

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Energy Efficient Path Planning for 3D Aerial Inspections

RAFAEL M. CLARO^{1,2}, MARIA I. PEREIRA^{1,2}, FRANCISCO S. NEVES^{1,2} AND ANDRY M. PINTO^{1,2}

¹Faculty of Engineering, University of Porto, 4200-465 Porto, Portugal

²Centre for Robotics and Autonomous Systems, INESC TEC, 4200-465 Porto, Portugal

Corresponding author: Rafael M. Claro (e-mail: rafael.marques@inesctec.pt).

This work is partly funded by the Portuguese Government through the FCT - Foundation for Science and Technology, 2021.05841.BD (to Rafael Claro), by the European Union's Horizon 2020 research and innovation programme under grant agreement No 871571, and by FLY.PT - P2020 Mobilizador within project POCI-01-0247-FEDER-046079.

ABSTRACT The use of Unmanned Aerial Vehicles (UAVs) in different inspection tasks is increasing. This technology reduces inspection costs and collects high quality data of distinct structures, including areas that are not easily accessible by human operators. However, the reduced energy available on the UAVs limits their flight endurance. To increase the autonomy of a single flight, it is important to optimize the path to be performed by the UAV, in terms of energy loss. Therefore, this work presents a novel formulation of the Travelling Salesman Problem (TSP) and a path planning algorithm that uses a UAV energy model to solve this optimization problem. The novel TSP formulation is defined as Asymmetric Travelling Salesman Problem with Precedence Loss (ATSP-PL), where the cost of moving the UAV depends on the previous position. The energy model relates each UAV movement with its energy consumption, while the path planning algorithm is focused on minimizing the energy loss of the UAV, ensuring that the structure is fully covered. The developed algorithm was tested in both simulated and real scenarios. The simulated experiments were performed with realistic models of wind turbines and a UAV, whereas the real experiments were performed with a real UAV and an illumination tower. The inspection paths generated presented improvements over 24% and 8%, when compared with other methods, for the simulated and real experiments, respectively, optimizing the energy consumption of the UAV.

INDEX TERMS Inspection path planning, Coverage path planning, Energy Model, Unmanned Aerial Vehicle, Energy efficiency, Travelling Salesman Problem

I. INTRODUCTION

Traditional methods for inspecting and monitoring diverse types of structures are often expensive, repetitive, time consuming and potentially dangerous for human operators [1]. To overcome this issue, several inspection tasks are currently being carried out by Unmanned Aerial Vehicles (UAVs) [2], [3], which represents a reduction of the inspection costs and of the total time consumed [4]. UAVs are autonomous aerial robots that can collect high quality data of structures [5], such as images (visual, thermal, among others) and point clouds. These vehicles can also easily access the entire surface of inspection. However, one of the challenges related with the use of UAVs in autonomous inspections is related with their limited battery capacity [5]. To allow a UAV to perform a wider inspection in a single flight, it is important to increase its endurance by selecting trajectories that optimize

the energy consumption during an inspection procedure. A path that minimizes the total displacement performed by a UAV does not guarantee an optimized energy consumption. Therefore, a preferable approach consists in selecting a path that directly minimizes the energy consumption of a UAV (that may not be the shortest path). This can be achieved by smoothing the trajectory of the UAV, by reducing its accelerations during the performance of the path. Therefore, this work presents an algorithm for 3D path planning for inspection of structures with UAVs, minimizing the energy loss. As a result, the contributions of this work are the following:

- 1) A novel Travelling Salesman Problem (TSP) formulation, named ATSP-PL (Asymmetric Travelling Salesman Problem, with Precedence Loss), of which the Precedence Loss designation is related to the objective

- function, where the loss of moving from a current position to the next one depends on the previous one.
- 2) An energy model of a UAV that relates each movement that the vehicle performs with its correspondent energy loss, and a resulting algorithm for inspection path planning that optimizes the energy consumption of a UAV, enabling it to autonomously perform a full coverage of the structure being inspected;
 - 3) Validation of the developed algorithm in both simulated and real scenarios, using realistic models of wind turbines and of a UAV in the first case, and a real UAV along with an illumination tower in the second one.

This article is organized as follows: Section II presents a brief review of the related work. Afterwards, section III presents the algorithm developed to obtain the inspection path and the respective UAV energy model. Experimental evaluation is demonstrated in sections IV and V, which includes experiments in both simulation scenarios and in a real environment. Finally, section VI clarifies the most important conclusions of this research.

II. RELATED WORK

The use of UAVs in different types of inspection tasks is increasing significantly. The most common sensors used for inspections with UAVs are both visual and thermal cameras [6], [7]. These sensors allow to execute non-destructive tests, capable of detecting the most significant in-service defects [7]. To improve the quality of the data collected by a camera sensor during a inspection, a LiDAR sensor can also be used [8], [9]. A 3D LiDAR represents a relevant method to obtain the characteristics of the inspected structure, given its capability of directly obtaining depth, resulting in accurate 3D point clouds of the surrounding environment [10]. These 3D sensors are used in robotics to improve the accuracy in distinct tasks, such as in object detection [11], [12], scene reconstruction and navigation [13], [14].

Although a UAV equipped with the required sensors is able to collect the necessary information for an inspection task, its limited flight endurance represents a considerable drawback. To increase the autonomy of the UAVs, it is important to optimize the path that the vehicles will perform as a function of its energy consumption, while assuring that the entire surface of inspection is completely verified. The energy model of a UAV represents the energy loss relative to each individual movement of the vehicle, such as the cost of displacement and the cost of directional changes [15]. Each UAV has its own particular characteristics, such as weight, dimensions, propellers and motors type. For that reason, it is not easy to derive a general parametric model that represents the energy consumption of each different UAV. Therefore, one possible way consists in performing tests in each particular UAV. Di Franco *et al.* [16], [17] performed a complete analysis on the energy loss based on real measurements of the UAV IRIS quadrotor, powered by a 3S LiPo battery with 5.5 Ah of capacity. Each movement, including moving forward, ascending/descending, and accel-

eration/deceleration, was analysed to create a model of the energy consumed relative to each UAV action. Besides its movement, the influence of environmental conditions also takes an important role on the autonomy of the UAV, such as the wind speed and its direction, and the atmospheric flow conditions [5]. These factors interfere with the stability of the UAV, resulting in energy losses required to compensate those external forces.

To achieve the least expensive sequence of inspection, it is possible to apply several path planning algorithms. The most adequate approach can depend on the shape of the structure to be inspected. A geometric method may result in appropriate paths for regular structures (like a cube or a cylinder) with a reduced computation time, whereas an heuristic approach may result in optimized paths for more complex structures. Given that the inspection path can be resampled in a set of viewpoints of the structure [18], [19], this problem can be formulated as the well-known Travelling Salesman Problem [15], [20]. Examples of used optimization techniques are genetic algorithms [15], ant colony [21], and particle swarm optimization (PSO) based algorithms [20], [22], among other heuristics and metaheuristics. A possible approach to reduce the complexity of the problem consists in dividing the structure in multiple sub-spaces of inspection. First it is computed the order that each sub-space will be inspected, and secondly it is obtained the path within each sub-space [23].

III. ENERGY EFFICIENT INSPECTION PATH PLANNING

Several steps need to be performed to achieve an energetically optimized path to inspect complex structures with a UAV, which are depicted in Fig. 1, that takes a wind turbine model as an example. Given a structure's model, it is possible to discretize its surface and define a map of the inspection poses, named viewpoints. A UAV's energy model can be defined by relating each movement with the correspondent energy loss. With these two formulation steps completed, an algorithm for path planning was developed, ensuring that the entire surface of interest is covered during the inspection, optimizing the energy consumption. By analysing the data collected during the inspection it is possible to reconstruct the structure's model, as well as to draw a confidence map that allows to pinpoint the confidence on the quality of the data in each location.

A. DEFINING THE MAP OF THE INSPECTION POSES

The map of the inspection poses consists in defining a set of viewpoints that ensure that the entire surface of interest of the structure is completely covered. The definition of these viewpoints depends upon the characteristics of the sensors used in the inspection, such as cameras and 3D LiDARs, and also on the structure's shape. To guarantee the coverage of the entire surface of the inspected structure, these sensors constrain the definition of the inspection map, through the following characteristics [16]:

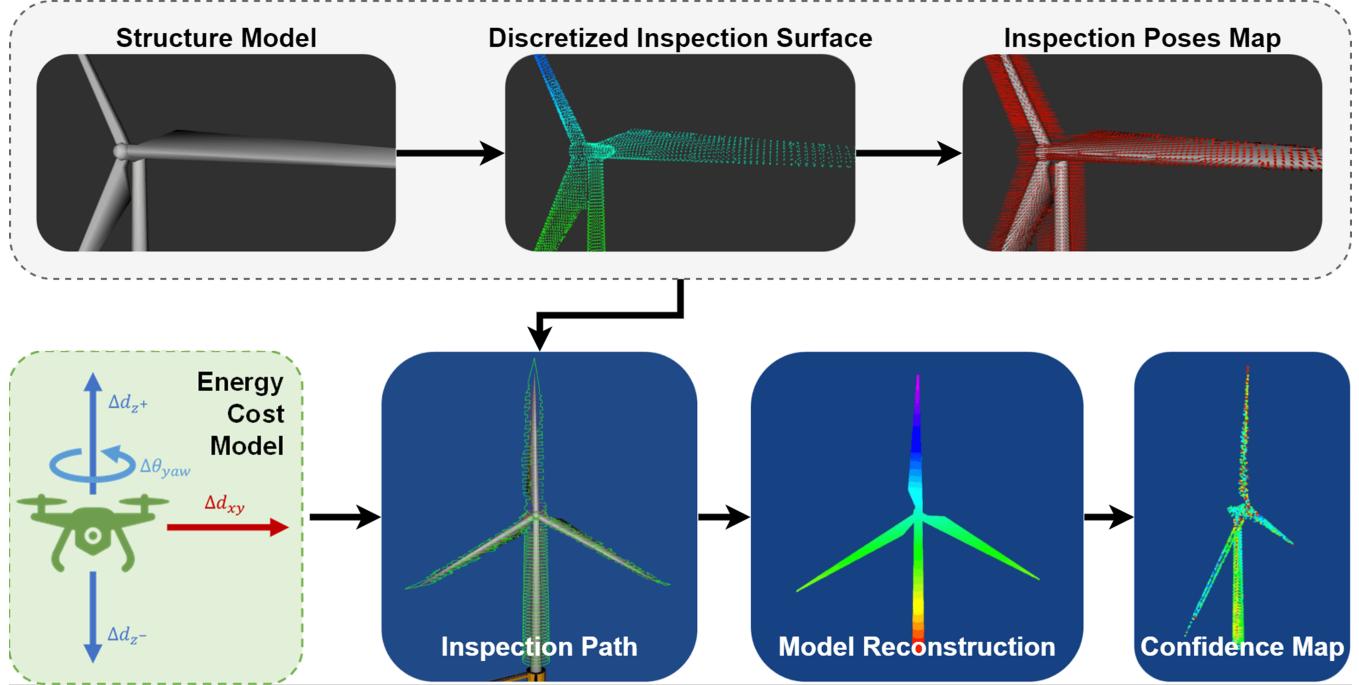


FIGURE 1. Scheme of the developed inspection path components, including the mapping of the relevant viewpoints, the UAV energy model, the generated inspection path and the correspondent model reconstruction and confidence map.

