 2D-PDF is modified using the wind speed and direction. Specifically, the 2D-PDF is computed in the wind frame and, then, the offset caused by the wind is determined by the drop time and the wind speed.
- 3) The probabilities P_{impact} and P_{fatality} are computed using the 2D-PDF and information about population density and sheltering factor in the impact area.
- 4) The probability P_c is computed.

Hence, since we assume the independence between failures and between descent events, the probabilities due to the descent events are simply added:

$$P_c(x) = P_c^{\text{bal}}(x) + P_c^{\text{ug}}(x) + P_c^{\text{par}}(x) + P_c^{\text{fa}}(x), \quad (2)$$

with P_c^{bal} , P_c^{ug} , P_c^{par} and P_c^{fa} are the probabilities of casualties for the ballistic descent, uncontrolled glide, parachute descent and fly-away, respectively.

A. Descent events

A reliable and realistic risk assessment should consider the behavior of the UAS during the uncontrolled descent on the ground. In fact, the type of descent determines how the UAS crashes on the ground, defining the impact area and the kinetic energy at impact. In [23] four different descent event types are used:

- Ballistic descent: happens when the UAS loses most of its lift and the aircraft is subject to a ballistic descent.
- Uncontrolled glide: occurs when the UAS enters in an uncontrolled descent governed by the glide ratio or autorotation descent angle, depending on the aircraft configuration.
- Parachute descent: happens when the UAS descends with a fully deployed parachute.

- Fly-away: happens when the ground station loses the operator control authority, while the on-board autopilot maintains the aircraft stable.

For each descent event type a mathematical model is defined and used to estimate the probabilistic impact area. Specifically, the 2D-PDF is computed taking into account drone specifications and initial conditions (e.g. flight altitude and initial velocity), as well as considering probabilistic assumptions on parameters. To have more details about the descent event type and their mathematical model, please refer to [23], [25], [26].

Figure 1 illustrates an example of probabilistic impact area. Specifically, this example refers to the DJI Mavic Pro, a small and lightweight UAS and, according to results of [23], is suitable to fly over populated areas. Due to the low mass, only the ballistic descent event involves the required amount of kinetic energy to cause casualties. In particular, Figure 1(a) shows the 2D-PDF taking into account a specific flight velocity and direction. On the other hand, Figure 1(b) illustrates the 2D-PDF considering all combinations of initial velocities and flight directions.

B. Probability of Impacting a Person

The probability P_{impact} is the probability of impact at least one person after the uncontrolled impact on the ground of the UAS. In this work we use a method widely used in the literature [11], [12], in which the probability is defined as

$$P_{\text{impact}}(x) = \rho(x) \cdot A_{\text{exp}}, \quad (3)$$

where $\rho(x)$ is the population density at the location x and A_{exp} is the area exposed to the crash.

The area exposed to the crash, also called *lethal area*, is defined using a method proposed in our previous work [22], taking into account the dimension of an average person and of the UAS

$$A_{\text{exp}}(\theta) = \pi(r_p + r_{\text{uav}})^2 \sin(\theta) + (r_p + r_{\text{uav}})(h_p + r_{\text{uav}}) \cos(\theta), \quad (4)$$

where θ is the impact angle, r_{uav} is the radius of the aircraft, r_p and h_p are the average radius and height of a person.

As detailed in [23], the risk assessment computes the probability P_{impact} evaluating the probabilistic impact area and using the estimated impact angle, described by the 2D-PDF. Hence, the 2D-PDF is used to compute the expected values of population density and lethal area

$$P_{\text{impact}}(x) = \sum_{x \in \text{PDF}} \text{PDF} \cdot \rho(x) \cdot A_{\text{exp}}(\theta(x)), \quad (5)$$

with ρ and $A_{\text{exp}}(\theta)$ are independent variables. The sum expression considers all the geo-referenced locations of the probabilistic impact area and their probabilities defined as a probability density function PDF.

C. Probability of Fatality

The probability P_{fatality} is the probability that the impacted person suffers fatal injuries. This probability is computed using the method proposed in [12], taking into account

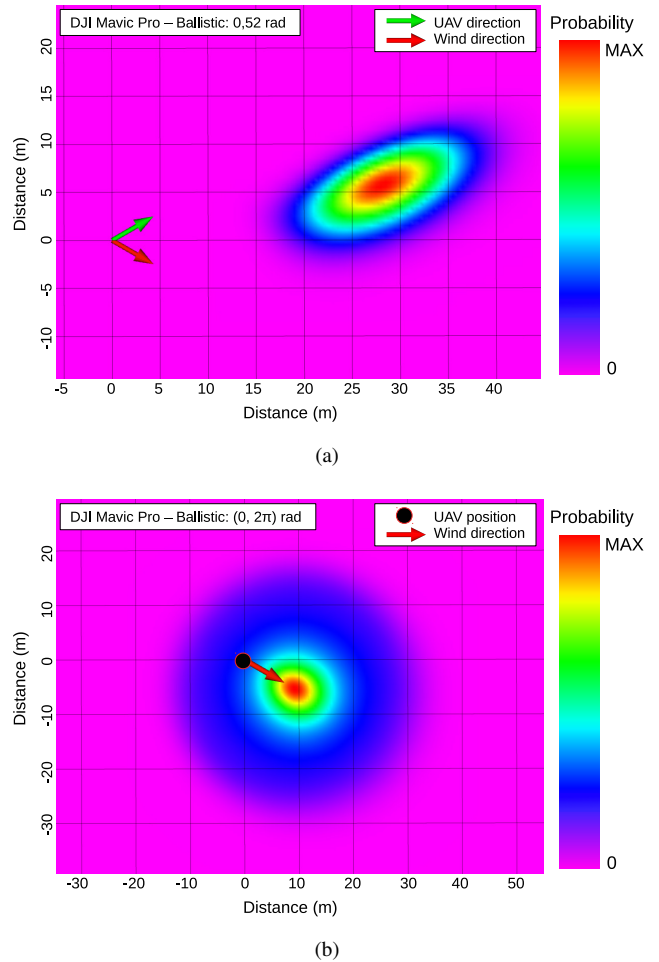


Figure 1. A 2D-PDF of the ballistic descent with the DJI Mavic Pro aircraft. In red, areas with the highest probability of fall, whereas in magenta areas with probability equal to zero. The UAS flies at an altitude of 30 m, the wind has a direction of -0.52 rad and speed 4 m/s. In (a) the aircraft has a heading angle of 0.52 rad and a cruise velocity of 10 m/s, while in (b) all directions with uniform distribution $U(0, 2\pi)$ rad and all velocities with uniform distribution $U(0, 10)$ m/s are considered.

