

Article

Adaptive Path Planning for Fusing Rapidly Exploring Random Trees and Deep Reinforcement Learning in an Agriculture Dynamic Environment UAVs

Gabriel G. R. de Castro ^{1,*}, Guido S. Berger ^{2,3,4}, Alvaro Cantieri ⁵, Marco Teixeira ⁶, José Lima ^{2,3,7}, Ana I. Pereira ^{2,3} and Milena F. Pinto ¹

¹ Department of Electronics Engineering, Federal Center of Technological Education of Celso Suckow da Fonseca (CEFET/RJ), Rio de Janeiro 20271-204, Brazil

² Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal

³ Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal

⁴ Engineering Department, School of Sciences and Technology, Universidade de Trás-os-Montes e Alto Douro (UTAD), 5000-801 Vila Real, Portugal

⁵ Applied Robotics and Computation Laboratory—LaRCA, Federal Institute of Paraná, Pinhais 3100, Brazil

⁶ Coordenação do Curso de Engenharia de Software, COENS, Universidade Tecnológica Federal do Paraná—UTFPR, Dois Vizinhos 85660-000, Brazil

⁷ INESC Technology and Science, 4200-465 Porto, Portugal

* Correspondence: gabriel.gelard@aluno.cefet-rj.br

Abstract: Unmanned aerial vehicles (UAV) are a suitable solution for monitoring growing cultures due to the possibility of covering a large area and the necessity of periodic monitoring. In inspection and monitoring tasks, the UAV must find an optimal or near-optimal collision-free route given initial and target positions. In this sense, path-planning strategies are crucial, especially online path planning that can represent the robot's operational environment or for control purposes. Therefore, this paper proposes an online adaptive path-planning solution based on the fusion of rapidly exploring random trees (RRT) and deep reinforcement learning (DRL) algorithms applied to the generation and control of the UAV autonomous trajectory during an olive-growing fly traps inspection task. The main objective of this proposal is to provide a reliable route for the UAV to reach the inspection points in the tree space to capture an image of the trap autonomously, avoiding possible obstacles present in the environment. The proposed framework was tested in a simulated environment using Gazebo and ROS. The results showed that the proposed solution accomplished the trial for environments up to 300 m³ and with 10 dynamic objects.

Keywords: aerial robots; multiple robots; path planning; dynamic environment; precision agriculture



Citation: Castro, G.G.R.d.; Berger, G.S.; Cantieri, A.; Teixeira, M.; Lima, J.; Pereira, A.I.; Pinto, M.F. Adaptive Path Planning for Fusing Rapidly Exploring Random Trees and Deep Reinforcement Learning in an Agriculture Dynamic Environment UAVs. *Agriculture* **2023**, *13*, 354. <https://doi.org/10.3390/agriculture13020354>

Academic Editors: Yanbo Huang and Jin Yuan

Received: 23 December 2022

Revised: 20 January 2023

Accepted: 28 January 2023

Published: 31 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With recent advances in optics, electronics, and informatics, countless possibilities have emerged for modern robotics systems [1]. As stated in Almaki et al. [2], the fourth industrial revolution (4IR) is bringing new opportunities to employ advanced technology to achieve an efficient and sustainable means of boosting crop productivity. For instance, unmanned aerial vehicles (UAVs) open new paths that advance the solution of existing problems, such as for package delivery in logistics [3], in the field of agriculture 4.0 [4], Industry 4.0 [5], artificial intelligence [6], the Internet of Things (IoT) [7], remote sensing [8] and many others. In precision agriculture, UAVs are widely used to map the spatial and temporal variability of various parameters in the cultivation environment, helping producers in decision-making and consequently improving agricultural management [9,10]. In this sense, according to Yaqot et al. [1], UAVs are a sustainable and efficient tool in the

agro-industry due to the automation level of their operation being low compared with other industry solutions.

These unmanned aerial robots have been applied in many different tasks due to their ability to perform activities that involve different levels of complexity, dynamicity of the environment, and need for motion flexibility [11–13]. Inspection of large areas is one of the leading applications that UAV systems are required to automate [14]. Regarding inspection and monitoring tasks in the agriculture context, UAVs play an essential role in ensuring that crops receive the care necessary to maintain their health and productivity [15,16].

UAVs represent the largest segment of robotic platforms used in precision olive growing, followed by the application of satellites [17]. The great advantage of UAVs compared to satellites is the fast acquisition of data and high resolution of images [18]. On the other hand, satellites have the benefit of covering large areas, although sometimes the presence of clouds will influence the quality of information collected [18].

Precision agriculture is undergoing a massive technological transformation where the applicability of UAVs in the agricultural segment is not restricted to remote-sensing techniques. With the integration of artificial intelligence [19], the IoT [20], and computer vision resources [21], it is now possible for UAVs to reach new layers of application for automating inspection processes and generating data for supporting decision-making, such as for selective and reduced pesticide applications, for example.

In autonomous inspection and monitoring tasks, the UAV needs to find an optimal or near-optimal collision-free route and target positions, demanding continuous monitoring of the vehicle during operation. There are two types of motion-planning strategies based on the sensory information acquired from the environment: global path planning and local path planning [22]. In the literature, several solutions regarding path-planning strategies have been described [23,24]. In dynamic environments, if the sensors detect unaccounted obstacles, the path planning must present a rapid response to re-compute the assigned path to avoid the detected obstacle until it reaches its final position [25].

1.1. Olive Trees Insect Trap Inspection

Olive tree cultures *Olea europaea* L. are an important economic source for several countries. One of the most significant plagues that affects olive production is the *Bactrocera oleae*, commonly known as the “olive fly”. The female of the fly puts its eggs in the fruits where the growing larvae feed, impacting the product’s quality. Fighting this kind of infestation commonly uses pesticides, but ecological concerns and the associated economic costs necessitate proposal of new methods to reduce the use of these products. One possible way to achieve this objective is to use insect traps to capture the fly and evaluate the infestation level, allowing intelligent decision-making control. This method is widely applied in this kind of culture [26]. The traps are fixed in the tree’s crops at eye level, in variate numbers and semi-randomly distributed [27].

A range of similar variate manual inspection methodologies are used. It is common to exchange the traps every two weeks, for example, and perform a visual count of the insects captured, which is a slow and demanding process. Moreover, the time interval for data analysis can be problematic because of the delay in action to control the infestation. In this sense, the development of methodologies for monitoring insects in chromotropic traps by UAVs has been explored in the literature, with the objectives of replacing traditional methods carried out by human observation, analyzing on a large scale the dynamics of insect distribution, increasing the flow of information in a short time, as well as visualizing and summarizing this flow of information in a statistically reliable sense [17]. A possible way to achieve this objective is by using autonomous small-size UAV systems due to their flexible characteristics.

Motion-planning algorithms are needed for autonomous UAVs that operate in dynamic and high-dimensional environments. Developing autonomous UAVs for this kind of application remains challenging, especially in low-altitude flight operations in large-scale unstructured and dynamic environments involving operation between tree rows under

canopy level to conduct the trap-inspection process [28]. It is necessary to create a viable and low-energy cost path to the UAV and to provide obstacle detection and avoidance to ensure motion security. Proposing an intelligent path-planning algorithm with collision-avoidance capability for this specific application is the objective of the present work.

1.2. Objectives and Main Contributions

The main objective of the present work is to propose an autonomous path-planning and collision-avoidance algorithm for a UAV applied to the olive fly trap inspection task, providing a viable route between the tree crops and allowing the UAV to reach the pre-programmed inspection points avoiding static and dynamic obstacles present in the space.

Therefore, this work aims to advance a methodology for online adaptive path planning based on the fusion of rapidly exploring random trees (RRT) and deep reinforcement learning (DRL) applied to unmanned aerial vehicles. This research considers an olive-growing environment to assess the proposed methodology. In this scenario, the UAV will move among the aisles of olive groves. During this movement, the UAV will hover at previously known inspection set points to allow the capture of a trap image. Implementing the capture algorithm, image-processing, and automatic insect counting is beyond the scope of this study. This work will only address the online path-planning problem and collision avoidance of the UAV in this navigation process. The framework is assessed by simulations performed in the robot operating system (ROS) along with the Gazebo platform [29].

The main contribution of this work is the development of a novel adaptive path-planning approach for UAVs with obstacle-detection capability, based on the fusion of RRT and DRL algorithms, applied to capture images from insect traps fixed in the olive trees, considering the dynamic agriculture environment.

1.3. Organization

The next part of this paper is organized as follows: Related work on the application of UAVs in precision agriculture and online path planning is reviewed in Section 2. Section 3 presents an overview of the proposed path-planning strategy and its mathematical foundations. The validation and assessment of the proposed strategy are described in Section 4. Finally, concluding remarks and ideas for future work are given in Section 5.

2. Background and Related Work

2.1. UAVs Applied to Precision Agriculture

In precision agriculture, UAVs, along with several technologies, such as sensors and companion computers, are used to estimate the characteristics of the local culture. They can determine culture vigor using multispectral or hyperspectral cameras [30], potential local water stress using thermal cameras [31], pest and disease severity using computer-vision techniques [32], or by reconstructing three-dimensional maps using LiDAR (light detection and ranging) sensors to estimate the density of local vegetation [33], among others. For instance, precision olive growing represents an essential agricultural management method for the countries of southern Europe that aims to maximize the yield of olive groves, offering benefits from productive, qualitative and environmental points of view. Given this fact, developing and applying new technologies to monitor olive groves using multi-rotor UAVs has increased rapidly in recent decades [34].

In this sense, different solutions are presented in the literature, aiming to maximize the inspection process of various parameters of the local culture. UAVs equipped with thermal cameras and applying computer-vision resources are used to collect valuable information, which can be used, for example, to analyze and classify regions with potential water deficit, identify the water needs of a set of olive groves, and to use the local distribution of water more efficiently, thus avoiding risks in located areas [35]. Aerial systems equipped with multispectral and hyperspectral cameras can estimate regions with potentially deficient metabolic processes to indicate diseases in olive groves [36]. Using the same sensory technology, it is also possible to estimate potential problems related to nutritional stress

through spectral reflectance image analysis. Quantifying the vigor of the local biomass using different vegetation indices can later indicate regions needing fertilizers [37].

Usually, UAVs are applied at a very high altitude when monitoring crops. In this case, the photogrammetry technique is widely used for olive grove canopy reconstruction. Typically, the UAV is equipped with an RGB camera to carry out counting and forest inventory in the region [38]. Using LiDAR sensors, three-dimensional maps with high spatial resolution are used to obtain production estimates based on the canopy diameter. Moreover, the approach can be used to carry out a forest inventory or even assess damage in forest plots due to different phenomena, such as fires [39]. Despite recent technological advances, there is still a significant reliance on manual sampling and human observation for crop monitoring [40]. In addition, most operations are conducted at a high flight ceiling, preventing more accurate and localized identification of problems existing at the site [41]. Furthermore, when using UAVs, a human operator must constantly check the UAV trajectories and performance of the task.