- α - Field of View (FoV), represents the angular aperture of the data captured by the sensor;
- (I_x, I_y) - data resolution, expresses the size of the captured data in the smallest addressable element;
- $\rho = \frac{I_x}{I_y}$ - aspect ratio, represents the relation between the width and height of the data captured by the sensor;
- f_S - sampling frequency, represents the time expenditure between two consecutive data collections.

When the UAV is moving at a given working distance (d_w) from the target, a sensor acquires data that covers a projected area ($L_x \times L_y$) that depends on its field of view (α) and aspect ratio (ρ), as can be seen in Fig. 2. This relation is expressed in (1) [16]:

$$L_x = 2d_w \cdot \tan\left(\frac{\alpha}{2}\right), \quad L_y = \frac{L_x}{\rho}. \quad (1)$$

To ensure a minimum spacial resolution ($R = \max(\frac{I_x}{L_x}, \frac{I_y}{L_y})$), the maximum working distance (d_w^{\max}) must be as follows (2) [16]:

$$d_w^{\max} = \frac{\min(I_x, I_y)}{2R \cdot \tan\left(\frac{\alpha}{2}\right)}. \quad (2)$$

To discretize the entire surface of the inspected model, ensuring that the whole structure is covered, the discretization rate (D_r) must be lower than or equal to $\min(L_x, L_y)$. After the discretization is performed, each point represents an inspection point of the structure that the UAV must meet. The last constraint that the inspection sensors influence is the maximum speed at which the UAV can navigate. This

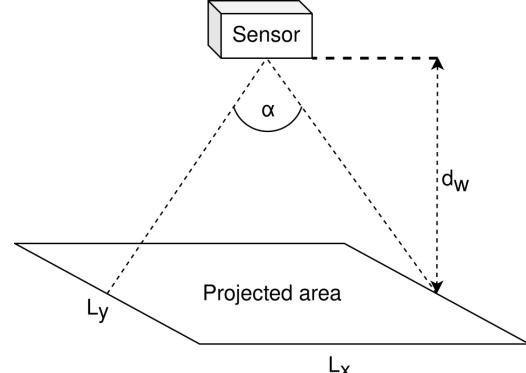


FIGURE 2. Projected area ($L_x \times L_y$) covered by a sensor with α FoV at a given working distance d_w .

limitation is related to the sampling frequency (f_S) of the sensor, given that the UAV cannot move faster than the ability of the sensor to gather the information in each inspection point. Given that, the speed of the UAV must be limited according to (3) [16]:

$$v^{\max} = \min(L_x, L_y) \cdot f_S. \quad (3)$$

Each inspection point has a correspondent viewpoint that consists of the pose where the UAV must be located and oriented to collect the required information of that inspection point. Each viewpoint is obtained according to the working distance (d_w) and to the orientation of the normal (\vec{n} , with only the yaw component) relative to each inspection point.

Fig. 3 exhibits the method used to calculate each viewpoint. The total inspection path, defined by the complete set of

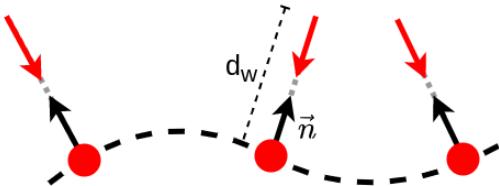


FIGURE 3. Viewpoints calculation: the dashed curve indicates a surface, the red dots are the inspection points, the black arrows show the direction of the normals of each inspection point, and the red arrows are the viewpoints that are d_w away from the inspected surface.

viewpoints, is obtained by applying this method to the whole 3D surface of the inspected structure.

B. UAV ENERGY MODEL

The energy model relates the energy loss of each movement of the UAV, which can be calculated according to (4):

$$E_{total} = E_d + E_a, \quad (4)$$

where the total energy loss (E_{total}) corresponds to the sum of the following components:

- E_d - displacement cost:
 - $E_{d_{xy}}$ - linear displacement on the xOy plane;
 - E_{d_z} - linear displacement on the z axis;
 - $E_{d_{yaw}}$ - angular displacement on the z axis.
- E_a - acceleration cost:
 - $E_{a_{xy}}$ - linear acceleration on the xOy plane;
 - E_{a_z} - linear acceleration on the z axis;
 - $E_{a_{yaw}}$ - angular acceleration on the z axis.

Regarding the displacement cost, the linear movement according to the xOy plane and z axis was analysed, as well as the angular motion along the z axis, that corresponds to *yaw* in Euler angles. The change of direction of the UAV is related with its acceleration and deceleration, and corresponds to another significant factor of energy consumption. The acceleration costs that were analysed depend on the linear acceleration on both xOy plane and z axis, and also on angular acceleration on the z axis (*yaw*). This cost model enables to objectively quantify the cost of moving from a current pose to another pose. Note that, for calculation purposes, the speed of each movement of the UAV is considered constant, and that each acceleration/deceleration is considered instantaneous.

1) Displacement cost

The displacement cost is related with linear and angular movements performed by the UAV, from a given pose p_i to another pose p_{i+1} . Fig. 4 exhibits the displacement diagram of all considered movements that a UAV can perform. Δd_{xy} corresponds to the displacement on the xOy plane, Δd_{z+} and Δd_{z-} corresponds to the ascending and descending displacement on the z axis, respectively, and $\Delta\theta_{yaw}$ corresponds to the rotation on the z axis. The relation of these movements

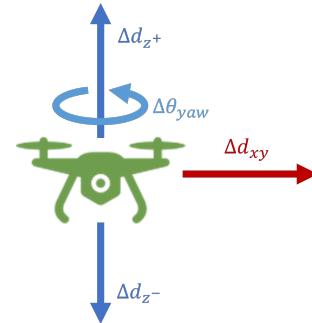


FIGURE 4. Displacement diagram: Δd_{xy} corresponds to the displacement on the xOy plane, Δd_{z+} and Δd_{z-} corresponds to the ascending and descending displacement on the z axis, respectively, and $\Delta\theta_{yaw}$ corresponds to the rotation on the z axis.

with its displacement cost (E_d) can be expressed according to (5):

$$E_d = E_{d_{xy}} + E_{d_z} + E_{d_{yaw}}, \quad (5)$$

where $E_{d_{xy}}$ corresponds to the linear displacement on the xOy plane, E_{d_z} to the linear displacement on the z axis, and $E_{d_{yaw}}$ to the angular displacement on the z axis.

The energy cost of each linear displacement component can be calculated according to (6) [16]:

$$E_{d_\tau} = P_{d_\tau}(v_\tau) \frac{\Delta d_\tau}{v_\tau}, \quad \tau \in [xy, z^+, z^-]. \quad (6)$$

The energy loss caused by the movement according to the xOy plane ($E_{d_{xy}}$) is related to its linear velocity (v_{xy}), its displacement (Δd_{xy}), and the power consumption as a function of the linear velocity ($P_{d_{xy}}(v_{xy})$).

The energy loss caused by the movement according to the z axis (E_{d_z}) depends on its direction, resulting in a higher energy cost when the UAV is ascending ($E_{d_{z+}}$) than when it is descending ($E_{d_{z-}}$). This cost is related to the ascending/descending velocity (v_{z+} and v_{z-}), its displacement, (Δd_{z+} and Δd_{z-}), and the power consumption as a function of the linear velocity ($P_{d_{z+}}(v_{z+})$ and $P_{d_{z-}}(v_{z-})$). Note that the total energy cost of the movement along the z axis (E_{d_z}) corresponds to (7):

$$E_{d_z} = E_{d_{z+}} + E_{d_{z-}}. \quad (7)$$

The energy cost of the angular displacement component can be calculated according to (8) [16]:

$$E_{d_{yaw}} = P_{d_{yaw}}(\omega_{yaw}) \frac{\Delta\theta_{yaw}}{\omega_{yaw}}. \quad (8)$$

The energy loss caused by the rotation according to the z axis ($E_{d_{yaw}}$) is related to its angular velocity (ω_{yaw}), its displacement ($\Delta\theta_{yaw}$), and the power consumption as a function of the angular velocity ($P_{d_{yaw}}(\omega_{yaw})$).

2) Acceleration Cost

The acceleration cost is related with the change of direction of the UAV during its path. Given a current pose p_i , a previous pose p_{i-1} , and the next pose p_{i+1} , it is possible to detect a change of direction and the correspondent acceleration or deceleration by calculating the distance between the displacement vectors $\|\overrightarrow{p_i - p_{i-1}}\|$ and $\|\overrightarrow{p_{i+1} - p_i}\|$. Fig. 5 exhibits an acceleration diagram for the considered changes of direction, and the corresponding accelerations and decelerations. v_l and a_l correspond to the velocity and acceleration components on axis $l = \{x, y\}$, respectively.