the kinetic energy at impact and the sheltering factor. The sheltering factor is a parameters that determines how people on the ground are sheltered by buildings and other obstacles from a possible impact with a UAS. It is essential to evaluate it, because the presence of buildings and obstacles in the impact area reduces the kinetic energy at impact. In [12] the sheltering factor is an absolute real number in the range from 0 to infinite. However, according to [27], it is useless to consider too large values for the sheltering factor because, after a certain value, a huge kinetic energy is required to cause a fatality. Then, as defined in [27], we use a sheltering factor in the range from 0 to 10, where 0 refers to an area with no shelter and 10 to an area with the maximum level of sheltering factor, such as an industrial building. Similarly with the probability of impact a person, the probability P_{fatality} is computed considering the 2D-PDF

and the estimated kinetic energy at impact. Hence,

$$P_{\text{fatality}}(x) = \frac{1 - k}{1 - 2k + \sqrt{\frac{\alpha}{\beta} \left[\frac{\beta}{E[E_{\text{imp}}(x)]} \right]^{\frac{3}{E[s(x)]}}}}, \quad (6)$$

with $k = \min[1, (\frac{\beta}{E[E_{\text{imp}}(x)]})^{\frac{3}{E[s(x)]}}]$. Where $s(x)$ is the sheltering factor at the location x , $E_{\text{imp}}(x)$ is the estimated kinetic energy at impact in the location x , the α parameter is the impact energy for a fatality probability of 50% when the sheltering factor is equal to 6, and the β parameter is the impact energy to cause a fatality when the sheltering factor goes to zero. According to [28], the fatality limit is defined with $\beta = 34J$. In Equation 6, $E[\cdot]$ refers to the expected value

$$E[E_{\text{imp}}(x)] = \sum_{x \in \text{PDF}} \text{PDF} \cdot E_{\text{imp}}(x), \quad (7)$$

$$E[s(x)] = \sum_{x \in \text{PDF}} \text{PDF} \cdot s(x), \quad (8)$$

where the sum expression takes into account all the locations x of the probabilistic impact area and their probabilities defined as a probability density function (PDF).

III. RISK-BASED PATH PLANNING

In this section we describe the risk-based path planning strategy proposed in this work. As already explained in the previous paragraphs, the proposed risk-aware path planning approach does not use a risk-based map to estimate the risk, as in [22], but it directly computes the risk during the path planning phase using the risk assessment method described in the previous section. However, the risk assessment requires some essential and detailed information about the population density and the sheltering factor in the area equivalent to the state space of the path planning. For this reason, a multi-layer framework is used, in which each layer contains useful data of homogeneous nature. Technically, a layer is a geo-referenced matrix in which each element is associated with a geo-referenced location and has a specific value dependent on the type of layer. In this work, the multi-layer framework is composed by the following layers:

- **Population density layer:** defines the population density distribution in the area;
- **Sheltering factor layer:** defines the sheltering factor of each element of the layer;
- **Obstacles layer:** defines the height of buildings and other obstacles in the area;
- **No-fly zone layer:** defines in which areas the flight is not allowed.

While the population density and sheltering factor layers are used to estimate the risk level, the obstacles and the no-fly zones layers are used by the path planning to avoid areas where the flight is not allowed. The multi-layer framework is illustrated in Figure 2.

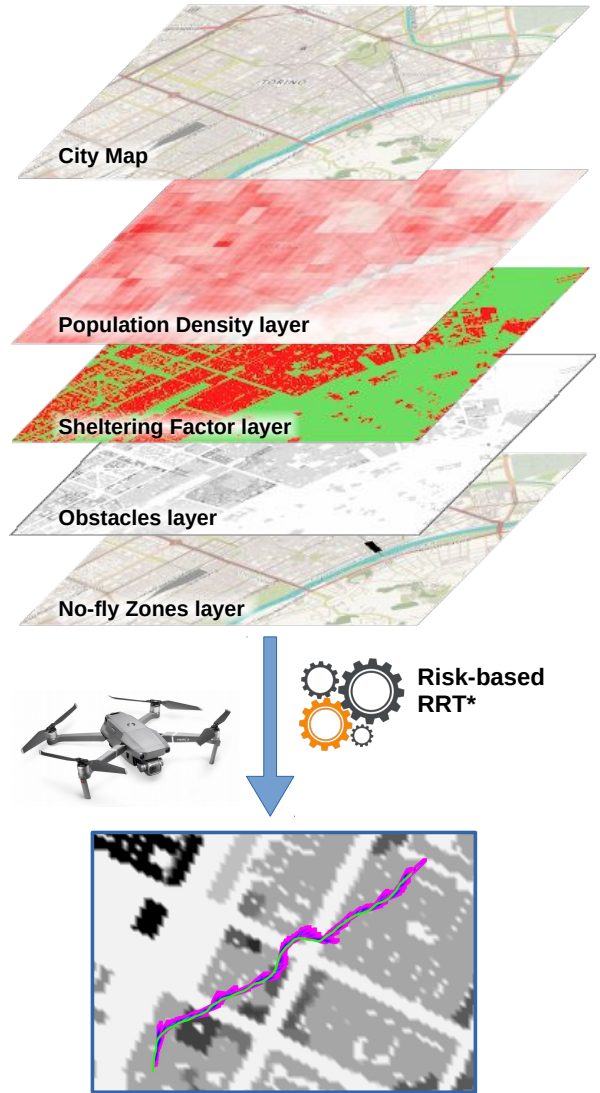


Figure 2. The multi-layer framework used by the proposed risk-based path planning. Layers are used by the path planning to estimate the risk level and to avoid obstacles at the flight altitude and no-fly zones.

A. Notation and Problem statement

Let \mathcal{X} denote the d-dimensional state space, where \mathcal{X} is a measurable metric space that has finite measure $\mathcal{L}(\mathcal{X}) = k$, where $\mathcal{L}(\cdot)$ is the Lebesgue measure of the space with $k < \infty$. Each state of the state space is denoted as $x \in \mathcal{X}$. With a slight abuse of notation, here and henceforth we will refer to x as a state of state space, as the geo-referenced location corresponding to the state and as a node of the graph generated by the path planning algorithm. The notation $d(x_1, x_2)$ is the Euclidean distance function on \mathcal{X} with $d(x_1, x_2) \geq 0$.