Agriculture autonomous robotic system UAVs present significant challenges [42], especially considering that the agricultural environment can be unstructured and highly dynamic, potentially including obstacles of different dimensions, such as tree branches or leaves, gusts of wind, occlusion caused by multiple obstacles, and light variation [43]. Most studies reported in the literature focus on UAV applications in open spaces. Applications that require high proximity to particular targets, such as identifying insects or traps scattered along the canopies of olive groves, demand precise navigation control near to the trees [28]. A possible approach to achieve this objective is the use of neural networks trained to convert the inputs (e.g., vehicle and obstacle positions) to a robot's headings to guarantee collision-free movement [44,45], which is the case for the research reported in this work. Several solutions have been proposed to address mobile robots' motion planning. For instance, Sing et al. [44] proposed using an NN to control the mobile robot speed. Sung et al. [45] addressed NN performance depending on the training path data. Yu et al. [46] proposed a path-planning algorithm for mobile robots based on neural networks and hierarchical reinforcement learning. For UAVs, [47] developed an online control algorithm whereby a base station downloads the UAS state and trains a Hamilton–Jacobi–Bellman equation (HJB) NN that is solved in real-time, yielding a sub-optimal control action. Their approach was validated through numerical simulation. All the studies referred to rely on non-realistic environment simulation to test their approaches.

2.2. Online Path Planners and Reinforcement Learning Path Planners

In recent decades, several kinds of 3D path-planning approaches have been proposed, including those described in [48–50], among others. These proposed path-planning strategies have specific features that can be useful for different robots used in different environments. They include Visibility Graph [51], randomly sampling search algorithms, such as Rapidly Exploring Random Tree [48] and Probabilistic Roadmap [49], optimal search algorithms, such as Dijkstra's algorithm [52], A^* [53], D^* [54], and bio-inspired planning algorithms [55], among others.

There are two types of NN implementation for UAV path planning. In the first approach, the path of the UAV is based on a sample trajectory. Then, a computing method is utilized to perform the trajectory optimization [56]. The other type of implementation uses NNs to approximate the system dynamics, objective functions and gradients. This reduces the size of the nonlinear programming (NLP) problem [57]. Usually, the NNs are mixed with other algorithms, such as artificial potential field (APF), GA, PSO, etc. [58].

A path-planning strategy using an NN for a UAV in a dynamic environment was proposed in the work of Chen et al. [59]. In their proposal, classical APF issues were minimized (i.e., the local minimum issue). The authors of [60] designed a controller based on a multilayer feed-forward NN for autonomous mobile robot navigation. An interesting approach using an NN and reinforcement learning was proposed in [61]. Their approach was to make the mobile robot learn navigation rules automatically without initial rules settings.

More recently, Queresh et al. [62] designed a deep neural network (DNN) based on an iterative motion-planning algorithm for high-dimensional problems. They encoded the given workspaces directly from a point cloud measurement. Then, an end-to-end collision-free path was generated. The tests were evaluated using a seven DOF Baxter robot manipulator. Shiri et al. [47] developed an online NN learning-based control algorithm HJB for UAV. However, only numerical simulations were conducted. Another solution for UAV was proposed by [63]. Their work involved the development of a deep-reinforcement-learning (DRL)-technique for UAV path planning based on global situation information.

Regarding unknown and dynamic environments, some studies in the literature have utilized reinforcement learning strategies to navigate UAVs in this kind of scenario [64–66]. An improvement in reinforcement learning can be achieved by fusing it with a deep learning approach to avoid the curse of dimensionality, as stated by [64]. This strategy helps when considering high-dimensional spaces.

As mentioned by Zhang et al. [66], real-time path planning considering multiple UAVs and information-sharing is still not well studied. Thus, they proposed using geometric reinforcement learning (GRL) to calculate the path planning of multiple UAVs. The reward matrix of the GRL is adaptively updated based on the geometric distance and risk information shared by the robots. The perspective of using Q-Learning as a reinforcement learning strategy is a classical solution for path-planning problems. However, in many real-world problems, there are several possible states and action spaces. In this sense, tabular representation cannot store all possible combinations of pair values. To mitigate this issue, in the work of Tong et al. [64], the authors developed an improved DRL approach based on a deep Q-network (DQN) to decompose the UAV navigation task into two simpler sub-tasks. This approach provides robust navigation in highly dynamic environments. Another interesting approach was proposed by [67]. They developed a solution for double Q-network deep reinforcement learning to perform dynamic path planning in an unknown environment. The authors of Yang et al. applied DQN for multi-robot path planning to slow convergence and excessive randomness. They showed that their solution converged faster than the classic DRL. Moreover, it was quicker to learn the path-planning solutions. Therefore, this work combined RRT and DQN to eliminate drawbacks possessed by each algorithm. DQN is difficult to implement and is not a robust method. RRT is easy and robust but has the disadvantage of being a random sampling method. Table 1 summarizes the advantages and disadvantages of contemporaneous path-planning methods.

Table 1. Advantages and disadvantages of contemporaneous path-planning algorithms.

Method	Advantages	Drawbacks	References
RRT	Simple, low computational time	For static environments, non-optimal paths	Yang et al. [48]
PRM	Can handle complex environments	For static environments, non-optimal, time expensive	Yang et al. [49]
Dijkstra's Algorithm	Easy implementation	High time complexity, for static environment	Dijkstra et al. [52]
A*	Fast searching ability	Static environment only, heavy time burden, non-smoothness	Hart et al. [53]
APF	Fast convergence	Local minima	Chen & Zhang [59]

Table 1. *Cont.*

Method	Advantages	Drawbacks	References
Genetic algorithm	Able to solve NP-hard and multiobjective problems	High time complexity, premature convergence	Tsai et al. [68]
CNN	Can handle dynamic environments and unknown environments	High complexity, high number of hyperparameters, non-optimal	Wu et al. [69]
DQN	Adaptative	Non-optimal, difficult implementation	Tong et al. [64]

3. Proposed Methodology

3.1. Problem Formulation

The olive fly trap inspection process using a UAV is performed by capturing images of the traps fixed in the crop of trees along a growing space. This kind of space is non-structured for the UAV flight, demanding a capacity for collision detection and avoidance to assure the security of the process. Figure 1 presents a picture of an olive-growing area (a) and the closed vision of the trap fixed in the tree crop (b).

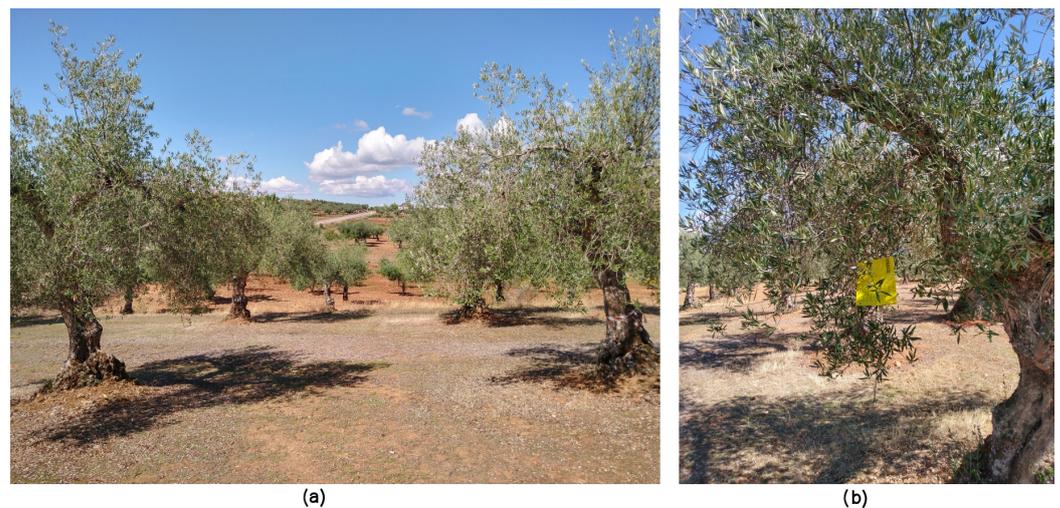


Figure 1. Example of olive-growing space environment complexity. (a) A picture of an olive-growing space and trees. (b) Detail of a trap fixed in the tree crop.

Capturing the trap images requires the UAV to fly between the tree crops at a height near the mean of the tree sizes. As is possible to observe in the figure, the flight space is complex, with some twigs projecting outside the tree crop. In addition, other kinds of dynamic objects, such as vehicles, humans, and animals, are sometimes present in this space, requiring a response from the flight controller to avoid them when the UAV is moving.

A representation of the inspection process is presented in Figure 2. In this figure, it is possible to observe the proposed behaviour of the UAV during the displacement to reach an inspection point. The UAV receives the route program and starts moving to the target point while scanning the environment using the LIDAR. When an obstacle is detected, the DRL collision avoidance algorithm sets the UAV avoidance operation to bypass the object. When the UAV arrives at the trap point, it stops, and the image is captured, before moving to the next target point. In this work, some parameters are considered to evaluate the proposed solution:

- The traps are fixed in a random distribution between the spaces, with five traps per tree.
- The distance for capturing the trap picture is set to 3.0 m minimum.
- The UAV is capable of detecting obstacles in space using a planar LIDAR with 10.0 m range and a 360-degree scanning angle.
- The position of each trap is already known and is used to set the target points to the route planner.
- Object avoidance is only performed in the horizontal plane. No vertical direction change is performed to avoid obstacles.

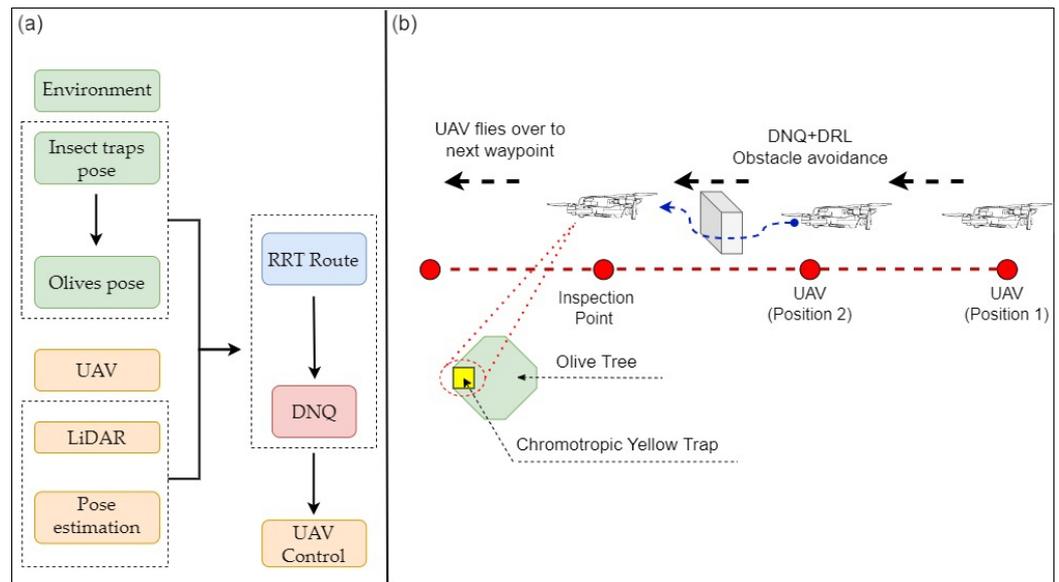


Figure 2. Representation of the UAV trap-inspection process. (a) Block diagram of the algorithm agents. (b) Representation of the UAV displacement and avoidance actions.

The choice for the UAV not to execute vertical collision avoidance is because of the LIDAR scanning model. There is no sensor embedded in the UAV to detect objects in the top-side or up-side of the aircraft, so it is impossible to implement vertical detection to allow the algorithm to work in this direction. This was chosen because it is expensive and complex to embed many sensors in a real-world application in the UAV, so the choice of working only in the horizontal plane to avoid obstacles is justified.