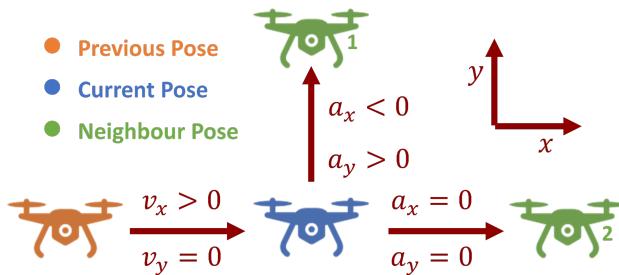


FIGURE 5. Acceleration diagram: v_l and a_l correspond to the velocity and acceleration components on axis $l = \{x, y\}$, respectively.

components on axis $l = \{x, y\}$, respectively. Considering that the UAV has a given speed $v_x > 0$ and $v_y = 0$ at the current pose, moving to neighbour pose 2 means keeping its current movement, not being necessary to accelerate or decelerate in the direction of any axis ($a_x = a_y = 0$), which does not imply any acceleration cost. On the other hand, moving to neighbour pose 1 implies a change of direction, meaning that the UAV has to decelerate in the x axis ($a_x < 0$) and accelerate in the y axis ($a_y > 0$), which causes an energy loss. The relation of these movements with its acceleration cost (E_a) can be expressed according to (9):

$$E_a = E_{a_{xy}} \cos(\phi)^2 + E_{a_z} \sin(\phi)^2 + E_{a_{yaw}}, \quad (9)$$

where $E_{a_{xy}}$ corresponds to the cost of the linear acceleration on the xOy plane, E_{a_z} to the cost of the linear acceleration on the z axis, and $E_{a_{yaw}}$ to the cost of the angular acceleration on the z axis. ϕ represents the difference on the polar angle between two consecutive displacement vectors. Its value allows to detect change of direction between the xOy plane and the z axis. As $\cos(\phi)^2 + \sin(\phi)^2 = 1$, (9) represents the relation between the acceleration along both directions.

The energy cost of each linear acceleration component (E_{a_τ}) can be calculated according to (10) [16]:

$$E_{a_\tau} = K_\tau \cdot P_{a_\tau}(v_\tau) \cdot \Delta t_{a_\tau}, \quad \tau \in [xy, z^+, z^-], \quad (10)$$

where K_τ is a multiplication factor that depends on the change of direction, $P_{a_\tau}(v_\tau)$ is the power consumption as a function of the velocity, and Δt_{a_τ} is the time that it takes to accelerate/decelerate until the desired velocity is reached.

When the UAV is moving in the xOy plane, it is possible to detect changes of direction by calculating the difference

of the direction between two consecutive displacement vectors. $|\theta| \in [0, \pi]$ represents that change on the direction of movement in the xOy plane. The energy loss caused by the linear acceleration/deceleration according to the xOy plane ($E_{a_{xy}}$) is related to its power consumption as a function of the linear velocity ($P_{a_{xy}}(v_{xy})$) and to the time that it takes to accelerate/decelerate until the desired velocity is reached ($\Delta t_{a_{xy}}$). $K_{xy} \in [0, 1]$ is a multiplication factor that assumes the values presented in (11):

$$K_{xy} = \begin{cases} 0, & \text{if } \theta = 0; \\ 1, & \text{if } \frac{\pi}{2} \leq |\theta| \leq \pi; \\ \left(1 - \sqrt{1 - \left(\frac{\theta}{\pi/2}\right)^2}\right), & \text{if } 0 < |\theta| < \frac{\pi}{2}. \end{cases} \quad (11)$$

Figure 6 helps to explain the values that K_{xy} takes, where $\theta = 0$ is represented with the dark red arrow and corresponds to no change of direction, $0 < |\theta| < \frac{\pi}{2}$ is represented in light blue and corresponds to a smooth change of direction, and $\frac{\pi}{2} \leq |\theta| \leq \pi$ is represented in purple and corresponds to a sharp change of direction. When $\theta = 0$ there is no accel-

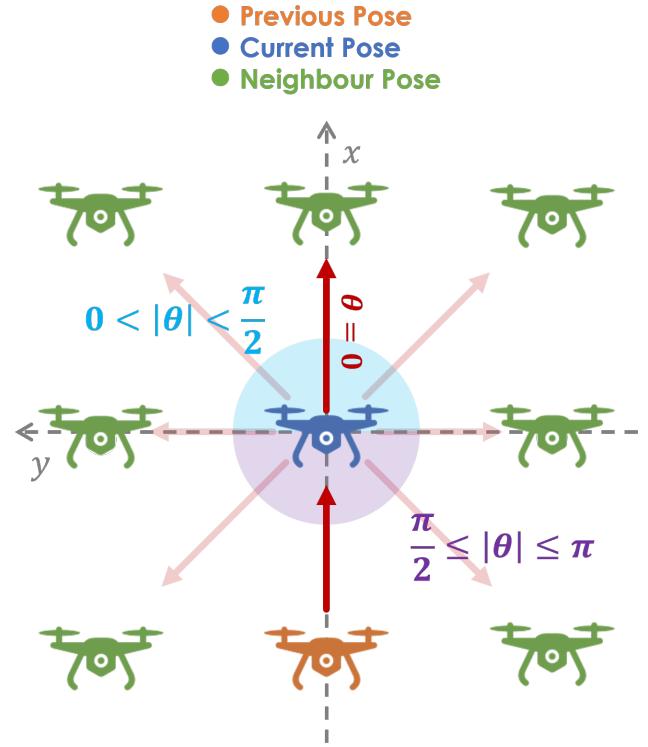


FIGURE 6. Different directions of movement in the xOy plane, according to θ . $\theta = 0$ is represented with the dark red arrow and corresponds to no change of direction, $0 < |\theta| < \frac{\pi}{2}$ is represented in light blue and corresponds to a smooth change of direction, while $\frac{\pi}{2} \leq |\theta| \leq \pi$ is represented in purple and corresponds to a sharp change of direction.

eration in the xOy plane in the UAV movement ($a_{xy} = 0$), so K_{xy} takes value 0. When $\frac{\pi}{2} \leq |\theta| \leq \pi$ the UAV has to decelerate and then accelerate ($a_{xy} \neq 0$), as it has to completely change its direction of movement, resulting in K_{xy} equal to 1. When $0 < |\theta| < \frac{\pi}{2}$, the UAV's acceleration

is an increasing function of θ ($a_{xy} \neq 0$). K_{xy} changes according to the ellipsoidal function presented in (12), that depends on θ :

$$\left(\frac{\theta}{\pi/2}\right)^2 + (K_{xy} - 1)^2 = 1. \quad (12)$$

The acceleration/deceleration cost according to the z axis (E_{az}) depends if the UAV is ascending ($E_{a_{z+}}$) or descending ($E_{a_{z-}}$). It is possible to detect changes of direction by calculating the difference of the direction between two consecutive displacement vectors. $|\phi| \in [0, \pi]$ represents that change on the direction of movement in the z axis. The acceleration cost is related to its power consumption as a function of the linear velocity ($P_{a_{z+}}(v_{z+})$ and $P_{a_{z-}}(v_{z-})$) and to the time that it takes to accelerate/decelerate until the desired velocity is reached ($\Delta t_{a_{z+}}$ and $\Delta t_{a_{z-}}$). $K_\zeta \in [0, 1]$, $\zeta = \{z^+, z^-\}$, is a multiplication factor that assumes the values presented in (13):

$$K_\zeta = \begin{cases} 0, & \text{if } \phi = 0; \\ 1, & \text{if } \phi \neq 0. \end{cases} \quad (13)$$

When $\phi = 0$ there is no acceleration in the z axis in the UAV movement ($a_z = 0$), so K_ζ takes value 0. When $\phi \neq 0$ the UAV has to accelerate/decelerate on the z axis ($a_z \neq 0$), resulting in K_ζ equal to 1. Note that the total energy cost given by the acceleration along the z axis (E_{az}) corresponds to (14):

$$E_{az} = E_{a_{z+}} + E_{a_{z-}}. \quad (14)$$

The energy cost of the angular acceleration component can be calculated according to (15) [16]:

$$E_{ayaw} = K_{yaw} \cdot P_{a_{yaw}}(\omega_{yaw}) \cdot \Delta t_{a_{yaw}}. \quad (15)$$

It is possible to detect changes of angular directions by calculating the difference of the angular direction between two consecutive displacement vectors. $|\psi| \in [0, \pi]$ represents that change on the angular direction of movement in the z axis. The energy loss caused by the angular acceleration according to the z axis ($E_{a_{yaw}}$) is related to its power consumption as a function of the angular velocity ($P_{a_{yaw}}(\omega_{yaw})$) and to the time that it takes to accelerate/decelerate until the desired angular velocity is reached ($\Delta t_{a_{yaw}}$). $K_{yaw} \in [0, 1]$, is a multiplication factor that assumes the values presented in (16):

$$K_{yaw} = \begin{cases} 0, & \text{if } \psi = 0; \\ 1, & \text{if } \psi \neq 0. \end{cases} \quad (16)$$

When $\psi = 0$ there is no angular acceleration in the UAV movement according to the z axis ($a_{yaw} = 0$), so K_{yaw} takes value 0. When $\psi \neq 0$ the UAV has to accelerate/decelerate on the z axis ($a_{yaw} \neq 0$), resulting in K_{yaw} equal to 1.

C. INSPECTION PATH PLANNING ALGORITHM

Given a set of viewpoints and the UAV energy model that relates the energy loss of moving from a viewpoint to another, it is possible to develop an algorithm that generates a path for the UAV to inspect a structure, optimizing its energy

loss. This problem was formulated as a new variant of the Travelling Salesman Problem that is defined as ATSP-PL (Asymmetric Travelling Salesman Problem, with Precedence Loss). The Asymmetric classification is due to the fact that the cost of moving from viewpoint p_i to viewpoint p_{i+1} is not necessarily the same of moving from viewpoint p_{i+1} to viewpoint p_i . The Precedence Loss designation is related with the fact that the cost of moving from the current viewpoint (p_i) to the next viewpoint (p_{i+1}) depends on the previous viewpoint (p_{i-1}), given the significant difference that the acceleration cost implies in the calculation of the total energy loss.