Let \mathcal{X}_{obs} denote the obstacle space that corresponds to obstacles or no-fly zones. The free space $\mathcal{X}_{\text{free}} = \mathcal{X} \setminus \mathcal{X}_{\text{obs}}$ is the remaining navigable space.

The start and goal states are x_{start} and x_{goal} respectively, with $x_{\text{start}}, x_{\text{goal}} \in \mathcal{X}_{\text{free}}$. The path connecting the start and goal states is denoted as $\sigma(x_{\text{start}}, x_{\text{goal}})$. The optimal path

planning searches for an optimal path $\sigma^*(x_{\text{start}}, x_{\text{goal}})$ as a sequence of states from x_{start} to $x_{\text{goal}} \in \mathcal{X}_{\text{free}}$, which minimizes a given cost function $\text{Cost}(\cdot) \geq 0$. Hence, the optimal path is the solution of the following program

$$\sigma^*(x_{\text{start}}, x_{\text{goal}}) = \arg \min_{\sigma(x_{\text{start}}, x_{\text{goal}}) \in \mathcal{X}_{\text{free}}} \text{Cost}(x_{\text{start}}, x_{\text{goal}}). \quad (9)$$

The cost function $\text{Cost}(\cdot)$ used in this work will be defined in the following section.

B. Risk-based RRT* algorithm

The proposed method is based on the well-know RRT* with the minimization of the risk. RRT* [24] is a sample-based algorithm able to compute near-optimal solutions even in high-dimensional spaces. RRT* searches for a solution by exploring the search space by building an incremental and asymptotically optimal graph. Practically, nodes are randomly sampled in the state space and used to extent incrementally the exploration graph. Unlike the RRT algorithm, RRT* ensures that nodes are reached through a minimum cost path. As a consequence, RRT* is asymptotically optimal and the cost of the returned solution converges to the optimum when the number of nodes tends to infinite. On the other hand, if a solution does not exist, the algorithm does not report it and continues to seek for a solution. However, in general, a termination condition is set to block the algorithm.

In our previous work [22], the RRT* algorithm minimized a motion cost taking into account the flight time and risk costs defined by a risk-based map that previously assesses the risk of flying over an urban area [25]. In this work, we propose a different approach, in which the risk-based path planning does not refer to a risk-based map, but assesses the risk during the exploration of the state space. Practically, the risk of flying over a populated area is computed at each step of the algorithm, i.e. during the graph extension and in the rewiring phase. In fact, each time a new node is sampled and added to the graph, the algorithm computes the risk involved by the new edge connecting the state to the graph. Specifically, the risk is estimated by using the ground risk assessment method described in Section II.

An example of graph extension and risk assessment is illustrated in Figure 3. The risk is computed considering the location of the node, the flight direction and velocity. As previously detailed, these parameters are used to estimate a probabilistic impact area, and, then, the ground risk is computed taking into account the population density and the sheltering factor in the impact area.

Similarly to RRT*, the algorithm explores the state space by minimizing a motion cost. In this paper, the algorithm minimizes the risk in respect of the flight time. In fact, as described in Section II, the risk is expressed in casualties *per hour* and, then, the risk of flying over a specific location is proportional with the time of overflight the location. Hence, the motion cost at the node x_i in the state space is defined as

$$\text{Cost}(x_i) = \text{Cost}(x_{i-1}) + \int_{t(x_{i-1})}^{t(x_i)} P_c(x) dt \quad (10)$$

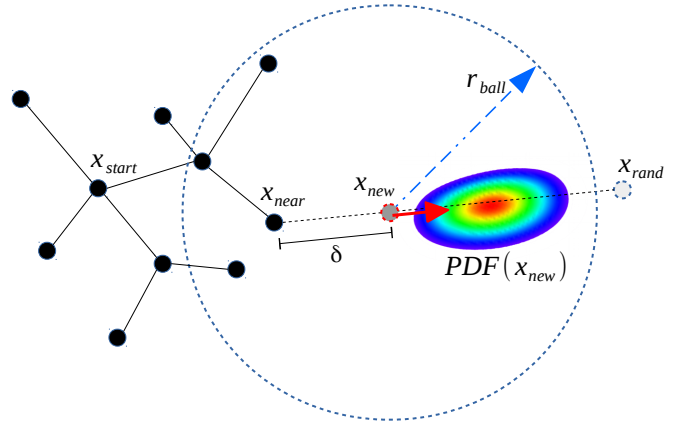


Figure 3. Example of graph extension with the risk-based RRT*. As detailed in Algorithm 1, after the definition of a new node x_{new} , the algorithm computes the motion cost of moving from $x_{\text{near}} \in \mathcal{X}_{\text{near}}$ to x_{new} by assessing the risk of flying over x_{new} considering a specific flight direction and velocity. The red arrow is the flight direction, while the ellipsoid is the 2D-PDF defining the probabilistic impact area.

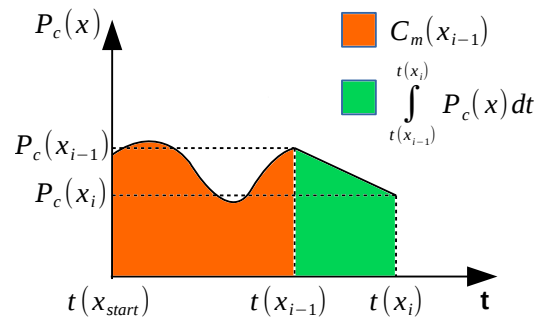


Figure 4. Representation of the computation of the motion cost at the node x_i considering the risk in respect of the flight time. In orange the motion cost at the previous node x_{i-1} , while in green the trapezoidal area that represents the motion cost of moving from x_{i-1} to x_i .

where $\text{Cost}(x_{i-1})$ is the motion cost computed at the adjacent node x_{i-1} , while the motion cost from x_{i-1} and x_i is the integral of the risk $P_c(x)$ on the flight time. In Equation 10, the cost function $\text{Cost}(x_i)$ always refers to the motion cost from the initial node x_{start} to x_i because, typically, the graph of RRT-based algorithms is rooted from the initial node.