The path-planning concept focuses on fusing RRT and DRL techniques. The RRT will generate the path among the aisles of olive groves and the DRL will learn and adapt the UAV current state path to guarantee a collision-free trajectory. In the case of dynamic obstacles, it is considered that the robot can share the space with other robots, people working on the farm, tree branch movement, etc. As proof of concept, a model of a car is used as a dynamic obstacle for the UAV during the simulations.

The UAV will take off from a home point, perform the inspection process, avoid possible obstacles, and then return to the home point. In this sense, the authors believe that the algorithms chosen in this research fit the context of the proposed research objectives, considering that the RRT can reach a map exploration rate in a short period and that the DRL has advantages over other techniques of control for dynamic systems and is capable of handling highly dynamic and time-varying environments.

3.2. UAV Quadrotor Model

The dynamics of the UAV were provided by the PX4 package and are detailed and described in Andrade et al. [70]. The UAV system considered in this work has four rotors, as presented in Figure 3.

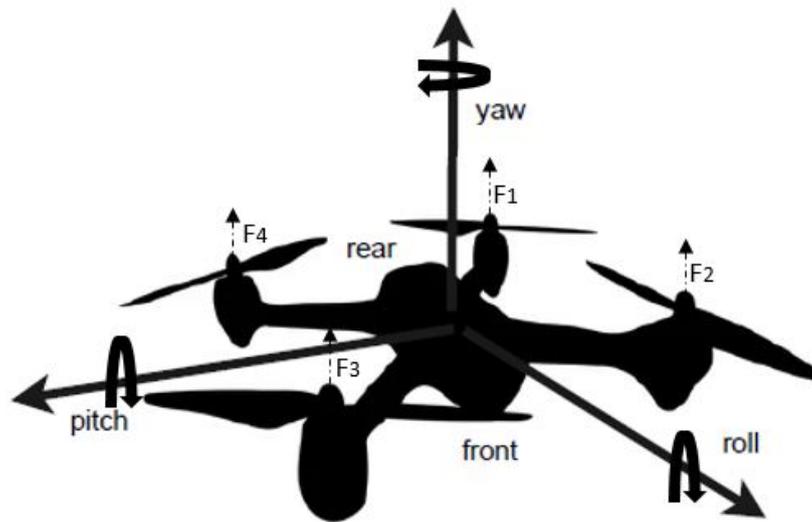


Figure 3. UAV quadrotor structure.

The Euler angles presented in the figure describe the orientation of the system about the local level surface, as well as in relation to the true north (azimuth reference). The roll (ϕ) and pitch (θ) angles give the tilt of the UAV about the gravity vector, while the yaw angle (ψ) describes the rotation around the local vertical direction. These angles, as shown in Equation (1), are used as inputs to the forward kinematics equation to calculate the world-to-frame conversion matrix to estimate the actual UAV position.

$$R_{\psi,\theta,\phi} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}. \quad (1)$$

Equation (2) gives the UAV body frame axes mathematical model.

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \\ \ddot{z} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \frac{I_{yy}-I_{zz}}{I_{zz}}\dot{\theta}\dot{\psi} + \theta \frac{J_r}{I_{xx}}\Omega_r + \frac{1}{I_{xx}}U_2 \\ \frac{I_{zz}-I_{xx}}{I_{yy}}\dot{\phi}\dot{\psi} + \phi \frac{J_r}{I_{yy}}\Omega_r + \frac{1}{I_{yy}}U_3 \\ \frac{I_{xx}-I_{yy}}{I_{zz}}\dot{\theta}\dot{\phi} + \frac{1}{I_{zz}}U_4 \\ g - \frac{\cos(\phi)\cos(\theta)U_1}{m} \\ \frac{U_1}{m}(\Delta + \sin(\phi)\sin(\psi)) \\ \frac{U_1}{m}(\Gamma + \sin(\phi)\cos(\psi)) \end{bmatrix} = \begin{bmatrix} \zeta_\phi(t) \\ \zeta_\theta(t) \\ \zeta_\psi(t) \\ \zeta_h(t) \\ 0 \\ 0 \end{bmatrix}. \quad (2)$$

Note that $\Delta = \cos(\phi)\sin(\theta)\cos(\psi)$ and $\Gamma = \cos(\phi)\sin(\theta)\sin(\psi)$ are auxiliary variables, U_i ($i = 1, \dots, 4$) represents the thrust and the actuation torques for the UAV movements, Ω_r is the engine's residual angular speed, and, finally, $\zeta_{(\phi,\theta,\psi,h)}$ are the dynamic disturbances affecting the angles ϕ , θ , and ψ . Note that these variables are used in the motor mixing (i.e., Equation (2)), which specifies the magnitude of forces that should be applied to each motor of a multirotor. U_1 is the thrust equivalence force, U_2 represents the rotational force on roll axis, U_3 represents the rotational force on the pitch axis, and U_4 represents the rotational force on the yaw axis. Thus, it is possible, using the motor mixing algorithm, to analyze the U_i inputs in relation to the angular speed ($\Omega_{(1,2,3,4)}$). The $b(Ns^2)$ and in Equation (3) are the impulse coefficient and the drag coefficient, respectively.

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ \Omega_r \end{bmatrix} = \begin{bmatrix} b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ b(-\Omega_2^2 + \Omega_4^2) \\ b(\Omega_1^2 - \Omega_3^2) \\ d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{bmatrix}. \quad (3)$$

3.3. Path-Planning Algorithms

3.3.1. Rapidly Exploring Random Tree Algorithm

RRT is an algorithm based on decision trees and random sampling searches around a determined state space. This space of search is described by Z , wherein a three-dimensional environment Z is a subset of the R^3 . The space occupied with objects is Z_{occ} , and the free one is Z_{free} . The main objectives of this algorithm are to create a path between $Z_{start} \subset Z_{free}$ and $Z_{target} \subset Z_{free}$ without collision, in the least possible time and with least path cost. Equation (4) shows the relation between these three spaces.

$$Z_{free} = \frac{Z}{Z_{occ}} \cdot Z_{target}. \quad (4)$$

In this work, the authors focused on using the RRT algorithm to tackle the problem of finding a reliable path between some given waypoints (states) that belong to Z_{free} . For the 3D path planning, the waypoints are defined by $S(x, y, z) \subset Z_{free}$. The generated path should link all these states but only sometimes be optimal. Since RRT is a sample algorithm, there is no guarantee of finding the optimal solution to the problem. The authors then use specific criteria to stop the algorithm using N the numbers of max iterations and J the minimal distance path. Algorithm 1 provides a description of how it works.

Algorithm 1 Rapidly Exploring Random Tree

```

1: Path ← Empty_Array
2: Waypoints ← Insert_Waypoints()
3: for Z_start, Z_target in Waypoints do
4: end for
5: R_tree ← Begin_Tree()
6: R_node ← Begin_node(Z_free, Z_start, R_tree)
7: for i = 0 to i = N do
8: end for
9: if random() < 0.10 then
10: else
11: end if
12: Z_random = random(Z_near_target)
13: Z_random = random(Z_free)
14: Z_near ← Nearest(R_tree, Z_random)
15: if Path_collision_free(Z_near, Z_random) then
16: end if
17: R_tree ← Z_random
18: if Norm(Z_random - Z_target) <= J then
19: end if
20: Path ← Get_minimal_path(R_tree)
21: Break

```

3.3.2. Deep Reinforcement Learning and Deep Q-Network

The DQN algorithm is a DRL method. The rationale for using the DQN algorithm is that it can combine deep learning and reinforcement learning algorithms. In this sense, the DQN algorithm effectively solves the problems and challenges faced by the deep learning and reinforcement learning fusion process by combining the Q-learning algorithm and

empirical playback mechanism and generating a target Q value based on a convolutional neural network [71]. DQN is based on Q learning and, for this, uses a convolutional neural network (a deep neural network) to represent the action value function. The network is trained based on reward feedback from the game. Like other neural network machine learning approaches, DQN can be seen as a three-step challenge.

First, there is the training phase. Second, there is the validation. The final step is about testing the performance. In this work, the DQN algorithm is applied to dynamically adapt the UAV's previous trajectory (actuation stage) when random objects are en route to collision with the agent.

A DRL algorithm in an autonomous navigation task aims to find the optimal balance to guide the UAV to its target position through interaction with the environment. These methods represent the navigation process as a Markov decision process (MDP) that uses the sensor observation as a state, aiming to maximize the expected performance of the action. DRL-based navigation has the advantages of solid learning ability and low dependence on sensor accuracy. The DRL algorithm, when integrating with a given agent, can replace the localization and map construction module, being able to move to the destination point, avoiding static or dynamic obstacles present in the environment [72].

The main idea of the present study is to develop a trajectory planning methodology for UAVs based on RRT and DRL algorithms. The proposed method in this research consists of two main stages. The UAV starts its navigation system in an unknown environment by traversing the location to collect information through the RRT offline path-planning algorithm. The RRT creates points at random locations throughout the existing state space in the flying area of the UAV and quickly grows a "tree" from the start point to the end point, thus creating multiple paths that the UAV can traverse.

In the second step, the DQN algorithm is applied to dynamically adapt the trajectory of the UAV, seeking optimized routes for UAV navigation. Suppose the path that the UAV is taking is not free—in that case, the DRL obstacle avoidance algorithm is initialized, preventing the agent from entering a collision course, either by static or dynamic obstacles.

Figure 4 illustrates the methodology proposed for this research. Faced with an unknown scenario, namely scenario A, the UAV starts the trajectory planning algorithm by exploring multiple path options in a complex and unstructured environment. Once the RRT algorithm has calculated all possible routes, optimized route planning is performed based on the DQN and DRL algorithms, making it possible to deal with changes in the environment in real time, as illustrated in scenario B.

Based on the proposed methodology, the objective of the present study is to implement an offline system with reduced computational capacity, rapid convergence for decision-making in highly dynamic environments, and reduced path length to be traversed by the UAV.

The authors designed two different environments for this work in the Gazebo software [29]. One is applied during the training DQN phase and the other tests the application's robustness. They are shown in Figures 5 and 6, respectively.