The algorithm developed in this research, named EEPPA (Energy Efficient Path Planning Algorithm), proposes a method to solve this novel TSP formulation (the ATSP-PL), and is divided into two main procedures:

- Spiral Procedure - a greedy and time efficient method, specially suited to cover regular shaped structures, such as cylinders or cubes;
- ATSP Solver followed by a Local Search Procedure - the original ATSP-PL problem is relaxed and solved as a traditional ATSP, followed by a Local Search Procedure that adapts the obtained solution to take into account the Precedence Loss property.

1) ATSP-PL Formulation

As in the classical TSP, the solution of the ATSP-PL consists of a sequence of viewpoints organized by its order of inspection. The first viewpoint is the only viewpoint whose position in the sequence is previously known, as it is user defined. Each solution sequence has an associated total energy cost and total linear displacement. While performing the path, the UAV has to meet all viewpoints exactly once, with the objective of minimizing the energy loss during the execution of the complete path. With this information it is possible to modulate this problem as follows [24]:

$$x_{ij} = \begin{cases} 1, & \text{if the path goes from viewpoint } i \text{ to } j, \\ 0, & \text{otherwise,} \end{cases} \quad (17)$$

$$\sum_{\substack{i=1, \\ i \neq j}}^N x_{ij} = 1, \forall j = 1, \dots, N, \quad (18)$$

$$\sum_{\substack{j=1, \\ j \neq i}}^N x_{ij} = 1, \forall i = 1, \dots, N, \quad (19)$$

$$\sum_{i \in Q} \sum_{\substack{j \neq i, \\ j \in Q}} x_{ij} \leq |Q| - 1, \forall Q \subseteq \{1, \dots, N\}, |Q| \geq 2, \quad (20)$$

where N is the total number of viewpoints. The constraint presented in (20) ensures that no sub-sequences (Q) are formed, resulting in a single sequence of viewpoints [24]. The objective function is the following:

$$\min \sum_{k=1}^N \sum_{i=1}^N \sum_{\substack{j=1, \\ i \neq k, j \neq k, \\ j \neq i}}^N x_{ij} \cdot c_{kij}, \quad (21)$$

where c_{kij} is the energy cost of moving from viewpoint i to j , given a previous viewpoint k . Note that the constraints presented in (17) to (20) are equivalent to the standard ATSP formulation, while only the objective function, presented in (21), has been altered to fit the ATSP-PL formulation. This new formulation minimizes the cost of moving from the current viewpoint to the next viewpoint, given a previous viewpoint, justifying the Precedence Loss designation.

2) EEPPA - Energy Efficient Path Planning Algorithm

The developed EEPPA algorithm, whose flowchart is presented in Fig. 7, takes as input the map of viewpoints of the inspected structure and the energy model of the UAV, and outputs the inspection path as a sequence of viewpoints. The

performed in the first stage of the algorithm (represented with the blue diamond in Fig. 7) recurring to model fitting and segmentation algorithms [25], capable of detecting objects with regular shapes, such as cylinders or cubes, allowing to divide the complete structure into regular shaped and complex shaped sub-structures.

The path planning for the regular shaped sub-structures is generated with the Spiral Procedure method (represented in blue on the left side of Fig. 7), due to its ability to achieve a feasible solution in a short computation time. This greedy method is led by a simple dispatching rule with a considerable cost-benefit ratio between the quality of the solution and the time spent to achieve it. The flowchart of Spiral Procedure method is presented in Fig. 8. The Spiral

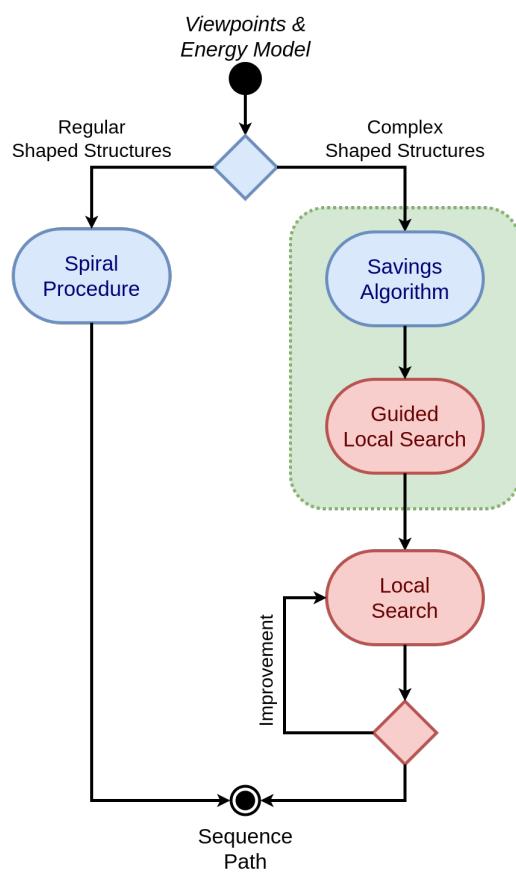


FIGURE 7. Flowchart of the EEPPA algorithm: constructive heuristics represented in blue, improvement heuristics represented in red, and ATSP Solver heuristic represented in green.

map of viewpoints comprises all inspection poses that the UAV must meet, while the energy model enables to calculate the energy consumption of moving from one pose to the following one, which not only iteratively assists on selecting the subsequent viewpoint, but also allows to determine the total energy loss of the path produced. The inspected structure can be divided into sub-structures to allow the application of different methods that may fit better depending on the shape of the sub-structure to be inspected. This division is

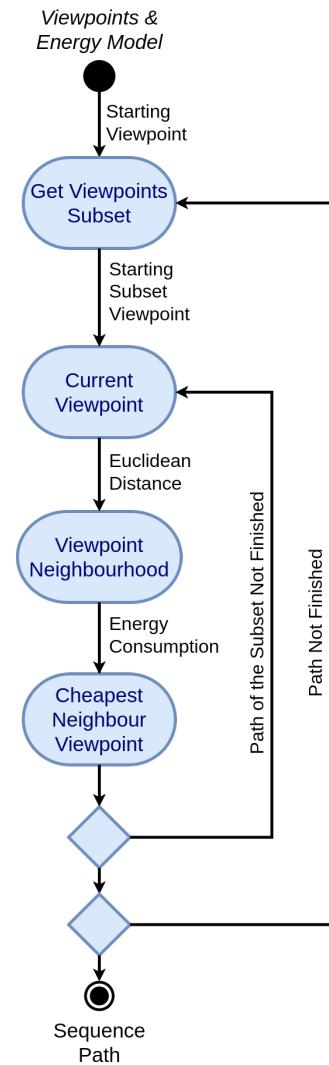


FIGURE 8. Flowchart of the Spiral Procedure method.

Procedure method is similar to a cheapest neighbour approach. A solution is generated by iteratively selecting the least expensive neighbour viewpoint to move to, according to the energy loss. Each viewpoint neighbourhood is defined by the euclidean distance within a subset of viewpoints with

similar height. When all the viewpoints of that subset are selected, the last viewpoint's neighbourhood is expanded to the next subset of viewpoints, and this process is repeated until all viewpoints have been covered.

A different approach is used for the inspection of a sub-structure with a complex shape. By relaxing the Precedence Loss property of the ATSP-PL, it is possible to formulate and solve this problem as a traditional ATSP (represented with the green background on the right side of Fig. 7). As the ATSP is a well-known problem, using one of the several optimized open source solvers is the most suitable option. The selected open source ATSP Solver was the OR-Tools from Google Developers [26], in particular the Routing algorithm. This solver allows to select the greedy algorithm that generates the initial solution, and the metaheuristic algorithm capable of improving it. Its input is a cost matrix with a size of $N \times N$ (N represents the number of viewpoints in the inspection map), in which each cell has the energy cost of moving from one viewpoint to another. For the purpose of this problem, the Savings algorithm [27] was selected for the initial solution algorithm, and the Guided Local Search for the metaheuristic. This metaheuristic is generally the most efficient metaheuristic for routing problems [26], as it enables the solver to escape from a local minimum, proceeding with the search for better solutions. To adapt and improve the solution obtained with the ATSP Solver for the ATSP-PL formulation, a Local Search method is applied (represented in red on the right side of Fig. 7). The flowchart of this method is presented in Fig. 9. The local search is a well-known approach that is able to enhance a solution by searching for an improving one within its neighbourhood. In the developed Local Search method, two different approaches were followed. One is based on the Relocate algorithm [28], and consists of swapping pairs of viewpoints of a sequence within a defined neighbourhood (represented on the left side of Fig. 7). The other is based on the 2-Opt algorithm [29], [30], and consists in flipping the order of a sub-sequence between two viewpoints within a defined neighbourhood (represented on the right side of Fig. 7). Fig. 10 exhibits an example of a flipped sub-sequence (3, 6). This Local Search method will be continuously seeking for an enhanced solution, as long as each of its approaches is able to find an improving solution at each iteration.

IV. SIMULATION SCENARIO RESULTS

The EEPPA algorithm was applied in the *Gazebo* simulator, which allows to replicate real environment conditions and to represent the UAV and its sensors with realistic models, as well as a structure to be inspected. This section displays the generated map of viewpoints and the obtained path of inspection, presenting the correspondent energy consumption, total displacement and time spent during the inspection. By processing the data collected during the execution of the inspection path, it is possible to reconstruct the structure's model and to map the confidence on the quality of the data at each location. The results obtained in the conducted simulation were performed on a computer with the following

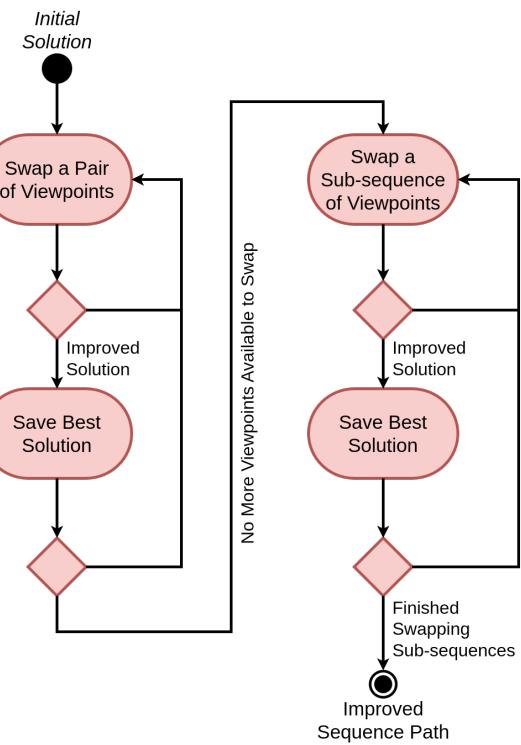


FIGURE 9. Flowchart of the Local Search method, which includes the swap of pairs of viewpoints (on the left side), and the swap of sub-sequences of viewpoints (on the right side).