Practically, the risk cost function is computed with an approximate and incremental method where the integral is defined as a trapezoidal area between two adjacent nodes

$$\begin{aligned} \text{Cost}(x_i) &= \text{Cost}(x_{i-1}) + c(x_{i-1}, x_i) \\ &= \text{Cost}(x_{i-1}) + \frac{P_c(x_{i-1}) + P_c(x_i)}{2} t(x_{i-1}, x_i) \end{aligned} \quad (11)$$

where $t(x_{i-1}, x_i)$ is the time flight expressed in hour to move from the node x_{i-1} and the node x_i . A graphical representation of the risk function is shown in Figure 4.

The risk-based RRT* algorithm is a variant of the original RRT*. The pseudocode of the algorithm is reported in Algorithm 1, where the algorithm differs from RRT* only in lines 15 and 26, i.e. where the risk is computed during the graph extension and rewiring procedures.

Similarly to other RRT-based algorithm, risk-based RRT* constructs a graph $\mathcal{G} = (V, E)$ embedded in \mathcal{X} and rooted from x_{start} , with V is the set of nodes and E is the set of edges. As result, the algorithm returns a graph with a unique branch that connects x_{goal} with x_{start} , which corresponds with the solution path.

First of all, the graph G is initialized by inserting x_{start} as a node in the set of nodes V and setting the set of edges E as an empty set. The algorithm executes an iterative routine (lines from 3 to 34) that continues until a certain number n of nodes are sampled. Hence, a new node x_{rand} is sampled (line 4) and the algorithm searches for the nearest node x_{near} in the graph. If the Euclidean distance between x_{nearest} and x_{rand} is higher than the maximum planner range δ , a new node x_{new} is defined at the distance δ (line 7), otherwise it corresponds to x_{rand} (line 9).

Then, if x_{new} is valid, i.e. it is not inside an obstacle or a no-fly zone, the algorithm searches for the best parent node among the near nodes (lines from 12 to 20). Specifically, the set X_{near} consists in all nodes within a ball of radius r_{ball} centered in x_{new} . According to [24], the radius r_{ball} is defined as

$$r_{\text{ball}} = \min \left\{ \gamma \left(\frac{\log |V|}{|V|} \right)^{1/d}, \delta \right\}, \quad (12)$$

with γ a parameter of the algorithm, d is the dimension of the Euclidean state space \mathcal{X} , and the notation $|\cdot|$ defines the cardinality of a set.

For each state x_{near} of the set, the algorithm assesses the risk of flying from x_{near} to x_{new} (line 15) and computes the motion cost using the formulation of Equation 11 (line 16). The best parent node is the node that provides the lowest motion cost. If a parent node exists, the node x_{new} and the edge connecting it to the best parent are added to the graph (lines 22, 23).

In order to guarantee that nodes in the graph are reached by the minimum cost path, the rewiring routine is executed (lines from 25 to 32). For each state x_{near} the algorithm verifies if the motion cost of the node x_{near} is lower passing through the new node x_{new} . If it is the case, the graph is updated by setting the node x_{new} as a new parent of x_{near} . The rewiring routine is an essential element of the RRT* algorithm, because it updates the edges of the graph evaluating the new node x_{new} and maintaining an optimal graph.

At the end of the iterative procedure, the algorithm returns a solution path as the unique branch in the graph G connecting the node x_{goal} to x_{start} . If the graph does not reach the goal, a solution does not exist.

As demonstrated in [24], RRT* is asymptotically optimal if the parameter γ satisfies the following inequality

$$\gamma > 2 \left(1 + \frac{1}{d} \right)^{1/d} \left(\frac{\mu(X_{\text{free}})}{\zeta_d} \right)^{1/d} \quad (13)$$

where $\mu(X_{\text{free}})$ is the Lebesgue measure (i.e., volume) of the free search space X_{free} , and ζ_d is the volume of the unit ball in the d -dimensional Euclidean state space.

Algorithm 1 risk-based RRT* algorithm

```

1:  $V \leftarrow \{x_{\text{start}}\}$ 
2:  $E \leftarrow \emptyset$ 
3: for  $i = 1, \dots, n$  do
4:    $x_{\text{rand}} \leftarrow \text{SampleNode}()$ 
5:    $x_{\text{nearest}} \leftarrow \text{Nearest}(G = (V, E), x_{\text{rand}})$ 
6:   if  $d(x_{\text{nearest}}, x_{\text{rand}}) > \delta$  then
7:      $x_{\text{new}} \leftarrow \text{Saturate}(x_{\text{rand}}, x_{\text{nearest}}, \delta)$ 
8:   else
9:      $x_{\text{new}} \leftarrow x_{\text{rand}}$ 
10:  end if
11:  if  $\text{isValid}(x_{\text{new}})$  then
12:     $X_{\text{near}} \leftarrow \text{Near}(G, x_{\text{new}}, r_{\text{ball}})$ 
13:     $c_{\text{min}} \leftarrow \text{InfiniteCost}$ 
14:    for each  $x_{\text{near}} \in X_{\text{nearest}}$  do
15:       $\text{RiskAssessment}(x_{\text{near}}, x_{\text{new}})$ 
16:      if  $\text{Cost}(x_{\text{near}}) + c(x_{\text{near}}, x_{\text{new}}) < c_{\text{min}}$  then
17:         $x_{\text{min}} \leftarrow x_{\text{near}}$ 
18:         $c_{\text{min}} \leftarrow \text{Cost}(x_{\text{near}}) + c(x_{\text{near}}, x_{\text{new}})$ 
19:      end if
20:    end for
21:    if  $c_{\text{min}} \neq \text{InfiniteCost}$  then
22:       $V \leftarrow V \cup \{x_{\text{new}}\}$ 
23:       $E \leftarrow E \cup \{(x_{\text{min}}, x_{\text{new}})\}$ 
24:    end if
25:    for each  $x_{\text{near}} \in X_{\text{near}}$  do
26:       $\text{RiskAssessment}(x_{\text{new}}, x_{\text{near}})$ 
27:      if  $\text{Cost}(x_{\text{new}}) + c(x_{\text{new}}, x_{\text{near}}) < \text{Cost}(x_{\text{near}})$  then
28:         $x_{\text{parent}} \leftarrow \text{Parent}(x_{\text{near}})$ 
29:         $E \leftarrow E \setminus \{(x_{\text{parent}}, x_{\text{near}})\}$ 
30:         $E \leftarrow E \cup \{(x_{\text{new}}, x_{\text{near}})\}$ 
31:      end if
32:    end for
33:  end if
34: end for
35: return  $\text{getPath}(G, x_{\text{goal}})$ 

```

IV. RESULTS

The proposed risk-based path planning method is implemented in C++ as an executable process in the ROS (Robot Operating System) framework [29].