It is possible to observe the clearness of the training environment when facing the complexity of the test one. This is to improve the simulation time step, which improves the time spent training the model. The training environment has only twenty two models, including UAV (Figure 7), pickup (Figure 5b), and olive trees (Figure 5c). The olive trees are the static models, while pickup can move in the Z_{free} plane changing acceleration \vec{P}_a and velocity \vec{P}_v by every agent's iteration with the environment. Equation (5) shows how these two vectors are described, using (Θ, Φ) as a random vector to describe the orientation and (X_1, Y_1) (X_2, Y_2) to determine the module.

$$\begin{aligned}
 \vec{P}_a(\Theta) &= \Theta \mid \Theta \in Z \text{ and } 0 \geq \Theta \leq \pi, \\
 \vec{P}_v(\Phi) &= \Phi \mid \Phi \in Z \text{ and } 0 \geq \Phi \leq \pi, \\
 \|\vec{P}_a(X_1, Y_1)\| &= \sqrt{X_1^2 + Y_1^2} \mid (X_1, Y_1) \in Z \text{ and } 0 \geq (X_1, Y_1) \leq 2, \\
 \|\vec{P}_v(X_2, Y_2)\| &= \sqrt{X_2^2 + Y_2^2} \mid (X_2, Y_2) \in Z \text{ and } 0 \geq (X_2, Y_2) \leq 2.
 \end{aligned}
 \tag{5}$$

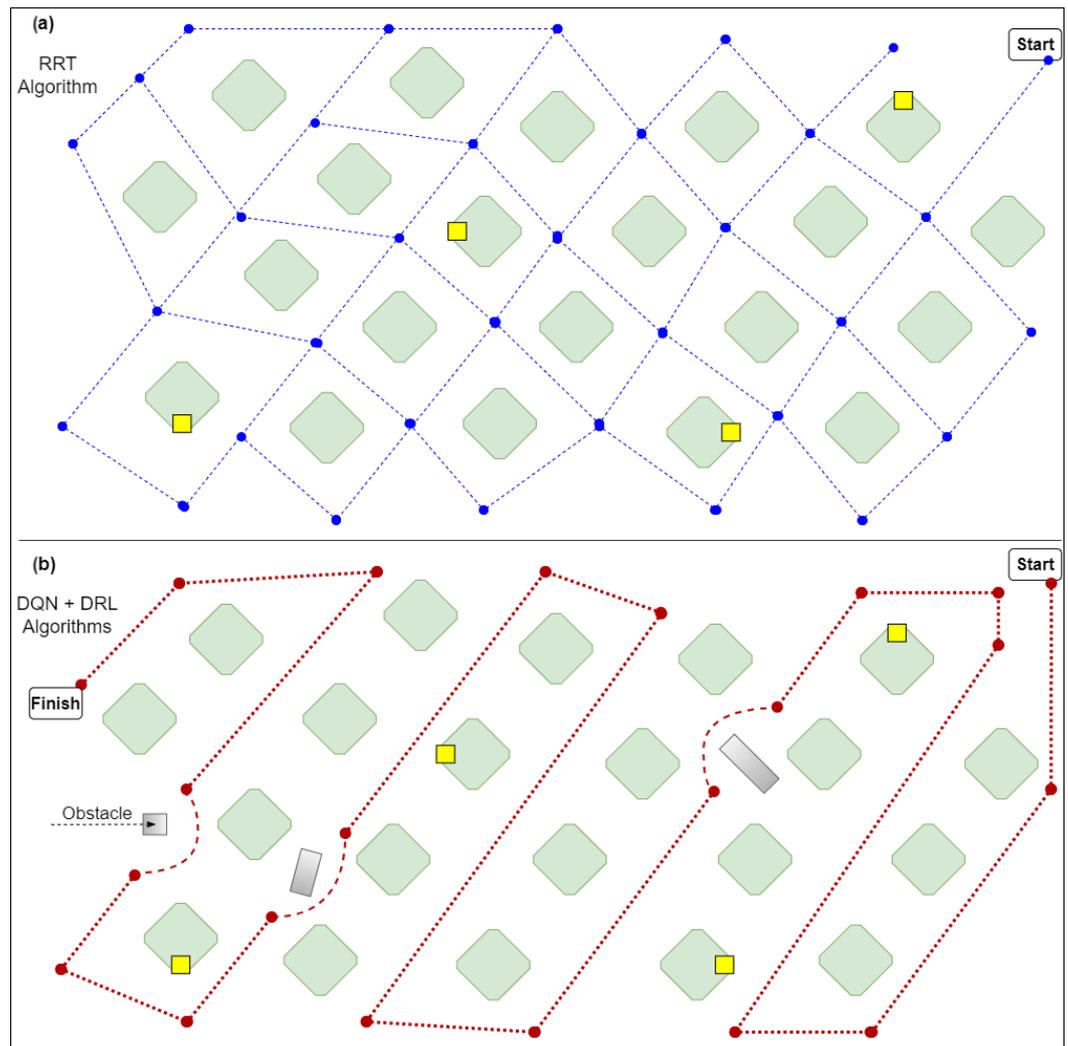


Figure 4. Overall idea of the proposed methodology. In Figure (a), the UAV starts the trajectory planning algorithm by exploring multiple path options in a complex and unstructured environment. Figure (b) illustrated when the RRT algorithm calculated all possible routes, and the optimized route planning was performed based on the DQN and DRL algorithms, making it possible to deal with changes in the environment in real-time.

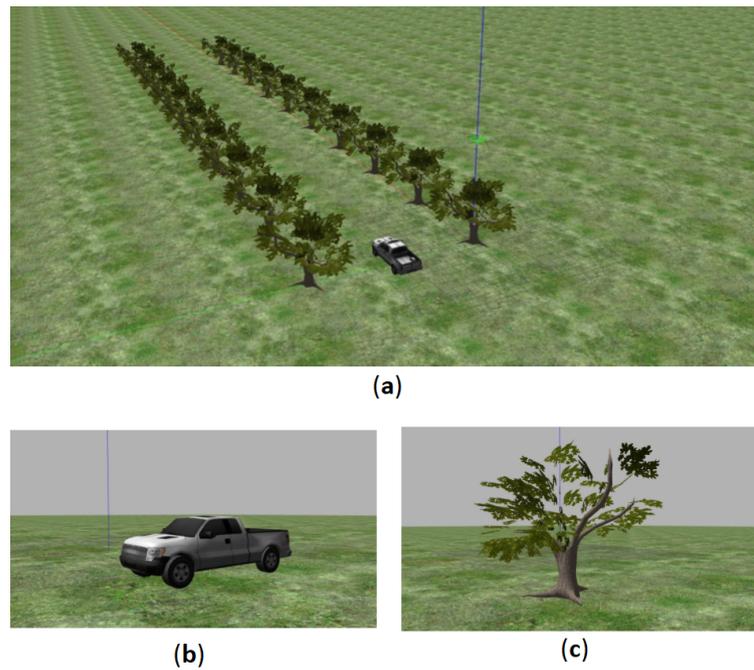


Figure 5. Environment built in the Gazebo software. (a) Simplified world used for training DQN. (b) Pickup model. (c) Olive-tree model.

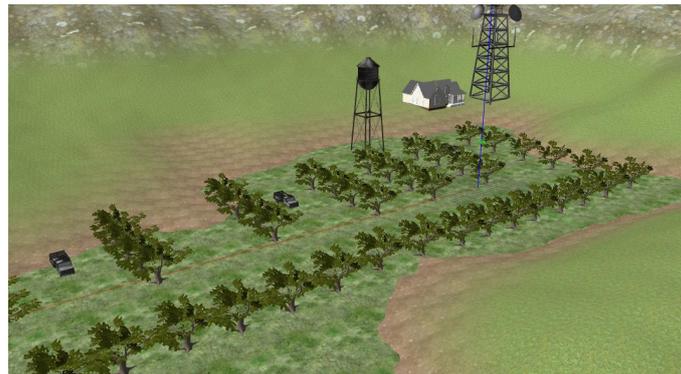


Figure 6. Olive-growing environment simulation.



Figure 7. IRIS UAV used in the simulation.

3.3.3. Training Phase

The authors decided to split the training phase into two different scenarios. The first is a random simulated environment that does not simulate any kinematics or physics during the simulation. The second is the previously discussed environment, with all the physics and kinematics provided by the PX4 ROS package and Gazebo software [29].

The main reason for this split is the time consumed during the training. Therefore, the authors decided to first test neural network structures and hyperparameters on the non-realist simulation and to fine-tune them in the complex environment.

Simple Simulation

In this implementation, the environment Z is split into grids of 1 m^3 each, the UAV has the exact size of one grid, and each obstacle fills N numbers of grids. The UAV actual pose and objects compose the Z_{occ} space. The space of states is described by S_{state} and comprises information about the surrounded UAV grids, the difference between the UAV header angle D_α and targets $T(x, y, z) \subset Z_{free}$, and the output DQN is the next action the UAV will take in the next step. Algorithm 2 shows the entire implementation.

Algorithm 2 Simple environment DQN implementation.

```

1:  $Events \leftarrow$  Number of events
2:  $Steps \leftarrow$  Number of steps
3:  $Z_{occ} \leftarrow$  InitiatileOjs( $N$ )
4:  $Starts \leftarrow$  random.starts( $Z_{free}$ )
5:  $Targets \leftarrow$  random.targets( $Z_{free}$ )
6:  $Dqn \leftarrow$  InitializeNeuralNetwork()
7: for  $pose, goal$  in  $Starts, Targets$  do
8:   for  $e$  in  $events$  do
9:      $NewPose = pose$ 
10:    for  $s$  in  $steps$  do
11:       $readings \leftarrow$  Readings()
12:       $S_{state} =$  DefineState( $D_\alpha, readings$ )
13:       $Dqn.reward \leftarrow$  GetReward( $S_{state}$ )
14:       $Dqn.memory \leftarrow$  ConstructMemory( $S_{state}, NN.reward$ )
15:       $Action \leftarrow$   $Dqn.output(S_{state})$ 
16:       $NewPose = NewPose + Action$ 
17:      if  $Dqn.memory.size > samples$  then:
18:         $BatchState, BatchReward \leftarrow$  random.sample( $Dqn.memory, samples$ )
19:         $Dqn.TrainModel(BatchState, BatchReward)$ 
20:      end if
21:    end for
22:  end for
23: end for
24:  $Dqn.SaveModel()$ 

```

1 . *InitializeObjs*(N): This function only receives the number of the grid that will be random pop from Z_{free} space and add to Z_{occ} .

2 . *random.targets*(Z_{free}), *random.starts*(Z_{free}): Each function returns a list of arrays, random gets from Z_{free} space.

3 . *InitializeNeuralNetwork*(): Construct the DQN class and its model's parameters. Input size, output size, numbers of layers, neurons, activation functions, memory, and optimizer are some of them.

4 . *Readings*(): Returns an array with the surrounded area information if there is a grid that belongs to Z_{occ} a piece of this array gets the value 1. If not, it receives 0.

5 . *GetReward*(): Based on the agent's state, it returns the reward. For this simulation, the reward is a negative value based on how distant the actual state is from the target and if the state is equal to it, returns a positive reward.

6 . *ConstructMemory*(): Push into the memory array (initialize with *InitializeNeuralNetwork*()) the presented knowledge experimented by the agent.

4. Results and Discussion

4.1. Hardware and Environment Setup

The simulations were performed on a notebook with 8GB RAM and a 2.7 GHz Core-i5-5200 processor. The operating system distro was a 64 bits Ubuntu 18.04 bionic with a desktop environment LXDE. All algorithms were implemented using Python3.10. All codes and related information are available at https://github.com/gelardrc/RRT_DQN_PATH_PLANNING.git (accessed on 10 December 2022). Figure 7 shows the IRIS UAV used in the Gazebo software simulation. The Gazebo is a complete software simulation that provides many built-in models for the designer to construct a realistic environment. Gazebo not only contributes to the visual aspects, but also enables setting up of a whole dynamic interface, including gravity, wind, dynamic elements, etc.

Figure 6 presents the designed scenario for testing the proposed methodology. Note that this UAV included a PixHawk PX4 flight controller unit (FCU). The MavLink communication protocol was used for telemetry and offboard controllers. ROS Kinetic was used for developing the controller, the trajectory calculation, and camera node packages in the Python programming language. The MavLink was simulated through the MavROS package. Figure 8 shows an overview of the implementation.