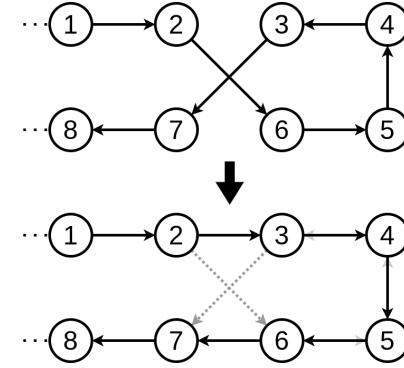


FIGURE 10. Example of a flipped sub-sequence (3, 6).

specifications:

- CPU - Intel® Core™ i7 7700HQ Quad-Core, 2.80 GHz;
- GPU - Nvidia GeForce GTX 1050 4GB GDDR5;
- RAM - 16GB DDR4 SDRAM.

The UAV model used in the simulator corresponds to the IRIS quadrotor controlled by an *ArduPilot* autopilot, powered by a 3S LiPo battery (5.5 Ah) [16]. For the inspection map definition, the considered sensors were a monocular camera and a 3D LiDAR, with the following characteristics:

- **Monocular Camera** - *Mako G-125* - Frame Rate: 30 Hz, Resolution: 1292 x 964 pixels, Field of View: 80° horizontal;

- **LiDAR - VLP-16** - Frame Rate: 10 Hz, Resolution: 1875 x 16 channels, Range: 100 m, Accuracy: 0,03 m, Field of View: 360° horizontal and 30° vertical.

Given these sensors' characteristics it is possible to establish the values of each inspection parameter, which are presented in table 1.

TABLE 1. Defined values for the inspection parameters.

Parameter	Variable	Value
Working Distance	d_w	5 m
Linear Velocity (x, y)	v_{xy}	5 m/s
Ascending Velocity (z)	v_{z+}	2.1 m/s
Descending Velocity (z)	v_{z-}	1.3 m/s
Angular Velocity (yaw)	ω_{yaw}	2.1 rad/s
Discretization Rate	D_r	2 m

For the purpose of this demonstration, the selected structure to be inspected consists of an offshore wind turbine, whose model is depicted in Fig. 11. This model consists

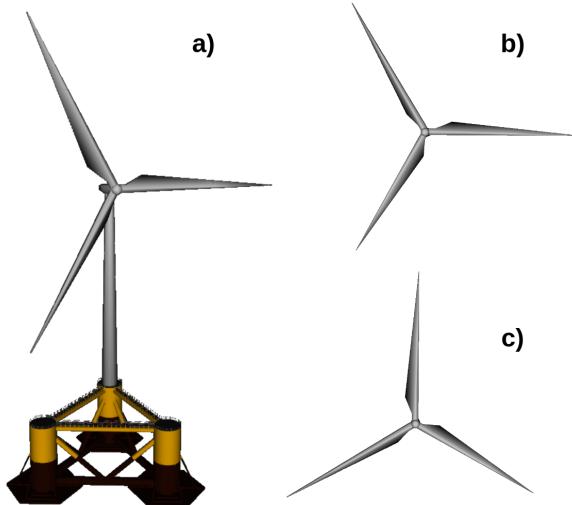


FIGURE 11. Wind turbine model: a) Windfloat, b) Horizontal blade configuration, c) Vertical blade configuration.

of the WindFloat design [31], that has a column height of 95 m and a blade length of 82 m. Different tests were performed with two different blades configurations, shown in Fig. 11b) and Fig. 11c). The top figure corresponds to the horizontal configuration, and the bottom one to the vertical configuration. Given the previously defined parameters and this wind turbine model, it is possible to generate the map of viewpoints of the structure's surface, for the UAV inspection. Fig. 12 exhibits the sequence of operations to obtain the map of viewpoints for the horizontal configuration. Fig. 12a) shows a section of the wind turbine model, while Fig. 12b) depicts the correspondent inspection points of its surface (colored dots), with a discretization rate (D_r) of 2 m. Fig. 12c) shows the viewpoints obtained (red arrows) for each inspection point, for a working distance (d_w) of 5 m. Fig. 12d) depicts the same viewpoints map and the correspondent wind turbine model. The number of viewpoints generated

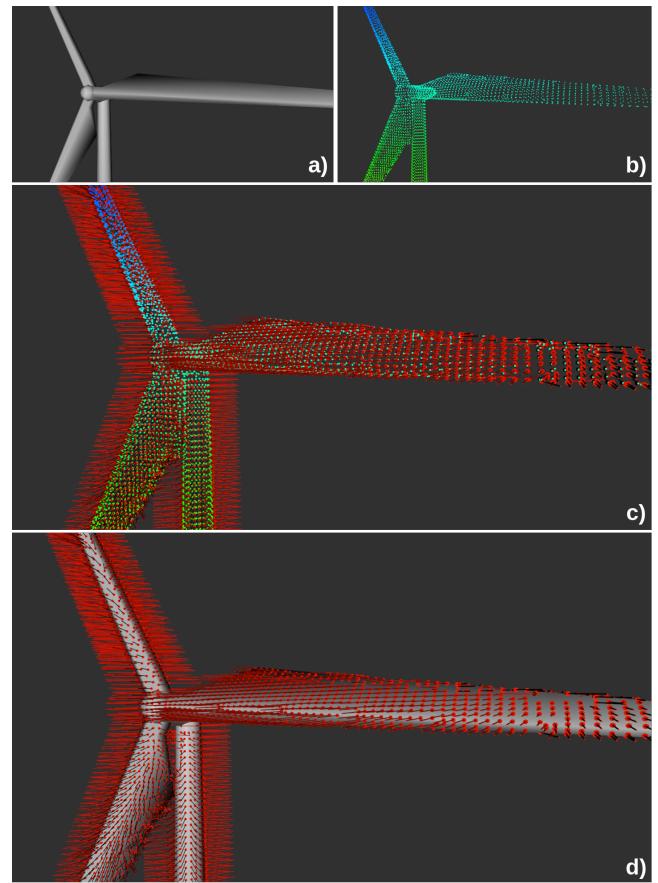


FIGURE 12. Sequence of operations to obtain the map of viewpoints: a) wind turbine model, b) discretized inspection surface ($D_r = 2$ m), represented by the colored dots, c) the correspondent viewpoint of each inspection point ($d_w = 5$ m), represented by the red arrows, d) the wind turbine model and the correspondent viewpoints map, represented by red arrows.

to inspect this wind turbine model is close to 1300. This procedure is analogous for both vertical and horizontal blade configuration, and for any other structure to be inspected.

A. PATH FOR THE INSPECTION OF THE STRUCTURE

Given the map of viewpoints of the structure and the energy model of the UAV, it is possible to apply the developed path planning algorithm. The results obtained by the EEPPA algorithm are compared with the results obtained with two distinct algorithms, OR-Tools and Cheapest Neighbour (CN). The OR-Tools corresponds to the previously mentioned ATSP solver [26], that obtains a solution based on the relaxation of the original problem (does not consider the acceleration cost of the UAV, that corresponds to the Precedence Loss characteristic of the ATSP-PL problem formulation). The Cheapest Neighbour algorithm is a greedy approach that generates a solution by iteratively selecting the least expensive neighbour viewpoint to move to, according to the energy loss. Each viewpoint neighbourhood is defined by the euclidean distance. Fig. 13 depicts the paths obtained for both wind turbine configurations, with the EEPPA, CN and OR-Tools algorithms. The paths generated are represented