The multi-layer framework is implemented using Grid Map [30], a C++ library with ROS interface to manage two-dimensional grid maps with multiple data layers. The risk assessment and the estimation of the probabilistic impact area is performed using the OpenCV library [31] to provide fast matrix computation.

The risk-based RRT* algorithm is implemented using the OMPL (Open Motion Planning Library) library [32] that consists in many state-of-the-art sampling-based motion planning algorithms and offers many functionalities to facilitate the implementation of a new algorithm.

The proposed strategy is tested considering an area of Torino (Italy) and taking into account a realistic population density and sheltering factor data. Figure 5 illustrates the map

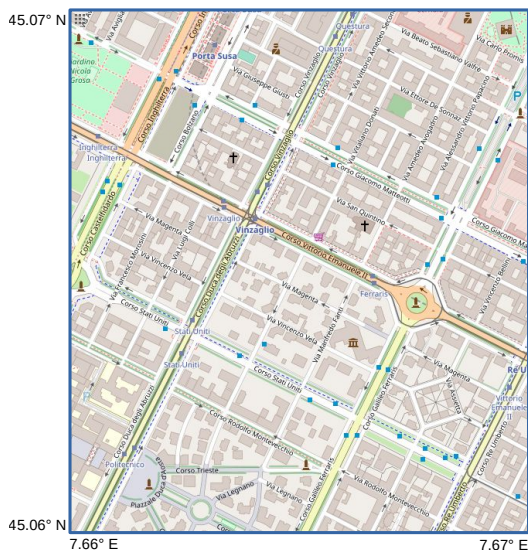


Figure 5. The map from OpenStreetMap of the Torino’s neighborhood evaluated to test the proposed risk-based path planning strategy.

of Torino’s neighbourhood obtained from OpenStreetMaps (OSM) with a dimension of 1020×1170 m.

In order to show the risk involved in this area, Figure 6 shows the risk-based map computed using the method proposed in [23]. The risk-based map is computed assuming the DJI Mavic Pro aircraft, a small quadcopter with a mass of about 0.7 kg and considering a cruise speed of 10 m/s. In order to ensure an appropriate level of safety, the risk of a flight operation should be lower than a *maximum acceptable risk*. As discussed in [12], [33], a conservative ”Equivalent Level Of Safety” (ELOS) is $1 \cdot 10^{-6} \text{h}^{-1}$. The risk-based map of Figure 6 has an average risk of $1.72 \cdot 10^{-6} \text{h}^{-1}$ and, taking into account the ELOS requirement, the Mavic Pro aircraft can fly only over few areas.

In this paper the path planning is executed considering some assumptions that simplify the scenario: (i) the path planning considers flight at constant altitude, then, a two-dimensional state space is used, in which a node x is represented with a 2D-position in the space; (ii) the UAS flies with a constant velocity; (iii) the UAS direction in a node x_i is defined by the direction of the segment connecting x_{i-1} and x_i , as illustrated in Figure 3.

The proposed risk-based RRT* is used to compute the minimum risk path in the map of Figure 5. The resulting path is computed by constructing a graph composed by 10000 nodes, with a maximum planner range of $\delta = 20$ m and with $\gamma = 1174$ satisfying the condition of Equation 13. The path is reported in Figure 7, while the path length, the motion cost and the average risk of the path are reported in Table I. Specifically, the path has an average risk of $1.49 \cdot 10^{-6} \text{h}^{-1}$, which is higher than the ELOS. The same path is also evaluated as in [22] using risk costs of the risk-based map of Figure 6 obtaining an average risk of $9.85 \cdot 10^{-7} \text{h}^{-1}$, lower than the ELOS.

The same path planning problem is also solved using the risk-aware path planning proposed in [22], where the RRT*

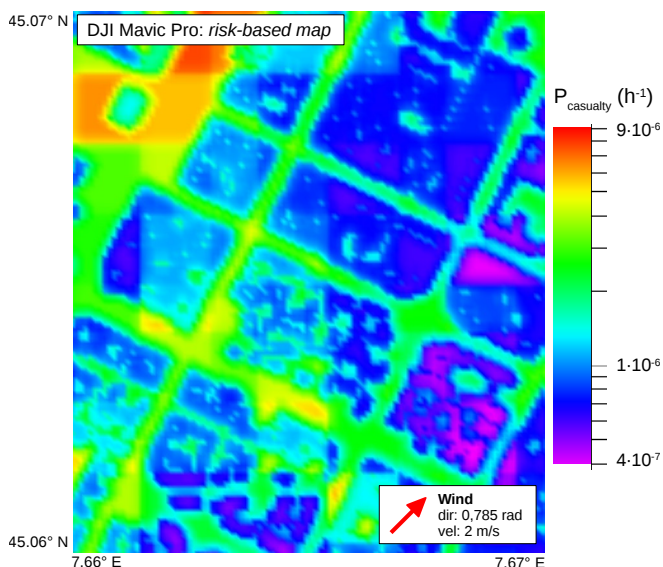


Figure 6. The risk-based map of the urban area of Figure 5. The risk-based map is computed with the strategy proposed in [23] and taking into account the DJI Mavic Pro aircraft at the flight altitude of 30 m and a wind with direction 0.785 rad and velocity 2 m/s.