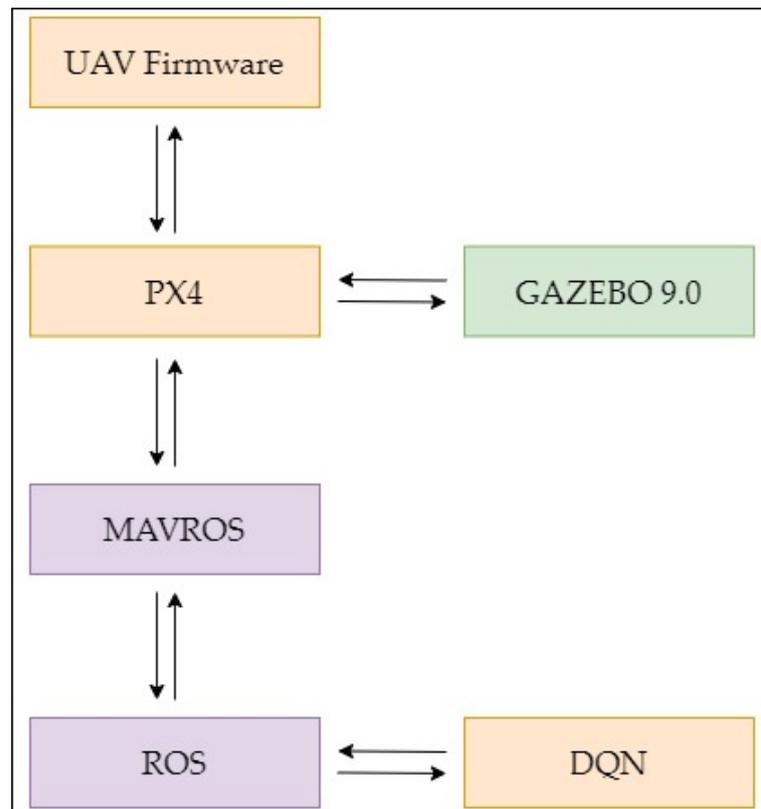


Figure 8. Framework overview. Gazebo simulates all the environments which interact with the UAV FCU PX4. DQN algorithm controls the agent via ROS/MAVROS.

4.2. Integration

To integrate the whole system, three ROS nodes were created. One was to control the action of the UAV based on DQN (agent). The other two were for running the RRT algorithm (RRT mission planner) and controlling the environment in Gazebo (world builder). Figure 9 shows the node structure and messages used to communicate. All nodes communicate with each other through messages sent by 1 kHz.

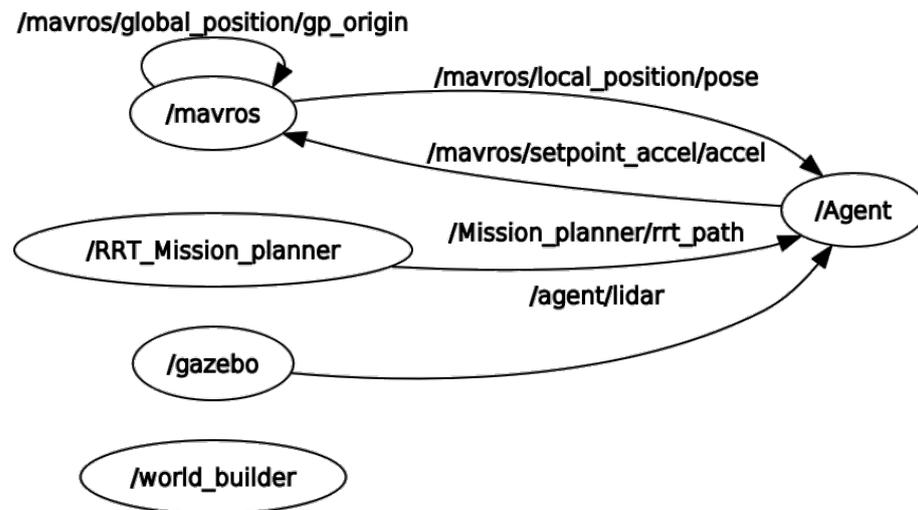


Figure 9. ROS node structure.

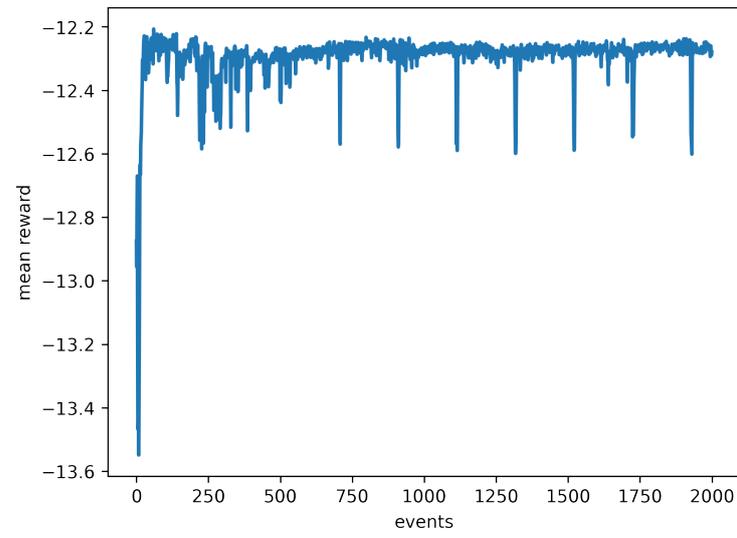
4.3. Simulation Results

The first set of simulation results is shown in Figure 10. For an environment with $[0, 0, 0] \leq Z \leq [10, 10, 10]$ and $N = 15$, the authors decided to try different DQN models to improve results, including the one proposed by de Castro et al. [73]. Table 2 shows the information about the four NN architectures with the best performance for these methodologies. It is possible to observe that the number of events, learning rate (lr), and discount factor 2000, 0.001, and 0.99, respectively, are fixed. Other values were tried for these parameters, but, since the results were poor, the authors decided just to set these parameters as a base and to fine-tune the other parameters. By analyzing Figure 10, it is possible to observe that Model 3 has a more stable response than the others. However, it has one of the least exploration factors. Model 1 has almost identical results but is slightly different during the exploitation phase. Even at that stage, the model keeps trying exploration in near states.

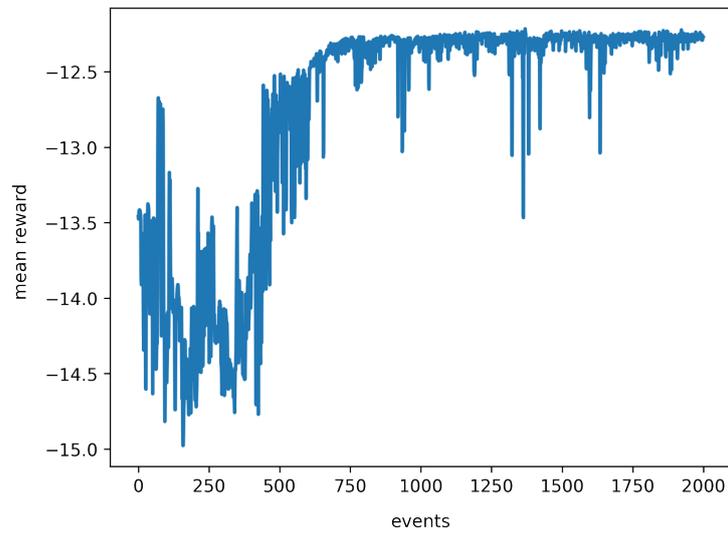
Model 4 has the worst response. It does not achieve the stable reward phase and keeps exploring the environment but cannot learn about it. Figure 11 elucidates these comparisons; Table 3 presents a comparison in terms of the total mean average.

Table 2. DQN models and hyperparameters set during the training phase.

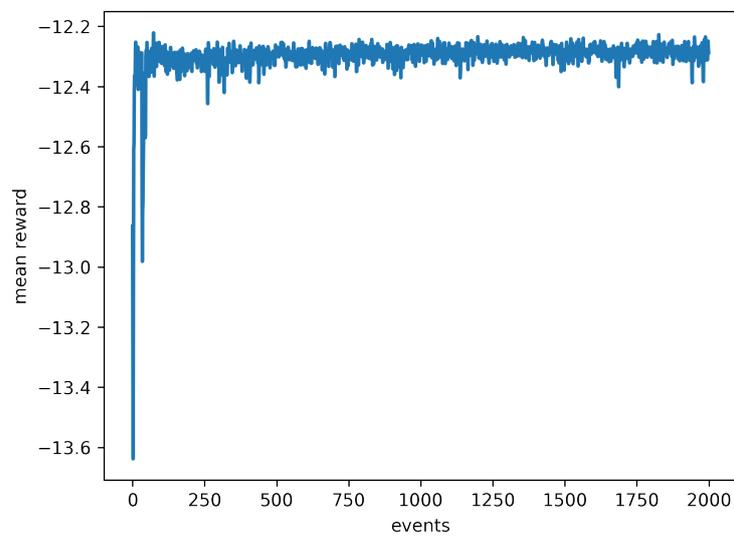
Events = 2000, lr = 0.001, $\gamma = 0.99$			
	Hidden Layers	Neurons	Steps
Model 1	1	30	1000
Model 2	1	100	500
Model 3	3	[100,60,30]	500
Model 4	1	[81,212,29]	500



(a)

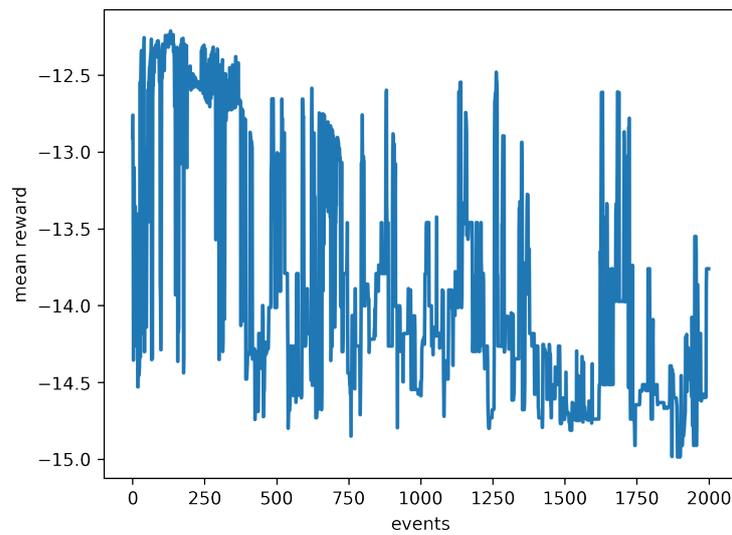


(b)



(c)

Figure 10. Cont.

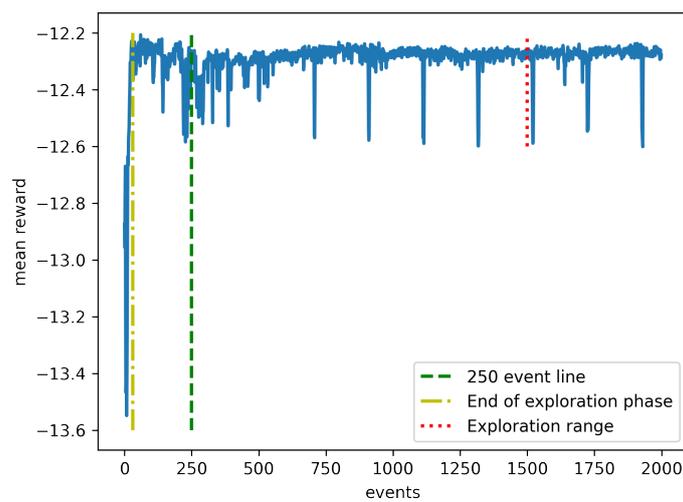


(d)

Figure 10. Reward returns for different DQN models with 2000 events and learning rate equal to 0.001. (a) Model 1 with one hidden layer 30 neurons, step size of 1000. (b) Model 2 with one hidden layer, step size of 500 and 100 neurons. (c) Model 3 with three hidden layers with 100, 60, and 30 neurons, respectively, and a step size of 500. (d) Model 4 has three hidden layers with 81, 212, and 29 neurons, respectively, and a step size of 500.