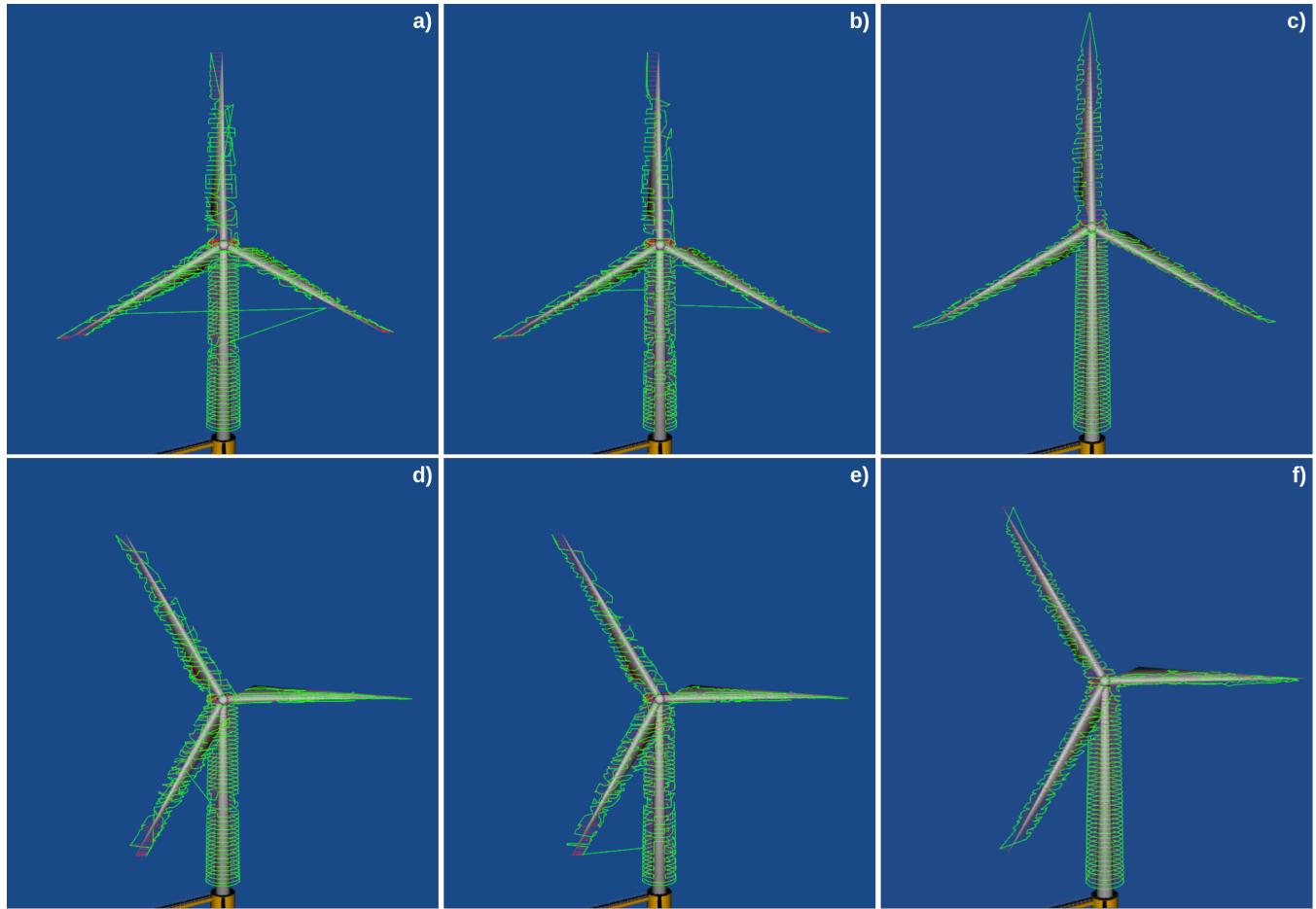


FIGURE 13. Paths obtained for the inspection of the wind turbine. Paths are represented by the green line and each viewpoint is represented by a red arrow: a) Vertical configuration - CN, b) Vertical configuration - OR-Tools, c) Vertical configuration - EEPPA, d) Horizontal configuration - CN, e) Horizontal configuration - OR-Tools, f) e) Horizontal configuration - EEPPA.

by the green line and each viewpoint is represented by a red arrow. By analysing this figure it is noticeable that the paths generated by EEPPA are more sequenced and with smoother movements, while the paths generated by CN and OR-Tools present rougher movements and a less sequential path. The inspection of both wind turbine configurations was performed by the UAV in the simulation environment, where the paths generated by EEPPA algorithm were followed¹. Tables 2 and 3 present a summary of the values obtained in the simulation scenario for the paths with the vertical and horizontal configurations, respectively, regarding energy loss, displacement and time spent during the inspection.

For the vertical configuration, the EEPPA algorithm resulted in a improvement of 27.4% regarding the CN, and of 26.6% regarding the OR-Tools. For the horizontal con-

¹Videos of both inspection paths available at <https://youtu.be/ZK6-UtJwNns> (horizontal configuration) and <https://youtu.be/on-gqbnurrM> (vertical configuration), displayed with ten times of the real speed. The wind turbine model is represented in grey and the UAV in yellow. On the left it is visible the images captures by the camera. The point cloud obtained by the LiDAR are represented by the white dots. The red arrows represent the pose (position and orientation) of the UAV at each instance of the inspection.

TABLE 2. Summary of the values obtained in the simulation scenario for the path of the **vertical** configuration of the wind turbine, regarding energy loss, displacement and time spent during the inspection.

	Energy Loss (Wh)	Displacement (m)	Time Spent (min)
CN ^a	128.29	4712	37
OR-Tools ^b	126.93	3999	39
EEPPA ^c	93.17	3930	32

^aCheapest Neighbour Algorithm;

^bOR-Tools ATSP Solver Algorithm [26];

^cEnergy Efficient Path Planning Algorithm.

figuration, the improvement regarding the CN and the OR-Tools are of 28.3% and 24.8%, respectively. The difference in the energy loss between the paths obtained with the two wind turbine configurations can be explained by the orientation of the blades. In the vertical configuration one of the blades is positioned perpendicularly to the ground, whereas in the horizontal configuration one of the blades is positioned in parallel. The horizontal blade allows the UAV to travel in a straight line more often, not having the necessity to change its direction of movement multiple times, resulting

TABLE 3. Summary of the values obtained in the simulation scenario for the path of the **horizontal** configuration of the wind turbine, regarding energy loss, displacement and time spent during the inspection.

	Energy Loss (Wh)	Displacement (m)	Time Spent (min)
CN ^a	125.85	4677	36
OR-Tools ^b	119.92	3907	39
EEPPA ^c	90.24	3848	32

^aCheapest Neighbour Algorithm;

^bOR-Tools ATSP Solver Algorithm [26];

^cEnergy Efficient Path Planning Algorithm.

in a smoother path. The vertical blade forces the UAV to successively accelerate and decelerate during its inspection, increasing its energy loss. The paths generated by EEPPA not only allowed a higher energy efficiency, which is the mains goal, but also to achieve faster and shorter paths. This can be explained by the fact that the EEPPA generates smoother and more sequenced paths, avoiding longer distances between two consecutive viewpoints.

Fig. 14 shows an example of the improvement generated by the Local Search method in the EEPPA algorithm. Fig. 14a) shows a segment of the path generated by the OR-Tools ATSP Solver, that is optimized to only reduce the displacement cost. Fig. 14b) shows the path obtained for the same path segment after applying the Local Search method, that also takes into account the acceleration cost. Although this change increases the total displacement, the *loop* generated by the Local Search decreases the number of changes in direction, which smooths the movement of the UAV and decreases the energy loss that this segment of the path would produce.

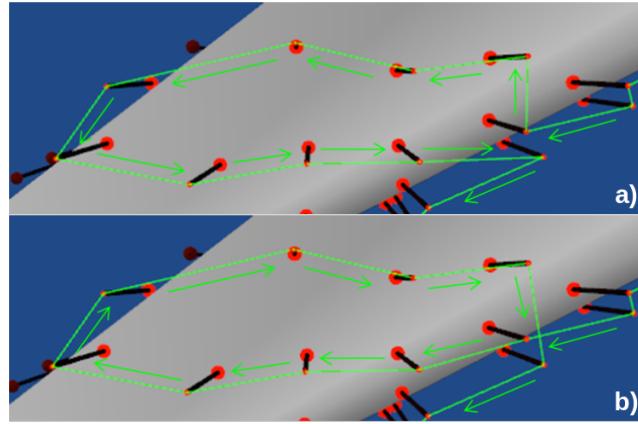


FIGURE 14. Improvement produced by Local Search method: a) before Local Search, b) after Local Search. The path and its direction is represented by the green line and arrows, respectively. Each viewpoint is represented by a red arrow.

B. 3D MODEL RECONSTRUCTION

During the inspection of the wind turbine in the simulation scenario, the UAV was equipped with a LiDAR sensor which is able to collect a 3D point cloud of the surrounding

environment. With a point cloud registration method, it is possible to concatenate successive point clouds collected at each instance, reconstructing the entire structure of inspection, which results in a complete representation of the wind turbine surface. The used method consists in pre-aligning each point cloud based on the relative position between the UAV and the turbine (given by a RTK GNSS sensor), and then fine tuning the alignment with the ICP algorithm [32]–[35]. This 3D reconstruction allows to identify deviations on the blades or tower, or even detect if any part is missing or damaged. Fig. 15 depicts the point cloud of the reconstructed model of the inspected wind turbine, for the vertical blade configuration². A confidence parameter is assigned to each

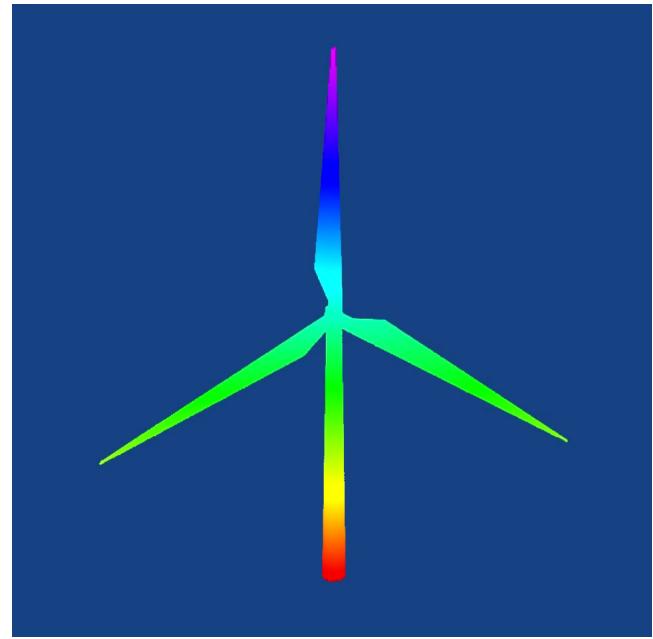


FIGURE 15. Obtained point cloud of the reconstructed wind turbine model (vertical configuration).

inspection point, allowing to create a confidence map of the inspection data. This confidence parameter is related with the fitting ratio returned by the ICP algorithm, that depends on the pre-alignment given by the relative position between the UAV and the wind turbine obtained by the RTK sensor. This confidence map identifies the certainty on the location of the collected data for the reconstructed model, mapping the areas with less reliable information that may need to be double checked. Figure 16 displays the confidence map of the inspection of the wind turbine for the vertical configuration. The green dots represent the poses with the highest confidence in the quality of the collected data, whereas red dots represent the poses with the lowest confidence in the data that was collected.

²Video of the iterative model reconstruction of the wind turbine model (vertical configuration) available at <https://youtu.be/xHD4EB6HuDc>

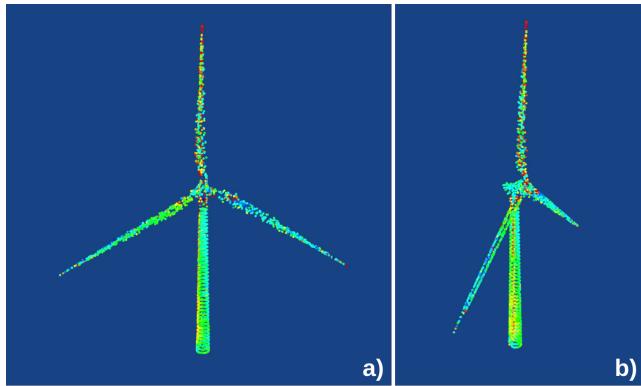


FIGURE 16. Confidence map of the model reconstruction: a) Front view, b) Side view. Green dots represent the poses with the highest confidence in the quality of the collected data, whereas red dots represent the poses with the lowest confidence in the collected data.