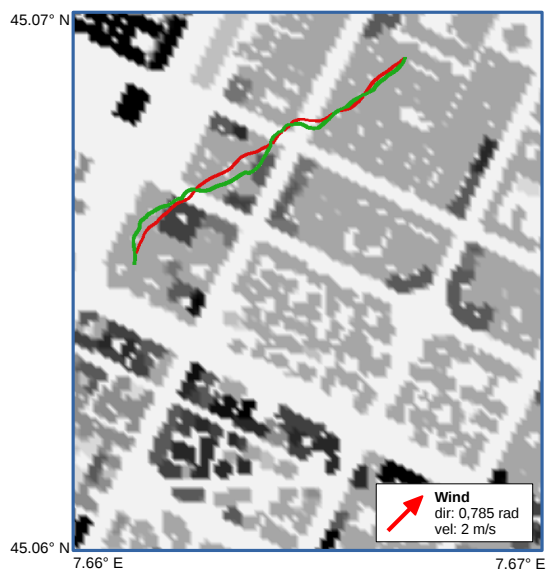


Figure 7. The minimum risk paths computed in the area of Figure 5 and considering the DJI Mavic Pro aircraft. In green the path computed with the risk-based RRT* proposed in this paper. In red the path computed using the risk-aware path planning proposed in [22]. Both paths are computed considering a flight altitude of 30 m and a wind with direction 0.785 rad and speed 2 m/s.

TABLE I
COMPARISON BETWEEN THE PROPOSED RISK-BASED RRT* AND THE RISK-AWARE PATH PLANNING STRATEGY PROPOSED IN [22].

Algorithm	Solve time [s]	Path length [m]	Optimization method	Motion cost	Average risk [h]
risk-based RRT*	2.42	813.12	actual risk	$1.21 \cdot 10^{-4}$	$1.49 \cdot 10^{-6}$
			risk cost	$8.01 \cdot 10^{-5}$	$9.85 \cdot 10^{-7}$
risk-aware RRT* [22]	0.38	771.31	risk cost	$7.64 \cdot 10^{-5}$	$9.90 \cdot 10^{-7}$
			actual risk	$1.28 \cdot 10^{-4}$	$1.65 \cdot 10^{-6}$

algorithm minimizes risk costs of a risk-based map. Hence, Figure 7 illustrates the path considering the same starting and target positions and the risk-based map of Figure 6, while the characteristics of the path are in Table I. The path has an average risk of $9.90 \cdot 10^{-7} \text{h}^{-1}$, but, if the same path is evaluated using the risk assessment strategy proposed in this work, the actual risk is $1.65 \cdot 10^{-6} \text{h}^{-1}$, higher than the ELOS.

Comparing the results of the two path planning approaches in Table I, it is clear that the risk assessment used in the optimization of the risk-based RRT* algorithm estimates a higher risk level compared with the strategy proposed in [22]. In fact, according to the method proposed in [22], the Mavic Pro is suitable to flight over the urban area of Figure 5. On the contrary, it involves an excessive risk using the strategy proposed in this work. Anyway, both algorithms compute a near-optimal solution in the map, but they use different assumptions about the risk assessment, obtaining different results.

This happens because risk costs of the risk-based map are defined evaluating a wide impact area and, as a consequence, the resulting risk is distorted by all values of population density and sheltering factor distributed in the impact area, resulting in an overestimated or, in the worst case, an underestimated risk. Moreover, risk costs are computed considering all velocities uniformly distributed from 0 m/s to the cruise speed. Since the risk is generally proportional to the initial velocity, risk costs estimate a lower risk compared with the actual risk estimated using a specific cruise velocity. On the contrary, the proposed risk-based path planning evaluates a more appropriate impact area and, as a consequence, estimates a more realistic risk.

However, as expected, the time required to solve the path planning problem with the proposed approach is higher than the risk-aware path planning of [22] because the proposed algorithm performs the risk assessment many times during the exploration of the state space increasing the time complexity of the algorithm.

The main advantage of the proposed path planning approach is shown in Figure 8, where the total probabilistic impact area evaluated along the whole path is drawn on the map. Specifically, Figure 8(a) illustrates the area analysed by the proposed approach considering the flight direction and velocity of each node of the path. On the other hand, Figure 8(b) shows the area evaluated by the path planning proposed in [22]. In Figure 8(b) the evaluated impact area is much greater than the area of Figure 8(a), because the path planning approach used in [22] minimizes risk costs of a risk-based map, in which the risk is computed considering any flight direction of the UAS. In fact, the risk-based map is generated before the path planning, without any knowledge about the UAS flight conditions.

The same path is also computed taking into account a DJI Mavic Mini aircraft, a small and light-weight drone with a mass of 0.249 kg. The resulting path is illustrated in Figure 9, while the characteristics of the path are in Table II. The path has an average risk of $1.91 \cdot 10^{-7}$, lower than the ELOS. It should be noted that the solve time to compute the path

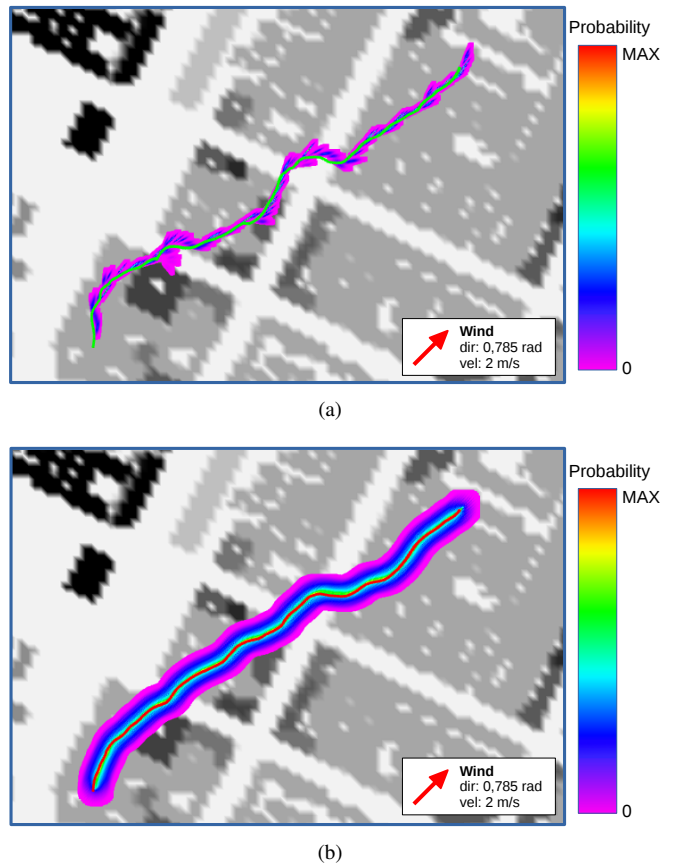


Figure 8. Comparison between probabilistic impact areas. In (a) the impact area evaluated by the proposed risk-based RRT* algorithm, while, in (b), the area evaluated by the risk-aware path planning presented in [22]. The whole area evaluated by the proposed algorithm is much smaller, because it estimates a more appropriate probabilistic impact area along the path.