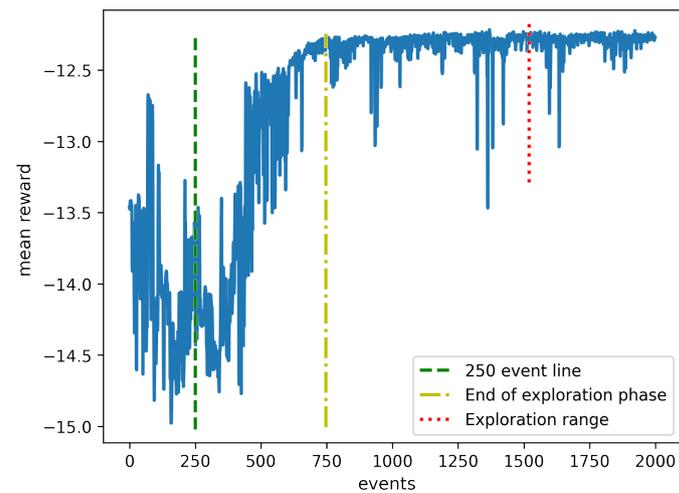
Table 3. Comparison of Model 1, 2, and 3.

	Total Mean Reward	Exploration Phase End (Event)
Model 1	-12.28	20
Model 2	-12.32	750
Model 3	-12.30	40

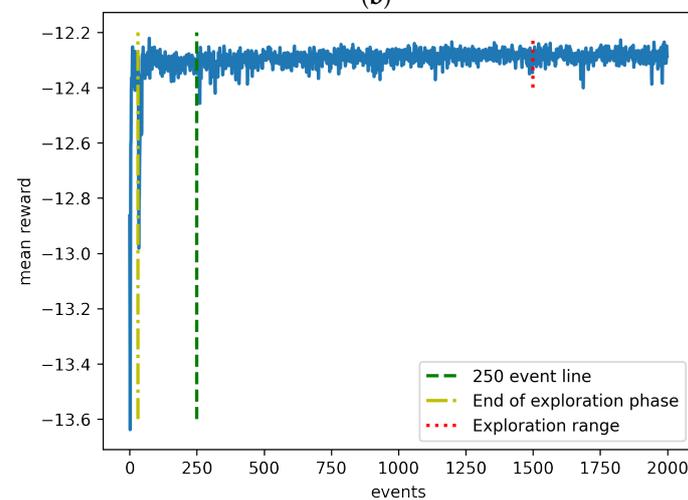


(a)

Figure 11. Cont.



(b)



(c)

Figure 11. Comparison between the best three models, where the green line shows how the model is performing in an eighth of the training, the yellow line marks the end of the exploration phase, and the red line is the variation in exploration in the exploitation phase. (a) Model 1. (b) Model 2. (c) Model 3.

4.4. Gazebo Simulation Results

The Gazebo simulation was implemented with almost the same Algorithm 2 presented in the last section, with just a few differences. The first one is the reward function. For this implementation, the reward is a function based on the difference between the D_α (UAV's velocity angle) and the T_α (RRT routes angle). The second change is that the action is no longer deterministic. The authors added a chance of 10% that the DQN output action could take any direction. The reason for this approach is to punish states that can put the agent in danger. The third is the number of steps and events. Due to the the slow training process and the results found during the simple simulation, the authors decided to adopt a strategy with fewer events and steps to reduce the training time. The last difference is the behavior of some objects.

The simulated UAV mission goal is to visit all four determined inset traps spread aleatorily inside the map using the proposed methodology to guarantee a free collision route; after visiting these points the UAV takes a free collision minimum path to return to its home position and then land. A successful experiment is when the agent achieves all these objectives.

A total of twenty experiments were simulated in the olive grown Gazebo world, with the dynamics of the objects and the insect trap poses changed in each different scenario. Figure 12 shows the position of the four insect traps in one of these simulations and the paths generated using the RRT algorithm to link them.

It is important to note that the RRT algorithm only has information about the static objects, so its path is not guaranteed to be free of collision. The line angles between the waypoints were sent via an ROS topic to the agent and then, the DQN algorithm was activated for obstacle avoidance.

In all the experiments, the proposed methodology successfully realizes the mission. Figure 13 presents how the header UAV angle differs from the angular coefficient of RRT paths during the validation tests. On average, these variations are between -1° and 1° and have a bigger divergence when the agent needs to avoid the moving obstacles.

The authors also validated the proposed methodology with an experiment to compare the DQN model and other path-planning algorithms in terms of run time. Table 4 shows the mean results for 10 simulations created with $N = 10$ and $(0, 0, 0) \leq Z_{(x,y,z)} \leq (30, 30, 30)$ and random start and stop points. Due to the randomness of the experiment, the measure run time was calculated for one single step of these algorithms in each simulation. Dijkstra's algorithm has the best results in terms of time, but, since it is a graph search algorithm, it relays on grid (1m) solutions, which returns rough paths. The Genetic Algorithm has the worst results, but returns a smoother path than the others. Finally, RRT and RRT+DQN provide continuous solutions and a smooth path, but RRT+DQN has the advantage of only calculating the RRT path for the first step and then uses DQN with 1.7 ms of run time to adjust the route for every new step when facing new obstacles.

Table 4. Path planning algorithm mean run time with 10 obstacles in a 300 m³ area.

Algorithms	Run Time (ms)
RRT+DQN	8.2
GA	8.7
Dijkstra's algorithm	2.4
RRT	6.5

It is important to note that the authors created the simulation trying to match reality, adding non-deterministic environments, non-stochastic agent actions, and random functions to change the velocity of objects dynamically. The important issue is the robustness of the algorithm. It is difficult to affirm that this methodology implementation is flawless. Even during the validation phase, when the authors forced the agent to face the most challenging environments with larger spaces of Z (>400 m³) and N (>20), the DQN's response tended to be far from optimal. These facts do not invalidate the proposed method.

In fact, in most farmer's environments, there will only be humans, animals, big machines, or other UAVs in a sparse space, which allows plenty of room to split UAV missions into mini-missions and to tackle the environment as cells. For these tasks, the proposed method can be applied and, most of the time, would work as a contingency semi-optimal obstacle avoidance letting RRT lead the path.

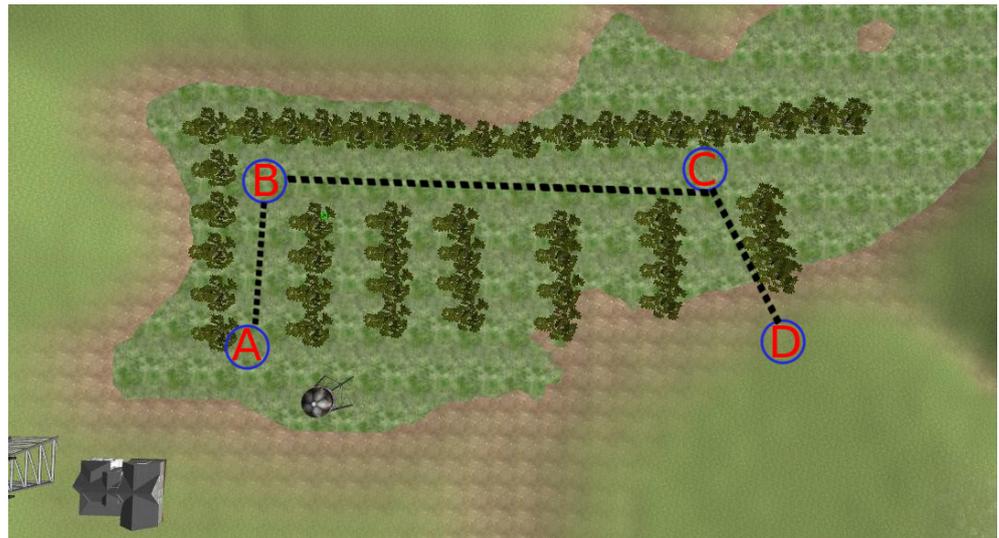


Figure 12. Insect trap poses (A–D) and paths generated by the RRT path-planning algorithm linking them.

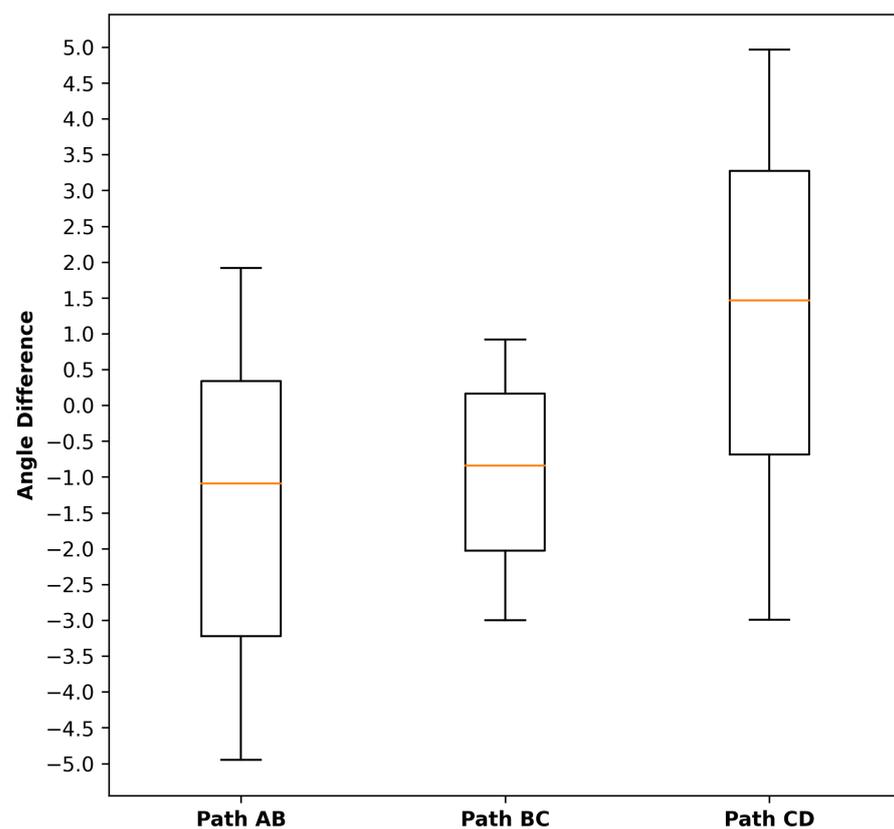


Figure 13. Angle difference between RRT path, and UAV header.

5. Conclusions and Future Work

This work presents a methodology to automate inspections on insect traps in olive-grown cultures using UAVs. Due to the dynamic nature of the environment, a strategy to guarantee the safety of the UAV and to successfully accomplish the inspection, embedded path planning, was presented in this work.

To validate this strategy, the authors created a whole simulation in Gazebo/ROS to mimic a real olive-grown culture, adding dynamic objects, UAV kinematics and non-stochastic characteristics to the UAV's movement. Twenty different scenarios were created

with random insect trap poses and object dynamics. A successful mission is when the UAV takes off, navigates to each inset trap, returns to its home location and then lands without suffering damage during the route. The simulations showed that the proposed approach succeeded in accomplishing the mission in all the experiments.