V. REAL SCENARIO RESULTS

The developed algorithm for inspection path planning was also tested in a real scenario. The used aerial robot is INESC TEC's CROW (Copter Robot for Offshore Wind-farms) UAV. This UAV is based on a quadcopter frame (with a wingspan of 0.7 m, and a maximum payload of 2 Kg) with a PixHawk controller running the ArduPilot software. Its navigation system is composed of a GNSS/RTK sensor, that has a positioning accuracy of 2.5 m and of 0.025 m when paired with a base station. It has an onboard processing unit (Raspberry Pi 4B 8Gb). CROW is supplied by a 6S LiPo battery of 14 Ah, which enables a total flight time of at least 25 minutes (depending on the payload). The values defined for the linear and angular speeds of the UAV during the inspection are the same used in the simulation experiments. For the inspection map definition, the considered sensors were a monocular camera and a 3D LiDAR, with the following characteristics:

- **Monocular Camera - TIS-DFM-37UX273-ML** Frame Rate: 5 Hz, Resolution: 1440×1080 pixels, Field of View: 90° horizontal;
- **LiDAR - OS1-64** - Frame Rate: 10 Hz, Resolution: 1024×64 channels, Range: 120 m, Accuracy: 0,05 m, Field of View: 360° horizontal and 45° vertical.

These sensors allow the UAV to gather high quality information about the inspected structure. Figure 17 depicts CROW with the described sensors attached.

The structure inspected in this experiment was a 25 m high illumination tower of the football field of the Faculty of Sport of the University of Porto (FADEUP), depicted in Fig. 18. Given that a 3D model of the illumination tower was not available, a coarse sketch of its surface was obtained by collecting several point clouds at distinct points of the structure, recurring to the LiDAR mentioned above. To approximate this experience to the problem of inspecting a larger structure (as the wind turbine model used in the simulation experiments) the discretization rate was considerably reduced to increase the number of inspection points, also resulting in the increase of the total number of viewpoints. In particular,



FIGURE 17. CROW UAV and its sensors payload: a monocular camera and a 3D LiDAR.



FIGURE 18. Inspected structure: illumination tower of FADEUP's football field.

the discretization rate (D_r) was set to 0.2 m horizontally and 0.5 m vertically, which resulted in about 800 viewpoints. The discretized inspection surface of the illumination tower is depicted in Fig. Va). The working distance (d_w) was set to 3 m, which allowed to obtain the map of viewpoints exhibited in Fig. Vb). Note that the irregularity and roughness of the obtained viewpoints map is due to the coarse sketch used as model of the illumination tower. Similarly to the simulation scenario, the results obtained by the EEPAA algorithm were compared with the results obtained with the OR-Tools and the CN algorithms. Due to the regular shape of the structure used in these tests, the results obtained when applying only the Spiral Procedure algorithm were also analysed. The path obtained with the EEPAA algorithm for the inspection of the

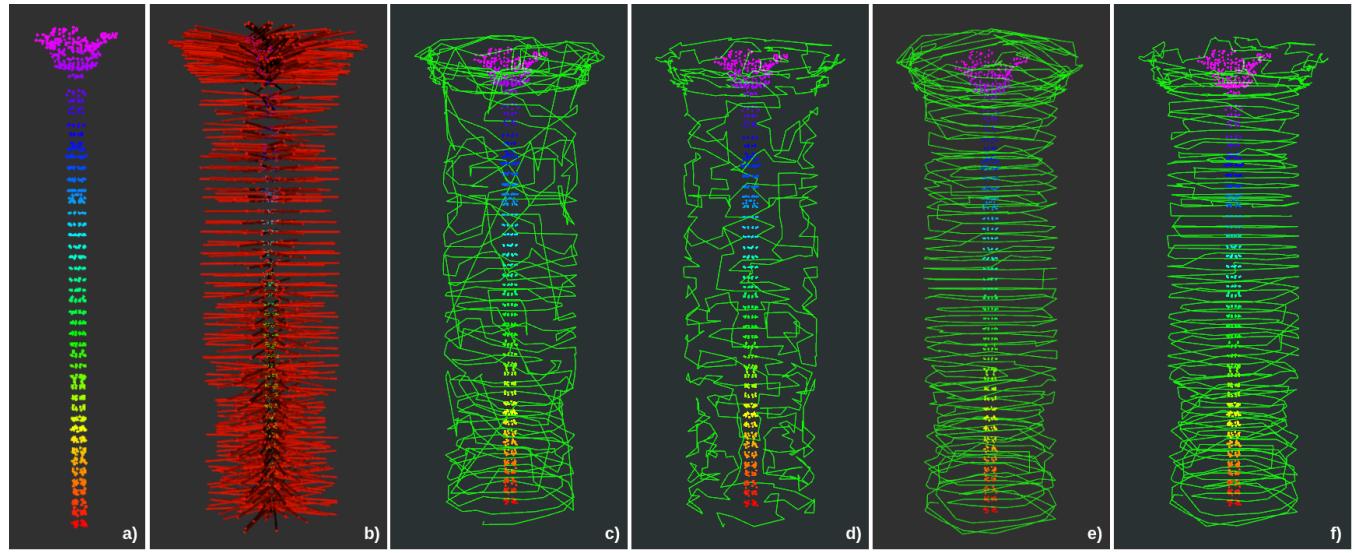


FIGURE 19. Path obtained to inspected the illumination tower: a) Discretized inspection surface ($D_r = 0.2$ m horizontal and 0.5 m vertical), b) Viewpoints map ($d_w = 3$ m), c) CN inspection path, d) OR-Tools inspection path, e) Spiral Procedure inspection path, and f) EEPPA inspection path.

illumination tower is displayed in Fig. Vf), whereas the paths obtained with the CN, OR-Tools and Spiral Procedure are depicted in Fig. Vc), Fig. Vd) and Fig. Ve), respectively. Table 4 presents a summary of the values obtained in this real scenario, for the paths generated for the inspection of the illumination tower, regarding energy loss, displacement and time spent during the inspection.

TABLE 4. Summary of the values obtained in the real scenario for the paths of the illumination tower, regarding energy loss, displacement and time spent during the inspection.

	Energy Loss (Wh)	Displacement (m)	Time Spent (min)
CN ^a	38.34	955	12
OR-Tools ^b	44.17	590	13
SP ^c	36.71	1210	14
EEPPA ^d	35.01	1147	14

^aCheapest Neighbour Algorithm;

^bOR-Tools ATSP Solver Algorithm [26];

^cSpiral Procedure;

^dEnergy Efficient Path Planning Algorithm.

The EEPPA algorithm resulted in an improvement of 20.7%, of 8.7% and of 4.6%, when compared with the OR-Tools, CN and Spiral Procedure algorithms, respectively. Although the total displacement generated by the path of EEPPA is considerable higher than the displacement of CN and OR-Tools, the energy loss is considerably lower. This can be explained once again by the fact that EEPPA is an algorithm focused on minimizing the energy consumption during the inspection, instead of the total displacement. The path generated by EEPPA is smoother and more sequential, preventing the UAV from successively accelerating and decelerating its movement. Due to the regular shape of the illumination tower, the results obtained with EEPPA are similar to the ones obtained with only the Spiral Procedure. The

main explanation for these results is related with the fact that EEPPA algorithm applies the Spiral Procedure to produce the path for the inspection of the column, only differing in the method used to generate the path to inspect the lamps on the top of the column. Nevertheless, even this small difference in the path generated for the lamps resulted in a more energy efficient path.

Fig 20 depicts CROW during an inspection of the illumination tower, and a sample of the collected data. By registering

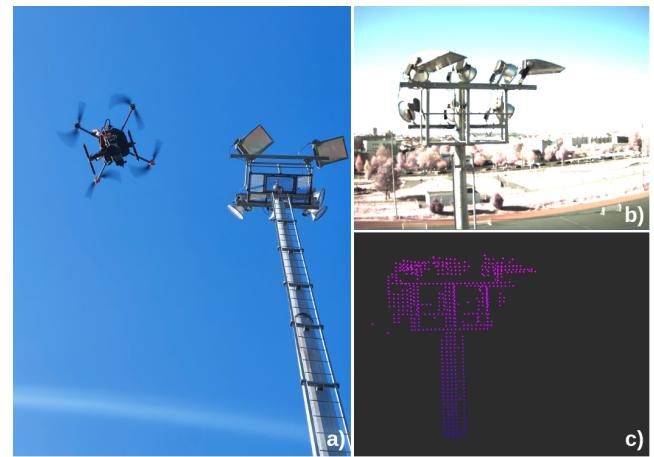


FIGURE 20. UAV inspection: a) CROW inspecting the illumination tower, b) sample of collected data with the visual camera, c) sample of collected data with the 3D LiDAR.

the point cloud collected during the inspection it was possible to reconstruct the illumination tower module that is depicted in Fig. 21.

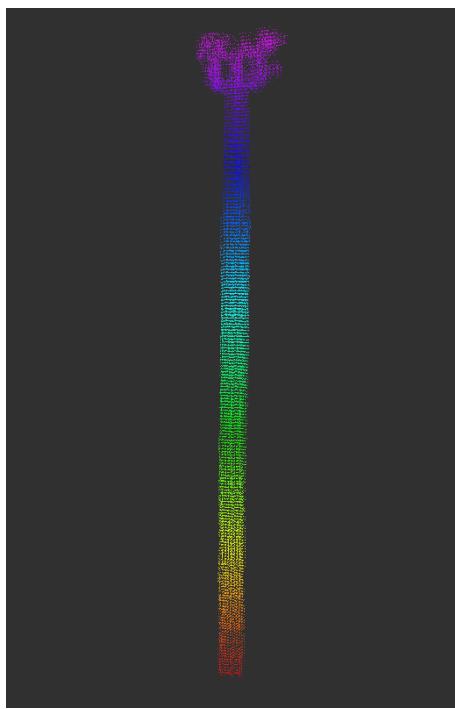


FIGURE 21. Reconstruction of the illumination tower using point clouds.