TABLE II
NUMERICAL RESULTS OF THE RISK-BASED RRT* USING THE DJI
MAVIC MINI.

Algorithm	Solve time [s]	Path length [m]	Optimization method	Motion cost	Average risk [h]
risk-based RRT*	1.36	819.98	actual risk	$1.56 \cdot 10^{-5}$	$1.91 \cdot 10^{-7}$
			risk cost	$1.08 \cdot 10^{-5}$	$1.31 \cdot 10^{-7}$

is 1.36 s, lower than the time required to compute the path with the Mavic Pro. This is caused by a smaller probabilistic impact area that requires less time to visit all the locations in the estimated 2D-PDF.

Another interesting scenario is shown in Figure 10 where the proposed risk-based path planning strategy is used to compute a path over a big crossroad in the presence of wind. Specifically, in Figure 10(a) the wind has a direction of 1.57 rad, while in Figure 10(b) has an opposite direction of -1.57 rad. As a consequence, the wind modifies the probabilistic impact areas evaluated during the construction of the path obtaining two different solutions. Anyway, as illustrated in Figure 8 the probabilistic impact areas never lie in the middle of the crossroad that involves a high risk.

V. CONCLUSIONS

In this paper, we have presented a novel risk-based path planning strategy to compute optimum risk path for UAS over populated areas. The proposed approach uses a variant of the well-know RRT* with the minimization of the overall risk along the path and the flight time.

Unlike in our previous work [22], where the RRT* algorithm is used to minimize risk costs of a risk-based map, in the proposed strategy the risk assessment is performed during the path planning phase. Specifically, each time a new node is added to the exploration graph, a risk assessment procedure is performed evaluating the flight direction and velocity and estimating a probabilistic impact area. As a consequence, the proposed risk-based path planning evaluates a more appropriate and reliable risk compared with the approach presented in [22].

Results show how the proposed strategy is able to compute the minimum risk path in an urban area. Moreover, the computed path is compared with the path planning presented in [22]. As reported in the results, the proposed path planning does not compute a path with a lower risk, but it evaluates a realistic area obtaining a more effective solution. In fact, with the method proposed in [22], the risk is computed by using a large impact area because the risk estimated in the risk-based map evaluates any flight direction and velocity. This may cause an overestimated or, in the worst case, an underestimated risk. However, the proposed strategy has a higher time complexity due to the risk assessment that is performed multiple times at each iteration of the risk-based RRT*. Anyway, the path planning can be executed on a remote server, such as on a Cloud platform, exploiting the Cloud and parallel computing, reducing the computation time of the algorithm.

Future works will include the inclusion of kinodynamic constraints of the vehicle and the adaptation to a tridimensional environment. Also the energy consumption of the aircraft should be taken into account to obtain an energy-efficient and safe path for UAS. Moreover, experimental tests will be conducted on a real robotic platform.

ACKNOWLEDGMENT

This work is partially supported by Compagnia di San Paolo and by an Amazon Research Award granted to Dr. A. Rizzo.

REFERENCES

- [1] H. González-Jorge, J. Martínez-Sánchez, M. Bueno *et al.*, “Unmanned aerial systems for civil applications: A review,” *Drones*, vol. 1.1, 2017.
- [2] N. Bloise, S. Primatesta, R. Antonini, G. P. Fici, M. Gaspardone, G. Guglieri, and A. Rizzo, “A survey of unmanned aircraft system technologies to enable safe operations in urban areas,” in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2019, pp. 433–442.
- [3] T. H. Grubestic and J. R. Nelson, “Uavs and urban spatial analysis,” 2020.
- [4] N. Mohamed, J. Al-Jaroodi, I. Jawhar, A. Idries, and F. Mohammed, “Unmanned aerial vehicles applications in future smart cities,” *Technological Forecasting and Social Change*, p. 119293, 2018.

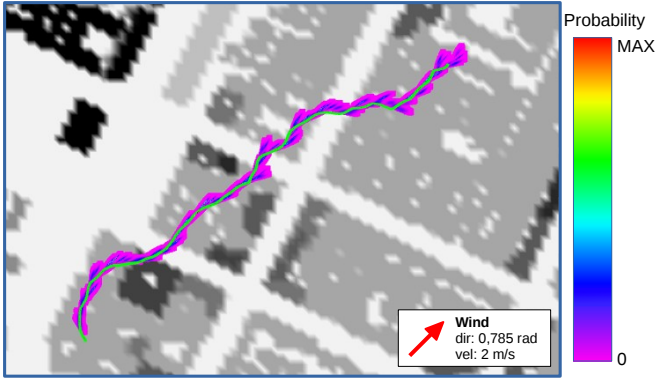
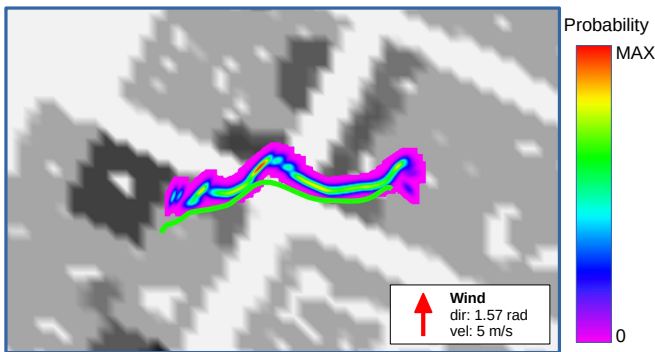
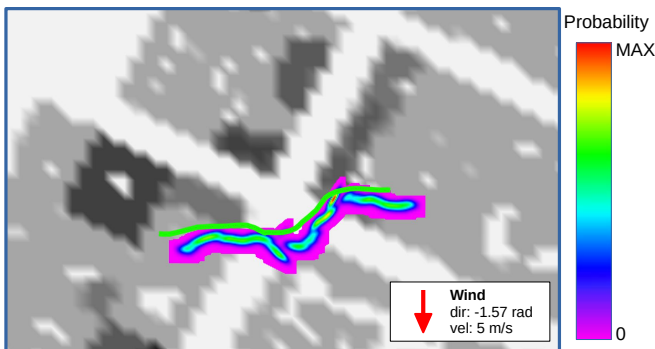


Figure 9. The minimum risk path computed in the area of Figure 5 and considering the DJI Mavic Mini aircraft. The path is computed considering a flight altitude of 30 m and a wind with direction 0.785 rad and speed 2 m/s.