The authors also simulated twenty other scenarios, but this time comparing the proposed methodology with other well-established path-planning algorithms in run-time terms. Although the time spent on the DQN+RRT methodology was the second highest, it presented an acceptable solution, producing smooth routes in a continuous space, different from Dijkstra, which was faster but relied on space simplification strategies, and from RRT, that had a similar run time step but became slower as the number of dynamic objects increased.

In terms of evaluation, this research work opens up several future possibilities. For instance, improvements in the training phase and validation of the learning-based planning algorithm are required. Additionally, further work is needed to test the proposed solution in a real UAV, fusing different kinds of sensors, such as a depth camera, LIDAR, and sonar, which will require expanding the training by including more scenarios.

This work is part of a larger project that will be implemented in a similar olive-grown culture, but using heterogeneous collaborative robots to perform inspections. Another part of this main project has already been published in [74]. The present work represents a solution to some of the future investigations suggested in [74].

Author Contributions: Conceptualization, G.G.R.d.C. and M.F.P.; methodology, G.G.R.d.C. and M.F.P.; validation, G.G.R.d.C.; formal analysis, G.S.B., M.F.P. and J.L.; investigation, G.G.R.d.C. and M.F.P.; writing—original draft preparation, G.G.R.d.C. and M.F.P.; writing—review and editing, G.S.B., G.G.R.d.C., M.F.P., A.C., A.I.P. and J.L.; visualization, A.C., M.T., G.S.B., M.F.P., A.I.P., J.L. and G.G.R.d.C.; supervision, M.F.P.; project administration, M.F.P.; funding acquisition, J.L. All authors have read and agreed with the submission of the current manuscript version.

Funding: The authors would like to thank the following Brazilian Agencies CEFET-RJ, CAPES, CNPq, and FAPERJ. The authors also want to thank the Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança–IPB (UIDB/05757/2020 and UIDP/05757/2020), the Foundation for Science and Technology (FCT, Portugal) for financial support through national funds FCT/MCTES (PIDDAC) to CeDRI, and Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC) and IPB, Portugal. This work was carried out under the Project “OleaChain: Competências para a sustentabilidade e inovação da cadeia de valor do olival tradicional no Norte Interior de Portugal” (NORTE-06-3559-FSE-000188), an operation to hire highly qualified human resources, funded by NORTE 2020 through the European Social Fund (ESF).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The trained DQN model, the data and simulation results can be found on https://github.com/gelardrc/RRT_DQN_PATH_PLANNING.git (accessed on 10 December 2022).

Acknowledgments: The authors would like to thank the Foundation for Science and Technology (FCT, Portugal) for financial support through national funds FCT/MCTES (PIDDAC) to CeDRI (UIDB/05757/2020 and UIDP/05757/2020) and SusTEC (LA/P/0007/2021). In addition, the authors would like to thank the following Brazilian Agencies CEFET-RJ, CAPES, CNPq, and FAPERJ. The authors also want to thank the Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança (IPB)–Campus de Santa Apolonia, Portugal and Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Portugal. This work was carried out under the Project “OleaChain: Competências para a sustentabilidade e inovação da cadeia de valor do olival tradicional no Norte Interior de Portugal” (NORTE-06-3559-FSE-000188), an operation used to hire highly qualified human resources, funded by NORTE 2020 through the European Social Fund (ESF).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

APF	Artificial Potential Field
LR	Learning Rate
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DRL	Deep Reinforcement Learning
DQN	Deep Q-Network
GRL	Geometric Reinforcement Learning
HJB	Hamilton–Jacobi–Bellman
IOT	Internet of Things
LiDAR	Light Detection and Ranging
NN	Neural Network
UAV	Unmanned Aerial Vehicle
UAG	Unmanned Ground Vehicle
RRT	Rapidly Exploring Random Trees
ROS	Robotic Operating System
SITL	Software in the loop

References

1. Yaqot, M.; Menezes, B.C. Unmanned aerial vehicle (UAV) in precision agriculture: Business information technology towards farming as a service. In Proceedings of the 2021 1st International Conference on Emerging Smart Technologies and Applications (eSmarTA), Sana'a, Yemen, 10–12 August 2021; pp. 1–7.
2. Almalki, F.A.; Soufiene, B.O.; Alsamhi, S.H.; Sakli, H. A low-cost platform for environmental smart farming monitoring system based on IoT and UAVs. *Sustainability* **2021**, *13*, 5908. [[CrossRef](#)]
3. Bai, X.; Cao, M.; Yan, W.; Ge, S.S. Efficient routing for precedence-constrained package delivery for heterogeneous vehicles. *IEEE Trans. Autom. Sci. Eng.* **2019**, *17*, 248–260. [[CrossRef](#)]
4. Barrile, V.; Simonetti, S.; Citroni, R.; Fotia, A.; Bilotta, G. Experimenting Agriculture 4.0 with Sensors: A Data Fusion Approach between Remote Sensing, UAVs and Self-Driving Tractors. *Sensors* **2022**, *22*, 7910. [[CrossRef](#)] [[PubMed](#)]
5. Luque Vega, L.F.; Lopez-Neri, E.; Arellano-Muro, C.A.; Gonzalez Jimenez, L.E.; Ghommam, J.; Carrasco Navarro, R. UAV Flight Instructional Design for Industry 4.0 based on the Framework of Educational Mechatronics. In Proceedings of the IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 18–21 October 2020; pp. 2313–2318. [[CrossRef](#)]
6. Liu, T.; Sun, Y.; Wang, C.; Zhang, Y.; Qiu, Z.; Gong, W.; Lei, S.; Tong, X.; Duan, X. Unmanned aerial vehicle and artificial intelligence revolutionizing efficient and precision sustainable forest management. *J. Clean. Prod.* **2021**, *311*, 127546. [[CrossRef](#)]
7. Raj, M.; Gupta, S.; Chamola, V.; Elhence, A.; Garg, T.; Atiquzzaman, M.; Niyato, D. A survey on the role of Internet of Things for adopting and promoting Agriculture 4.0. *J. Netw. Comput. Appl.* **2021**, *187*, 103107. [[CrossRef](#)]
8. Lyu, X.; Li, X.; Dang, D.; Dou, H.; Wang, K.; Lou, A. Unmanned Aerial Vehicle (UAV) Remote Sensing in Grassland Ecosystem Monitoring: A Systematic Review. *Remote Sens.* **2022**, *14*, 1096. [[CrossRef](#)]
9. Olson, D.; Anderson, J. Review on unmanned aerial vehicles, remote sensors, imagery processing, and their applications in agriculture. *Agron. J.* **2021**, *113*, 971–992. [[CrossRef](#)]
10. Su, J.; Zhu, X.; Li, S.; Chen, W.H. AI meets UAVs: A survey on AI empowered UAV perception systems for precision agriculture. *Neurocomputing* **2022**. [[CrossRef](#)]
11. Shakhathreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *IEEE Access* **2019**, *7*, 48572–48634. [[CrossRef](#)]
12. Stek, T.D. Drones over Mediterranean landscapes. The potential of small UAV's (drones) for site detection and heritage management in archaeological survey projects: A case study from Le Pianelle in the Tappino Valley, Molise (Italy). *J. Cult. Herit.* **2016**, *22*, 1066–1071. [[CrossRef](#)]
13. Ramos, G.S.; Pinto, M.F.; Coelho, F.O.; Honório, L.M.; Haddad, D.B. Hybrid methodology based on computational vision and sensor fusion for assisting autonomous UAV on offshore messenger cable transfer operation. *Robotica* **2022**, *40*, 2786–2814. [[CrossRef](#)]
14. Biundini, I.Z.; Melo, A.G.; Pinto, M.F.; Marins, G.M.; Marcato, A.L.M.; Honorio, L.M. Coverage Path Planning Optimization for Slopes and Dams Inspection. In Proceedings of the Robot 2019: Fourth Iberian Robotics Conference, Porto, Portugal, 20–22 November 2019; Silva, M.F., Luís Lima, J., Reis, L.P., Sanfeliu, A., Tardioli, D., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 513–523. [[CrossRef](#)]
15. Alsalam, B.H.Y.; Morton, K.; Campbell, D.; Gonzalez, F. Autonomous UAV with vision based on-board decision making for remote sensing and precision agriculture. In Proceedings of the 2017 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2017; pp. 1–12. [[CrossRef](#)]