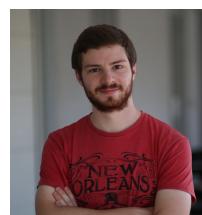
VI. CONCLUSION

This work presented an energy oriented path planning algorithm that minimizes the energy loss of a UAV during the inspection of a structure. The surface of a given model to be inspected was discretized, allowing to obtain an inspection map. This map is composed of a set of viewpoints that depend on the structure surface shape, and also on the sensors' characteristics. To optimize the path to be performed in terms of energy loss, a complete energy model of the UAV was defined. This model considers the energy consumption of distinct components, such as displacement and acceleration, related with the changes of direction of the movement of the UAV. With the list of viewpoints and the energy model defined, it was possible to solve the path planning problem with the developed algorithm, named EEPPA. The experiments performed in both simulation and real scenarios allowed to validate the developed algorithm, which obtained improvements over 24% and 8%, respectively, when compared with other methods. The simulated experiments were performed with realistic models of wind turbines and a UAV, while the real experiments were performed with a real UAV and an illumination tower. The collected inspection data, in particular the point cloud, allowed to perform model reconstructions of the inspected structures, as well as creating a confidence map that relates the confidence on the data collected in each viewpoint with its localization.

REFERENCES

- [1] L. Wang and Z. Zhang, "Automatic detection of wind turbine blade surface cracks based on uav-taken images," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 9, pp. 7293–7303, 2017.
- [2] D. F. Campos, M. Pereira, A. Matos, and A. M. Pinto, "Diius - distributed perception for inspection of aquatic structures," in *OCEANS 2021: San Diego – Porto*, 2021, pp. 1–5.
- [3] A. M. Pinto, J. V. A. Marques, D. F. Campos, N. Abreu, A. Matos, M. Jussi, R. Berglund, J. Halmé, P. Tikka, J. Formiga, C. Verrecchia, S. Langiano, C. Santos, N. Sá, J.-J. Stoker, F. Calderoni, S. Govindaraj, A. But, L. Gale, D. Ribas, N. Hurtós, E. Vidal, P. Ridao, P. Chieslak, N. Palomeras, S. Barberis, and L. Aceto, "Atlantis - the atlantic testing platform for maritime robotics," in *OCEANS 2021: San Diego – Porto*, 2021, pp. 1–5.
- [4] P. Cao, Y. Liu, C. Yang, S. Xie, and K. Xie, "Mec-driven uav-enabled routine inspection scheme in wind farm under wind influence," *IEEE Access*, vol. 7, pp. 179 252–179 265, 2019.
- [5] J. P. Aquilina, R. N. Farrugia, and T. Sant, "On the energy requirements of uavs used for blade inspection in offshore wind farms," in *2019 Offshore Energy and Storage Summit (OSES)*, 2019, pp. 1–7.
- [6] R. Dai, S. Fotedar, M. Radmanesh, and M. Kumar, "Quality-aware uav coverage and path planning in geometrically complex environments," *Ad Hoc Networks*, vol. 73, 02 2018.
- [7] C. Galleguillos, A. Zorrilla, A. Jimenez, L. Diaz, Á. Montiano, M. Barroso, A. Viguria, and F. Lasagni, "Thermographic non-destructive inspection of wind turbine blades using unmanned aerial systems," *Plastics, Rubber and Composites*, vol. 44, pp. 98–103, 04 2015.
- [8] P. Durdevic, D. Ortiz-Arroyo, and Z. Yang, "Lidar assisted camera inspection of wind turbines: Experimental study," in *2019 1st International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*, 2019, pp. 1–7.
- [9] A. Pinto and A. Matos, "Maresye: A hybrid imaging system for underwater robotic applications," *Information Fusion*, vol. 55, 08 2019.
- [10] S. Davis, K. G. Ricks, and R. A. Taylor, "Reflective fiducials for localization with 3d light detection and ranging scanners," *IEEE Access*, vol. 7, pp. 45 291–45 300, 2019.
- [11] M. I. Pereira, R. M. Claro, P. N. Leite, and A. M. Pinto, "Advancing autonomous surface vehicles: A 3d perception system for the recognition and assessment of docking-based structures," *IEEE Access*, vol. 9, pp. 53 030–53 045, 2021.
- [12] F. S. Neves, R. M. Claro, and A. M. Pinto, "End-to-end detection of a landing platform for offshore uavs based on a multimodal early fusion approach," *Sensors*, vol. 23, no. 5, 2023.
- [13] A. M. Pinto, P. Costa, A. P. Moreira, L. F. Rocha, G. Veiga, and E. Moreira, "Evaluation of depth sensors for robotic applications," in *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, 2015, pp. 139–143.
- [14] R. M. Claro, D. B. Silva, and A. M. Pinto, "Artuga: A novel multimodal fiducial marker for aerial robotics," *Robotics and Autonomous Systems*, p. 104398, 2023.
- [15] R. Shivan and Z. Dong, "Energy-efficient drone coverage path planning using genetic algorithm," in *2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR)*, 2020, pp. 1–6.
- [16] C. Franco and G. Buttazzo, "Coverage path planning for uavs photogrammetry with energy and resolution constraints," *Journal of Intelligent & Robotic Systems*, vol. 83, pp. 445–462, 2016.
- [17] C. Di Franco and G. Buttazzo, "Energy-aware coverage path planning of uavs," in *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, 2015, pp. 111–117.
- [18] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6423–6430.
- [19] A. Bircher, M. S. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, "Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots," *Autonomous Robots*, vol. 40, 08 2016.
- [20] M. D. Phung, T. H. Dinh, Q. Ha, and C. Quach, "Enhanced discrete particle swarm optimization path planning for uav vision-based surface inspection," *Automation in Construction*, vol. 81, pp. 25–33, 04 2017.
- [21] R. J. Silva, P. Nuno Leite, and A. M. Pinto, "Multi-agent optimization for offshore wind farm inspection using an improved population-based metaheuristic," in *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2020, pp. 53–60.
- [22] H. Baik and J. Valenzuela, "Unmanned aircraft system path planning for visually inspecting electric transmission towers," *Journal of Intelligent & Robotic Systems*, vol. 95, 09 2019.

- [23] C. Cao, J. Zhang, M. Travers, and H. Choset, "Hierarchical coverage path planning in complex 3d environments," in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 3206–3212.
- [24] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," Journal of the Operations Research Society of America, vol. 2, no. 4, pp. 393–410, 1954.
- [25] R. Claro, R. Silva, and A. Pinto, "Detection and mapping of monopiles in offshore wind farms using autonomous surface vehicles," in Global Oceans 2020: Singapore – U.S. Gulf Coast, 2020, pp. 1–4.
- [26] L. Perron and V. Furnon, "Or-tools," Google, 2019. [Online]. Available: <https://developers.google.com/optimization/>
- [27] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," Operations Research, vol. 12, no. 4, pp. 568–581, 1964.
- [28] M. Gendreau, A. Hertz, and G. Laporte, "New insertion and postoptimization procedures for the traveling salesman problem," Operations Research, vol. 40, pp. 1086–1094, 12 1992.
- [29] G. A. Croes, "A method for solving traveling-salesman problems," Operations Research, vol. 6, no. 6, pp. 791–812, 1958.
- [30] M. M. Flood, "The traveling-salesman problem," Operations Research, vol. 4, no. 1, pp. 61–75, 1956.
- [31] D. Roddier, C. Cermelli, A. Aubault, and A. Weinstein, "Windfloat: A floating foundation for offshore wind turbines," Journal of Renewable and Sustainable Energy, vol. 2, no. 3, p. 033104, 2010.
- [32] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239–256, 1992.
- [33] D. F. Campos, A. Matos, and A. M. Pinto, "Multi-domain mapping for offshore asset inspection using an autonomous surface vehicle," in 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2020, pp. 221–226.
- [34] D. Campos, A. Matos, and A. Pinto, "Multi-domain inspection of offshore wind farms using an autonomous surface vehicle," SN Applied Sciences, vol. 3, 04 2021.
- [35] P. Leite, R. Silva, A. Matos, and A. M. Pinto, "An hierarchical architecture for docking autonomous surface vehicles," in 2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2019, pp. 1–6.



RAFAEL M. CLARO received the M.Sc. degree in electrical and computer engineering from the Faculty of Engineering, University of Porto, Portugal, in 2020. He is currently pursuing the Ph.D. degree in electrical and computer engineering, in Faculty of Engineering of University of Porto. He is currently a Researcher with the Centre for Robotics and Autonomous Systems, INESC TEC.



MARIA I. PEREIRA received the M.Sc. degree in electrical and computer engineering from the Faculty of Engineering, University of Porto (FEUP), Portugal, in 2020. She is currently pursuing the Ph.D. degree in electrical and computer engineering, while conducting research with the Centre for Robotics and Autonomous Systems, INESC TEC. Her main research interests include maritime robotics, deep learning, and perception systems.



FRANCISCO S. NEVES received the M.Sc. degree in electrical and computer engineering from the Faculty of Engineering, University of Porto, Portugal, in 2021. He is currently pursuing the Ph.D. degree in electrical and computer engineering, in Faculty of Engineering of University of Porto. He is currently a Researcher with the Centre for Robotics and Autonomous Systems, INESC TEC.



ANDRY M. PINTO received the Ph.D. degree in electrical and computer engineering from the Faculty of Engineering, University of Porto, Portugal, in 2014. He is currently an Assistant Professor with the Faculty of Engineering, University of Porto, and a Senior Researcher with the Centre for Robotics and Autonomous Systems, INESC TEC. He is the Principal Investigator of national and international research and development projects related to robotic-based operation and maintenance (O&M) activities for offshore infrastructures. His main research interests include multi-domain perception, underwater imaging, artificial intelligence, and mobile robotics.

...