(a)



(b)

Figure 10. Example of the proposed risk-based path planning approach applied in an urban scenario with different wind parameters and a flight altitude of 30 m. In (a) the wind has a speed of 5 m/s and direction 1.57 rad. In (b) the wind has an opposite direction of -1.57 rad. The resulting paths are different because the estimated impact areas are modified by the wind.

- [5] A. Nayyar, B.-L. Nguyen, and N. G. Nguyen, "The internet of drone things (iodt): Future envision of smart drones," in *First International Conference on Sustainable Technologies for Computational Intelligence*. Springer, 2020, pp. 563–580.
- [6] E. Vattapparamban, İ. Güvenç, A. İ. Yurekli, K. Akkaya, and S. Uluğaç, "Drones for smart cities: Issues in cybersecurity, privacy, and public safety," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2016, pp. 216–221.
- [7] EASA, "Commission delegated regulation (eu) 2019/945," European Aviation Safety Agency, Tech. Rep., 2019.
- [8] —, "Commission implementing regulation (eu) 2019/947," European Aviation Safety Agency, Tech. Rep., 2019.
- [9] JARUS, "Jarus guidelines on specific operations risk assessment (SORA), joint authorities for rulemaking of unmanned systems JARUS," http://jarus-rpas.org/sites/jarus-rpas.org/files/jar_doc_06_jarus_sora.v2.0.pdf, Tech. Rep., 2019.
- [10] A. Washington, R. A. Clothier, and J. Silva, "A review of unmanned aircraft system ground risk models," *Progress in Aerospace Sciences*, vol. 95, pp. 24–44, 2017.
- [11] R. A. Clothier, R. A. Walker, N. Fulton, and D. A. Campbell, "A casualty risk analysis for unmanned aerial system (uas) operations over inhabited areas," in *AIAC12, Twelfth Australian International Aerospace Congress, 2nd Australasian Unmanned Air Vehicles Conference*, 2007, pp. 1–15.
- [12] K. Dalamagkidis, K. P. Valavanis, and L. A. Piegl, *On integrating unmanned aircraft systems into the national airspace system: issues, challenges, operational restrictions, certification, and recommendations*. Springer Netherlands, 2012, vol. 54.
- [13] A. La Cour-Harbo, "Quantifying risk of ground impact fatalities of power line inspection bvlos flight with small unmanned aircraft," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*. IEEE, 2017, pp. 1352–1360.
- [14] S. Bertrand, N. Raballand, F. Viguier, and F. Muller, "Ground risk assessment for long-range inspection missions of railways by uavs," in *ICUAS 2017, International Conference on Unmanned Aircraft Systems*. IEEE, 2017, pp. 1343–1351.
- [15] A. la Cour-Harbo, "The value of step-by-step risk assessment for unmanned aircraft," in *ICUAS 2018, International Conference on Unmanned Aircraft Systems*. IEEE, 2018.
- [16] S. Primatesta, G. Guglieri, and A. Rizzo, "A risk-aware path planning strategy for uavs in urban environments," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 2, pp. 629–643, 2019.
- [17] E. Ancel, F. M. Capristan, J. V. Foster, and R. C. Condotta, "Real-time risk assessment framework for unmanned aircraft system (uas) traffic management (utm)," in *17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017, p. 3273.
- [18] X. Zhang, Y. Liu, Y. Zhang, X. Guan, D. Delahaye, and L. Tang, "Safety assessment and risk estimation for unmanned aerial vehicles operating in national airspace system," *Journal of Advanced Transportation*, vol. 2018, 2018.
- [19] E. Rudnick-Cohen, J. W. Herrmann, and S. Azarm, "Risk-based path planning optimization methods for unmanned aerial vehicles over inhabited areas," *Journal of Computing and Information Science in Engineering*, vol. 16, no. 2, p. 13, 2016.
- [20] E. Rudnick-Cohen, S. Azarm, and J. W. Herrmann, "Planning unmanned aerial system (uas) takeoff trajectories to minimize third-party risk," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2019, pp. 1306–1315.
- [21] J. Castagno, C. Ochoa, and E. Atkins, "Comprehensive risk-based planning for small unmanned aircraft system rooftop landing," in *ICUAS 2018, International Conference on Unmanned Aircraft Systems*. IEEE, 2018.
- [22] S. Primatesta, L. S. Cuomo, G. Guglieri, and A. Rizzo, "An innovative algorithm to estimate risk optimum path for unmanned aerial vehicles in urban environments," *Transportation research procedia*, vol. 35, pp. 44–53, 2018.
- [23] S. Primatesta, A. Rizzo, and A. la Cour-Harbo, "Ground risk map for unmanned aircraft in urban environments," *Journal of Intelligent & Robotic Systems*, pp. 1–21.
- [24] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.
- [25] A. la Cour-Harbo and H. Schioler, "Ground impact probability distribution for small unmanned aircraft in ballistic descent," *Reliability engineering and system safety Submitted*, 2017.
- [26] A. la Cour-Harbo, "Quantifying risk of ground impact fatalities for small unmanned aircraft," *Journal of Intelligent & Robotic Systems*, vol. 93, no. 1-2, pp. 367–384, 2019.
- [27] G. Guglieri, F. Quagliotti, and G. Ristorto, "Operational issues and assessment of risk for light uavs," *Journal of Unmanned Vehicle Systems*, vol. 2, no. 4, pp. 119–129, 2014.
- [28] Range Commanders Council, "Standard 321-07 Common risk criteria standards for national test ranges: Supplement"," *USA Dept. of Defense*, 2007.
- [29] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, 2009, p. 5.
- [30] P. Fankhauser and M. Hutter, "A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation," in *Robot Operating System (ROS) – The Complete Reference (Volume 1)*, A. Koubaa, Ed. Springer, 2016, ch. 5.
- [31] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [32] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December 2012, <http://ompl.kavrakilab.org>.
- [33] G. Guglieri, A. Lombardi, and G. Ristorto, "Operation oriented path planning strategies for rpas," *American Journal of Science and Technology*, vol. 2, no. 6, pp. 1–8, 2015.