16. Zhang, C.; Kovacs, J.M. The application of small unmanned aerial systems for precision agriculture: A review. *Precis. Agric.* **2012**, *13*, 693–712. [[CrossRef](#)]
17. Anastasiou, E.; Balafoutis, A.; Fountas, S. Trends in Remote Sensing Technologies in Olive Cultivation. *Smart Agric. Technol.* **2022**, *3*, 100103. [[CrossRef](#)]
18. Alvarez-Vanhard, E.; Corpetti, T.; Houet, T. UAV & satellite synergies for optical remote sensing applications: A literature review. *Sci. Remote Sens.* **2021**, *3*, 100019. [[CrossRef](#)]
19. Parasuraman, K.; Anandan, U.; Anbarasan, A. IoT Based Smart Agriculture Automation in Artificial Intelligence. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021; pp. 420–427.
20. Anwarul, S.; Misra, T.; Srivastava, D. An IoT & AI-assisted Framework for Agriculture Automation. In Proceedings of the 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 13–14 October 2022; pp. 1–6. [[CrossRef](#)]
21. Shaikh, T.; Mir, W.; Rasool, T.; Sofi, S. Machine Learning for Smart Agriculture and Precision Farming: Towards Making the Fields Talk. *Arch. Comput. Methods Eng.* **2022**, *29*, 4557–4597. [[CrossRef](#)]
22. Kamil, F.; Tang, S.; Khaksar, W.; Zulkifli, N.; Ahmad, S. A review on motion planning and obstacle avoidance approaches in dynamic environments. *Adv. Robot. Autom.* **2015**, *4*, 134–142.
23. Garcia, M.P.; Montiel, O.; Castillo, O.; Sepulveda, R.; Melin, P. Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Appl. Soft Comput.* **2009**, *9*, 1102–1110. [[CrossRef](#)]
24. Orozco-Rosas, U.; Picos, K.; Montiel, O. Hybrid path planning algorithm based on membrane pseudo-bacterial potential field for autonomous mobile robots. *IEEE Access* **2019**, *7*, 156787–156803. [[CrossRef](#)]
25. Miao, H.; Tian, Y.C. Dynamic robot path planning using an enhanced simulated annealing approach. *Appl. Math. Comput.* **2013**, *222*, 420–437. [[CrossRef](#)]
26. López-Villalta, M.C. *Olive Pest and Disease Management*; International Olive Oil Council: Madrid, Spain, 1999.
27. Gonçalves, F.; Torres, L. The use of trap captures to forecast infestation by the olive fly, *Bactrocera oleae* (Rossi) (Diptera: Tephritidae), in traditional olive groves in north-eastern Portugal. *Int. J. Pest Manag.* **2013**, *59*, 279–286. [[CrossRef](#)]
28. Liu, X.; Chen, S.W.; Nardari, G.V.; Qu, C.; Ojeda, F.C.; Taylor, C.J.; Kumar, V. Challenges and Opportunities for Autonomous Micro-UAVs in Precision Agriculture. *IEEE Micro* **2022**, *42*, 61–68. [[CrossRef](#)]
29. Koenig, N.; Howard, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), Sendai, Japan, 28 September–2 October 2004; pp. 2149–2154. Volume 3. [[CrossRef](#)]
30. Roy Choudhury, M.; Das, S.; Christopher, J.; Apan, A.; Chapman, S.; Menzies, N.W.; Dang, Y.P. Improving Biomass and Grain Yield Prediction of Wheat Genotypes on Sodic Soil Using Integrated High-Resolution Multispectral, Hyperspectral, 3D Point Cloud, and Machine Learning Techniques. *Remote Sens.* **2021**, *13*, 3482. [[CrossRef](#)]
31. Awais, M.; Li, W.; Cheema, E.M.J.; Zaman, Q.; Shah, A.; Aslam, B.; Zhu, W.; Ajmal, M.; Faheem, M.; Hussain, S.; et al. UAV-based remote sensing in plant stress imagine using high-resolution thermal sensor for digital agriculture practices: A meta-review. *Int. J. Environ. Sci. Technol.* **2023**, *20*, 1135–1152. [[CrossRef](#)]
32. Velusamy, P.; Rajendran, S.; Mahendran, R.K.; Naseer, S.; Shafiq, M.; Choi, J.G. Unmanned Aerial Vehicles (UAV) in Precision Agriculture: Applications and Challenges. *Energies* **2022**, *15*, 217. [[CrossRef](#)]
33. Gao, T.; Gao, Z.; Sun, B.; Qin, P.; Li, Y.; Yan, Z. An Integrated Method for Estimating Forest-Canopy Closure Based on UAV LiDAR Data. *Remote Sens.* **2022**, *14*, 4317. [[CrossRef](#)]
34. Roma, E.; Catania, P. Precision Oliviculture: Research Topics, Challenges, and Opportunities—A Review. *Remote Sens.* **2022**, *14*, 1668. [[CrossRef](#)]
35. Morales Rodriguez, P.A.; Cano Cano, E.; Villena, J.; Lopez Perales, J.A. A Comparison between Conventional Sprayers and New UAV Sprayers: A Study Case of Vineyards and Olives in Extremadura (Spain). *Agronomy* **2022**, *12*, 1307. [[CrossRef](#)]
36. Di Nisio, A.; Adamo, F.; Acciani, G.; Attivissimo, F. Fast Detection of Olive Trees Affected by *Xylella Fastidiosa* from UAVs Using Multispectral Imaging. *Sensors* **2020**, *20*, 4915. [[CrossRef](#)]
37. Rahman, M.F.F.; Fan, S.; Zhang, Y.; Chen, L. A Comparative Study on Application of Unmanned Aerial Vehicle Systems in Agriculture. *Agriculture* **2021**, *11*, 22. [[CrossRef](#)]
38. Anifantis, A.S.; Camposeo, S.; Vivaldi, G.A.; Santoro, F.; Pascuzzi, S. Comparison of UAV Photogrammetry and 3D Modeling Techniques with Other Currently Used Methods for Estimation of the Tree Row Volume of a Super-High-Density Olive Orchard. *Agriculture* **2019**, *9*, 233. [[CrossRef](#)]
39. Radoglou-Grammatikis, P.; Sarigiannidis, P.; Lagkas, T.; Moscholios, I. A compilation of UAV applications for precision agriculture. *Comput. Netw.* **2020**, *172*, 107148. [[CrossRef](#)]
40. Ampatzidis, Y.; Partel, V.; Costa, L. Agroview: Cloud-based application to process, analyze and visualize UAV-collected data for precision agriculture applications utilizing artificial intelligence. *Comput. Electron. Agric.* **2020**, *174*, 105457. [[CrossRef](#)]
41. Awais, M.; Li, W.; Cheema, E.M.J.; Hussain, S.; Shah, A.; Aslam, B.; Liu, C.; Ali, A. Assessment of optimal flying height and timing using high-resolution unmanned aerial vehicle images in precision agriculture. *Int. J. Environ. Sci. Technol.* **2022**, *19*, 2703–2720. [[CrossRef](#)]

42. Lytridis, C.; Kaburlasos, V.G.; Pachidis, T.; Manios, M.; Vrochidou, E.; Kalampokas, T.; Chatzistamatis, S. An Overview of Cooperative Robotics in Agriculture. *Agronomy* **2021**, *11*, 1818. [[CrossRef](#)]
43. Delavarpour, N.; Koparan, C.; Nowatzki, J.; Bajwa, S.; Sun, X. A Technical Study on UAV Characteristics for Precision Agriculture Applications and Associated Practical Challenges. *Remote Sens.* **2021**, *13*, 1204. [[CrossRef](#)]
44. Singh, N.H.; Thongam, K. Neural network-based approaches for mobile robot navigation in static and moving obstacles environments. *Intell. Serv. Robot.* **2019**, *12*, 55–67. [[CrossRef](#)]
45. Sung, I.; Choi, B.; Nielsen, P. On the training of a neural network for online path planning with offline path planning algorithms. *Int. J. Inf. Manag.* **2021**, *57*, 102142. [[CrossRef](#)]
46. Yu, J.; Su, Y.; Liao, Y. The path planning of mobile robot by neural networks and hierarchical reinforcement learning. *Front. Neurobotics* **2020**, *14*, 63. [[CrossRef](#)]
47. Shiri, H.; Park, J.; Bennis, M. Remote UAV online path planning via neural network-based opportunistic control. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 861–865. [[CrossRef](#)]
48. Yang, K.; Sukkariéh, S. Real-time continuous curvature path planning of UAVs in cluttered environments. In Proceedings of the 2008 5th International Symposium on Mechatronics and Its Applications, Amman, Jordan, 27–29 May 2008; pp. 1–6. [[CrossRef](#)]
49. Yan, F.; Liu, Y.S.; Xiao, J.Z. Path planning in complex 3D environments using a probabilistic roadmap method. *Int. J. Autom. Comput.* **2013**, *10*, 525–533. [[CrossRef](#)]
50. Debnath, S.K.; Omar, R.; Bagchi, S.; Sabudin, E.N.; Kandar, M.H.A.S.; Foysol, K.; Chakraborty, T.K. Different Cell Decomposition Path Planning Methods for Unmanned Air Vehicles—A Review. In *Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019*; Lecture Notes in Electrical Engineering; Springer: Singapore, 2020; pp. 99–111.
51. Schøler, F.; la Cour-Harbo, A.; Bisgaard, M. Generating configuration spaces and visibility graphs from a geometric workspace for uav path planning. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Portland, OR, USA, 8–11 August 2011; pp. 8–11.
52. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [[CrossRef](#)]
53. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [[CrossRef](#)]
54. Carsten, J.; Ferguson, D.; Stentz, A. 3D field D: Improved path planning and replanning in three dimensions. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 3381–3386.
55. Juneja, S.S.; Saraswat, P.; Singh, K.; Sharma, J.; Majumdar, R.; Chowdhary, S. Travelling salesman problem optimization using genetic algorithm. In Proceedings of the 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 4–6 February 2019; pp. 264–268.
56. Horn, J.F.; Schmidt, E.M.; Geiger, B.R.; DeAngelo, M.P. Neural network-based trajectory optimization for unmanned aerial vehicles. *J. Guid. Control Dyn.* **2012**, *35*, 548–562. [[CrossRef](#)]
57. Horn, J.; Geiger, B.; Schmidt, E. Use of neural network approximation in multiple-unmanned aerial vehicle trajectory optimization. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Chicago, IL, USA, 10–13 August 2009; p. 6103.
58. Zhao, Y.; Zheng, Z.; Liu, Y. Survey on computational-intelligence-based UAV path planning. *Knowl.-Based Syst.* **2018**, *158*, 54–64. [[CrossRef](#)]
59. Chen, X.; Zhang, J. The three-dimension path planning of UAV based on improved artificial potential field in dynamic environment. In Proceedings of the 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 26–27 August 2013; Volume 2, pp. 144–147.
60. Singh, M.K.; Parhi, D.R. Path optimisation of a mobile robot using an artificial neural network controller. *Int. J. Syst. Sci.* **2011**, *42*, 107–120. [[CrossRef](#)]
61. Motlagh, O.; Nakhaeinia, D.; Tang, S.H.; Karasfi, B.; Khaksar, W. Automatic navigation of mobile robots in unknown environments. *Neural Comput. Appl.* **2014**, *24*, 1569–1581. [[CrossRef](#)]
62. Qureshi, A.H.; Simeonov, A.; Bency, M.J.; Yip, M.C. Motion planning networks. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 2118–2124.
63. Yan, C.; Xiang, X.; Wang, C. Towards real-time path planning through deep reinforcement learning for a UAV in dynamic environments. *J. Intell. Robot. Syst.* **2020**, *98*, 297–309. [[CrossRef](#)]
64. Tong, G.; Jiang, N.; Biyue, L.; Xi, Z.; Ya, W.; Wenbo, D. UAV navigation in high dynamic environments: A deep reinforcement learning approach. *Chin. J. Aeronaut.* **2021**, *34*, 479–489.
65. Smolyanskiy, N.; Kamenev, A.; Smith, J.; Birchfield, S. Toward low-flying autonomous MAV trail navigation using deep neural networks for environmental awareness. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 4241–4247.
66. Zhang, B.; Mao, Z.; Liu, W.; Liu, J. Geometric reinforcement learning for path planning of UAVs. *J. Intell. Robot. Syst.* **2015**, *77*, 391–409. [[CrossRef](#)]
67. Lei, X.; Zhang, Z.; Dong, P. Dynamic path planning of unknown environment based on deep reinforcement learning. *J. Robot.* **2018**, *2018*, 578159. [[CrossRef](#)]
68. Tsai, J.T.; Chou, J.H.; Liu, T.K. Tuning the structure and parameters of a neural network by using hybrid Taguchi-genetic algorithm. *IEEE Trans. Neural Netw.* **2006**, *17*, 69–80. [[CrossRef](#)] [[PubMed](#)]

69. Wu, K.; Esfahani, M.A.; Yuan, S.; Wang, H. TDPP-Net: Achieving three-dimensional path planning via a deep neural network architecture. *Neurocomputing* **2019**, *357*, 151–162. [[CrossRef](#)]
70. Andrade, F.A.; Guedes, I.P.; Carvalho, G.F.; Zachi, A.R.; Haddad, D.B.; Almeida, L.F.; de Melo, A.G.; Pinto, M.F. Unmanned Aerial Vehicles Motion Control with Fuzzy Tuning of Cascaded-PID Gains. *Machines* **2021**, *10*, 12. [[CrossRef](#)]
71. Yang, Y.; Juntao, L.; Lingling, P. Multi-robot path planning based on a deep reinforcement learning DQN algorithm. *CAAI Trans. Intell. Technol.* **2020**, *5*, 177–183. [[CrossRef](#)]
72. Zhang, W.; Song, K.; Rong, X.; Li, Y. Coarse-to-Fine UAV Target Tracking with Deep Reinforcement Learning. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1522–1530. [[CrossRef](#)]
73. de Castro, G.G.; Pinto, M.F.; Biundini, I.Z.; Melo, A.G.; Marcato, A.L.; Haddad, D.B. Dynamic Path Planning Based on Neural Networks for Aerial Inspection. *J. Control. Autom. Electr. Syst.* **2022**, *34*, 85–105.
74. Berger, G.S.; Teixeira, M.; Cantieri, A.; Lima, J.; Pereira, A.I.; Valente, A.; Castro, G.G.R.D.; Pinto, M.F. Cooperative Heterogeneous Robots for Autonomous Insects Trap Monitoring System in a Precision Agriculture Scenario. *Agriculture* **2023**, *13*, 239. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.